IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 22, 2019, accepted May 1, 2019, date of publication May 8, 2019, date of current version May 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2915673

Multiple Ant Colony Optimization Based on Pearson Correlation Coefficient

HONGWEI ZHU[1], XIAOMING YOU[1], AND SHENG LIU[2]
[1]College of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China
[2]School of Management, Shanghai University of Engineering Science, Shanghai 201620, China

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

This work was supported in part by the Natural Science Foundation of China under Grant 61673258, Grant 61075115, Grant 61403249, and Grant 61603242.

ABSTRACT Ant Colony Optimization algorithms have been successfully applied to solve the Traveling Salesman Problem (TSP). However, they still have a tendency to fall into local optima, mainly resulting from poor diversity, especially in those TSPs with a lot of cities. To address this problem, and further obtain a better result in big-scale TSPs, we propose an algorithm called Multiple Colonies Ant Colony Optimization Based on Pearson Correlation Coefficient (PCCACO). To improve the diversity, first, we introduce a novel single colony termed Unit Distance-Pheromone Operator, which along with two other classic ant populations: Ant Colony System and Max-Min Ant System, make the final whole algorithm. A Pearson correlation coefficient is further employed to erect multi-colony communication with an adaptive frequency. Besides that, an initialization is applied when the algorithm is stagnant, which helps it to jump out of the local optima. Finally, we render a dropout approach to reduce the running time. The extensive simulations in TSP demonstrate that our algorithm can get a better solution with a reasonable variation.

INDEX TERMS Multiple colonies, TSP, Pearson correlation coefficient, cities-dropout, Unit Distance-Pheromone Operator.

I. INTRODUCTION

TSP is one of the famous NP-hard problems in the field of mathematics where the salesman attempts to find the shortest tour through cities. But Each city can only be visited once. Many optimization algorithms can solve TSP, including Particle Swarm Optimization (PSO) [1], Genetic Algorithm (GA) [2], Simulated Annealing (SA) [3], Ant Colony Optimization (ACO) [4]–[6] and so on. Every algorithm has some superiorities and weaknesses. At present, ACO and PSO are the two main optimization algorithms in solving TSP.

Some scientists find that after an ant passes a path, it will release a kind of hormone, called pheromones. Other ants will choose the path to the food according to the concentration of pheromone. The Italian scholars M. Dorigo and V. Maniezzo, were inspired by the mechanism of biological evolution, simulated the behavior of natural ants and proposed the Ant System (AS) [4] which has attracted the attention of researchers all over the world. Experiments have shown that AS has achieved good results in TSPs.

The associate editor coordinating the review of this manuscript and approving it for publication was Mufti Mahmud.

The positive feedback effect of the pheromone makes the algorithm more directional, but at the same time, it's possibly to makes the algorithm easy to fall into local optima. To address it, in 1996, M. Dorigo proposed the Ant Colony System (ACS) [5] algorithm based on the AS. He proposed two pheromone updating methods and added a local search process. The algorithm reduces the probabilities of dropping down local optima.

Similarly, in order to limit the concentration of pheromone, T. Stutzle *et al.* proposed the MAX-MIN ant system (MMAS) [6], which set a threshold to restrict the maximum and minimum pheromone level. When the algorithm falls into local optima, MMAS reinitializes the pheromone, which improves the abilities of the ACO algorithm.

Although [5] and [6] have good performance on TSP, they still have a tendency to fall into local optima, especially in those TSPs with more nodes. So, researchers continued to improve the performance of these two algorithms.

The parameters setting of the ACO have a profound impact on the experimental results. Many researchers use various methods to optimize the parameters of the ACO [7]–[9]. They all used the PSO to optimize the parameters in ACO, which

reduce the impact of parameter selection on the experiment and enhance the quality of solutions.

Adding local optimization algorithms can also lead to a better result [10]–[16]. To improve the accuracy of the solution, Chen H, *et al.* proposed a search method in which the Tabu search algorithm has been added on the basis of the original [10]. Deif D *et al.* presented an Ant Colony Optimization with a local search heuristic to solve a reliable wireless sensor network [11]. Ellabib I *et al.* proposed multiple Ant Colony System with a local search to solve Vehicle Routing Problem and TSPs [12]. Yong W presented a hybrid Max-Min ant system with a local search algorithm. The algorithm is tested with small and large-scale TSP instances [13]. Zhou *et al.* proposed a discrete Invasive Weed Optimization (IWO) which used the 3-Optimization (3-Opt) local search algorithm for TSPs [14]. Othman *et al.* presented a new method based on the Water Flow-like Algorithm (WFA) combined with 2-Optimization (2-Opt) and 3-Opt simple local search algorithms [15].

In fact, there are various sorts of ants even in single colony. These kinds of ants have different divisions of labor. Under this division of labor, the ant colony will be allowed to complete some complex activities quickly. Therefore, the multi-colonies ant colony algorithm has stronger advantages in solving complex and difficult problems.

Twomey C *et al.* proposed a variety of methods for communication among ant colonies, like unidirectional ring (R), hypercube (HC) and fully-connected (FC) and so on [17]. Xu M, *et al.* divided the ants into two colonies, which combines ACS with Rank-based Ant System (RbAS) and uses the differences in pheromone update to complement each other to improve the quality of solution [18]. Chu S C *et al.* proposed seven communication strategies which update the pheromone level according to the best route of all groups, updating the pheromone level between each pair of groups and so on [19]. Wang L *et al.* presented four pheromone modification strategies in the ant colony system to solve Data-Intensive Service Provision [20].

Y.–Z. Y *et al.* suggested a reward mechanism of eliminating the fittest to balance between intensification and diversification. When an ant finds a best solution, its parameters are rewarded. These improvement strategies help AS algorithm achieve better performance [21]. Liao E *et al.* proposed a hierarchical hybrid algorithm for TSP. The TSP problem is decomposed into a few subproblems with small-scale nodes by density peaks clustering algorithm and every subproblem is resolved by ant colony optimization algorithm [22].

According to reference [23] and [24], they both introduced the Artificial Bee Colony (ABC) algorithm. They divided the swarm intelligence algorithms into two colonies: ACO and ABC. The results reveal that ACO-ABC produces better quality solutions. G.F. Dong *et al.* proposed a new hybrid algorithm, which combine both GA and ACO together in a cooperative manner to improve the performance of ACO for solving TSPs [25]. S.M. Chen and C-Y Chien presented a new cooperative method using SA, ACO, PSO for solving TSP. ACO generate initial solutions for GA. Then, GA enhances these initial solutions. PSO exchanges the pheromone among ant groups [28].

K. Jun-man *et al.* introduced individual variation and routing strategies to reduce the processing costs involved with routing of ants in the conventional ACO [26]. M. Peker proposed an algorithm based on ACO and the Taguchi method for the solving TSP. Taguchi method is used to optimize the parameters of ACO [27]. W. Junqiang *et al.* suggested a hybrid Ant Colony Optimizer and delete-cross method which can prevent the overlapped edges [29]. Zhao J H *et al.* proposed a Multi-objective Ant Colony System for the reliability optimization problems. The algorithm offers distinct advantages to these problems compared to other optimization methods [30].

ACO have been successfully applied to solve TSP. However, they still have a tendency to fall into local optima, mainly resulting from poor diversity, especially in those TSPs with a lot of cities. In this paper, we focus on improving the accuracy of the solution on large-scale TSPs and reduce running time of the algorithm. The main contributions of this paper are summarized as follows:

1. A Unit Distance-Pheromone Operator (UDPO) is proposed as a single colony by combining the distance and pheromone factors.

2. The Pearson correlation coefficient is introduced as the evaluation criterion, and reward the parameters of similar paths with an adaptive frequency. When the algorithm is stagnant, reinitializes the reward parameters.

3. A cities-dropout idea is proposed during path construction.

This paper is organized as follows. Section II introduces the ACS and MMAS algorithms. Section III presents the proposed improved method about Pearson correlation coefficient, reward, cities-dropout. Section IV reveals the experimental results in solving the TSPs and comparison among different ACO. Finally, Section V summarizes our work and describes some of our future directions.

II. MATERIALS AND METHODS

A. PEARSON CORRELATION COEFFICIENT

In machine learning, the Pearson correlation coefficient is an important method to measure the similarity of multiple data variables [31], [32]. Its value is between [−1,1]. When the correlation coefficient equals 1, it becomes completely positive correlation; when it equals −1, it becomes completely negative correlation. It means that the greater the absolute value of the correlation coefficient, the stronger the correlation and vice versa.

$$\rho_{X,Y} = \frac{\mathrm{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E((X-\mu_X)(Y-\mu_Y)}{\sigma_X \sigma_Y} = \frac{E(XY)-E(X)E(Y)}{\sqrt{E(X^2)-E^2(X)}\sqrt{B(Y^2)-E^2(Y)}} \quad (1)$$

where $\mathrm{cov}(X, Y)$ is the covariance between X and Y, σ_X, σ_Y are standard deviations of X and Y. $E(X)$ is the expected value of X.

B. ANT COLONY OPTIMIZATION WITH TSP

For different NP-hard problems, the mathematical model established using the ant colony algorithm is also different. This paper takes the most typical TSP problems in the NP-hard problem as an example to illustrate the model of the basic ant colony algorithm.

Let m be the number of ants and n be the number of cities. d_{ij} (i, j = 1,2,..) is the distance between city i and city j. $\tau_{\mathrm{ij}}(t)$ represents the concentration of pheromone between city i and city j at *t-th* iteration.

1) CONSTRUCT THE SOLUTION

The selection formula for each ant in the ACS from city i to city j:

$$S = \begin{cases} \arg \max\limits_{j \in allowed} \{\tau_{ij} \cdot \eta_{ij}^{\beta}\} & if\ q \le q_0 \\ s & otherwise \end{cases} \tag{2}$$

where η_{ij} represents the reciprocal of the distance between city i and cities j, τ_{ij} represents pheromone concentration between city i and city j, q_0 is a parameter which is a consistent value. q is a random variable subjected to a uniform distribution between 0 to 1. S is the next city to be selected. s is equal to Eq.(3). *Allowed* is the current feasible city set of ants. The Eq.(2) shows that cities with high pheromones or with relatively close distances are more likely to be selected. When $q < q_0$, use Eq.(2); otherwise, use Eq.(3).

$$P_{ij}(t) = \begin{cases} \dfrac{[\tau_{ij}(t)]^{\alpha}[\eta_{ij}]^{\beta}}{\sum\limits_{s \in allowed} [\tau_{is}(t)]^{\alpha}[\eta_{is}]^{\beta}} & if\ j \in allowed \\ 0 & otherwise \end{cases} \tag{3}$$

where *allowed* is the current feasible city set of ants.

2) PHEROMONE UPDATE

The method of pheromone updating in ACS is divided into global pheromone update and local pheromone update.

Global pheromone update: when all ants complete a round tour, the algorithm only allows the optimal ants of each iteration to release pheromones. The release of pheromone is mainly used to increase the convergence speed, which makes the choice of ants more directional. Eq.(4) and Eq.(5) are the ways of global pheromone update:

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

$$\Delta\tau_{\mathrm{ij}} = \begin{cases} 1/L_{gb} & if\ (i,j) \in BestTour \\ 0 & else \end{cases} \tag{5}$$

where ρ is the evaporation rate of the global pheromone update, $\Delta\tau_{ij}$ is the pheromone increment, L_{gb} is the length of the global-best tour till now.

Local pheromone update: when all ants complete the tour, the algorithm allows each ant to update the pheromone of the path where it passed through in Eq.(6). The main pheromone is to narrow the gap between the optimal path and non-optimal path, which increase the diversity of the ACS algorithm.

$$\tau_{\mathrm{ij}}(t+1) = (1-\zeta)\tau_{ij}(t) + \zeta\tau_0 \tag{6}$$

where $\tau_{\mathrm{ij}}(t)$ represents the concentration of pheromone between city i and city j at *t-th* iteration. τ_0 is the initial pheromone, ζ is the evaporation rate of local pheromone update.

C. MAX-MIN ANT SYSTEM

In order to avoid the algorithm converge too fast and stagnate, the MMAS algorithm limits the pheromone to a certain range: $[\tau_{\min}, \tau_{\max}]$. If $\tau_{ij} \le \tau_{\min}$, we set $= \tau_{\min}$; if $\tau_{ij} \ge \tau_{\max}$, we set $\tau_{ij} = \tau_{\max}$. The value of $\tau_{\max}$ and $\tau_{\min}$ are in Eq.(7) and Eq.(8).

$$\tau_{\max} = (1/\psi) * (1/T^{gb}) \tag{7}$$

$$\tau_{\min} = \tau_{\max}/2n \tag{8}$$

where T^{gb} is global optimal path, ψ is the evaporation rate.

1) PHEROMONE UPDATE

When an iteration completes, only one ant updates the pheromone, and the pheromone is updated on the iteration-best solution or the global-best solution. Therefore, the optimal solution is effectively utilized, and the exploration capability of the algorithm is also enhanced. The pheromone update rules are as shown in Eq.(9) and Eq.(10):

$$\tau_{ij}(t+1) = (1-\rho)\tau_{ij}(t) + \Delta\tau_{ij}^{best} \tag{9}$$

$$\Delta\tau_{ij}^{best} = 1/f(s^{best}) \tag{10}$$

where $f(s^{best})$ is either the iteration-best solution or the global-best solution.

III. MULTI-COLONIES ANT COLOGY OPTIMIZATION BASED ON PEARSON CORRELATION COEFFICIENT

On experimental data from the literature [18], they used the differences in pheromone update between ACS and RbAS to complement each other to improve the quality of solution. But they did not consider the influence of the parameters of ACO. Furthermore, the TSP instances they used belong to middle-scale, such as KroA100. They didn't have much experimentation on large-scale TSPs. Therefore, it has not been well solved to the drawbacks of falling into local optima and poor diversity in ACO especially on large-scale TSPs. In view of the above literature, we propose the PCCACO to solve large-scale problems. We named the ACS algorithm as Population1, the Unit Distance-Pheromone Operator is named Population2 (be introduced in section III, subsection A), MMAS is named Population3. They construct the path independently for each colony. We introduce the Pearson correlation coefficient in machine learning to measure the similarity of each colony and exchange information between the two colonies with the highest similarity.

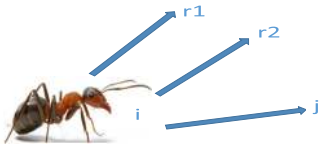

FIGURE 1. Path construction of the ant.

At the same time, we reward the common path parameters of the two colonies to achieve the faster convergence of PCCACO. The algorithm adjusts the communication frequency in the medium stage to ensure the variety of each colony. The following is the detailed analysis of the improvement of PCCACO.

A. UNIT DISTANCE-PHEROMONE OPERATOR (POPULATION 2)

In Fig.1, we assume that there are two cases.

Case1: when the length of i to r_1 is larger than i to $r_2(d_{ir_1} > d_{ir_2})$ and the pheromones of i to r_1 is less than i to r_2 $(\tau_{ir_1} < \tau_{ir_2})$, we can easily choose r_2 according to Eq.(2).

Case2: when the length of i to r_1 is smaller than i to $r_2(d_{ir_1} < d_{ir_2})$ and the pheromones of i to r_1 is less than i to r_2 $(\tau_{ir_1} > \tau_{ir_2})$, so it is difficult to judge which one of candidate path is best according to Eq.(2).

So, we propose a new operator: Unit Distance-Pheromone Operator (UDPO) as a single colony to simplify the process of path construction. Compared with Eq.(2), by increasing the weight of the distance factor and reducing the guidance of pheromone, UDPO has a good balance of these two factors.

The path construction equation is as follows:

$$\mathrm{r} = \max\left(\frac{\tau_{ir_1}}{d_{ir_1}}, \frac{\tau_{ir_2}}{d_{ir_2}} \cdots \frac{\tau_{ir_n}}{d_{ir_n}}\right) \tag{11}$$

where τ_{ir_n} is pheromone from city i to city r_n, d_{ir_n} is the distance between city i and city r_n.

Population2 uses the Eq.(12) to choose the path, which can enhance the ability of the algorithm to search the entire map, and combine the distance and pheromone factors to increase diversity of the PCCACO and make the algorithm more likely to jump out of the local optima.

$$S = \begin{cases} \max\left(\frac{\tau_{ir_1}}{d_{ir_1}}, \frac{\tau_{ir_2}}{d_{ir_2}} \cdots \frac{\tau_{ir_n}}{d_{ir_n}}\right) & q < q_0 \\ s & else \end{cases} \tag{12}$$

where s is Eq.(3), q_0 is a fixed value, q is a random number.

B. COMMUNICATION STRATEGY

We introduce Pearson correlation coefficient as the evaluation criterion. Two colonies with higher similarity are selected to reward the parameters of similar paths according to Eq.(1). $\mathrm{cov}(X, Y)$ is the covariance between two colonies, σ_X, σ_Y are standard deviations of two colonies, respectively in Eq.(1).

Every w iterations, the algorithm performs a judgment (the adaptive judgment mechanism will be introduced in 2) below. If it meets the conditions of communication, we selected the optimal 10 ants among all ants in each population to calculate the mean and variance of their path lengths, and bring them into the Eq.(1) for calculation. The number of ants in each colony is the same and written in Table1. Preliminary experiments show the reasonable number of optimal ants. When the correlation coefficient close to 1, it shows that the similarity between the two colonies is highest. According to Eq.(1), we can get the two most similar colonies and exchange information between them.

We have established a mathematical model to describe the communication process in similar colonies of this paper. $V^T_{population\,i}$ and $V^T_{population\,j}$ is matrix of the cities where all ants have passed in the *T-th* iteration in Population i and Population j. The matrix of the cities is as Eq.(13) and Eq. (14).

$$V^T_{population\,i} = \begin{bmatrix} n_1^{\mathrm{ant}1}, & n_2^{ant\,1} \cdots & n_n^{ant\,1} \\ n_1^{\mathrm{ant}2}, & n_2^{ant\,2} \cdots & n_n^{ant\,2} \\ \vdots & \vdots & \vdots \\ n_1^{\mathrm{ant\ m}}, & n_2^{ant\ m} \cdots & n_n^{ant\ m} \end{bmatrix} \tag{13}$$

TABLE 1. Parameters setting in each instance.

Instance	Number of ants	ACS parameters in PCCACO					MMAS parameters in PCCACO			
		α	β	ρ	ξ	q	α	β	ψ	q
berlin52	34	1	4	0.5	0.1	0.8	1	3	0.2	0.7
Eil51	31	1	4	0.5	0.1	0.8	1	3	0.2	0.7
Eil76	50	1	4	0.5	0.1	0.8	1	3	0.2	0.7
kroA100	66	1	4	0.5	0.1	0.8	1	3	0.2	0.7
Eil101	66	1	4	0.5	0.1	0.8	1	2	0.2	0.7
ch130	87	1	3	0.5	0.1	0.7	1	2	0.2	0.7
KroB150	100	1	3	0.4	0.1	0.7	1	2	0.2	0.7
KroA200	133	1	3	0.4	0.1	0.7	1	2	0.2	0.7
KroB200	133	1	3	0.4	0.1	0.7	1	2	0.2	0.7
tsp225	142	1	2	0.4	0.1	0.6	1	2	0.2	0.6
d198	132	1	3	0.4	0.1	0.7	1	2	0.2	0.7
lin318	212	1	3	0.4	0.1	0.7	1	2	0.2	0.7

$$V^T_{population\,j} = \begin{bmatrix} n_1^{\text{ant1}}, & n_2^{ant1} \cdots & n_n^{ant1} \\ n_1^{\text{ant2}}, & n_2^{ant2} \cdots & n_n^{ant2} \\ \vdots & \vdots & \vdots \\ n_1^{\text{ant m}}, & n_2^{ant\,m} \cdots & n_n^{ant\,m} \end{bmatrix} \quad (14)$$

where n is the number of cities, m is the number of ants.

We set k and h as the ants in two colonies that find the iteration-best solution in the *T-th* iteration. According to the Eq.(15), if the two colonies have three or more cities that are consecutively identical, the path is judged to be the common path.

$$S_{(i,j)} = V^T_{population\,i}[k] \cap V^T_{population\,j}[h] \quad (15)$$

where Population i and Population j represents two colonies, $S_{(i,j)}$ is the common path of two colonies. $V^T_{populatoni}[k]$ and $V^T_{population\,j}[h]$ is the path traveling by the *k-th* and *h-th* ants in the *T-th* iteration.

1) REWARD OPERATOR

When two colonies selected by Eq.(1) have the common path, we reward the parameters on this path. Reward can accelerate the convergence of the algorithm in the early stage. At the beginning of the algorithm, the parameters of α and β are the same between all cities.

$$\alpha_{ij}(t+1) = \begin{cases} \alpha_{ij}(t) + 0.02 & t \le 100 \\ \alpha_{ij}(t) + 0.1^{*t}/_{NC} & t > 100 \end{cases} \quad (16)$$

$$\beta_{ij}(t+1) = \begin{cases} \beta_{ij}(t) + 0.02 & t \le 100 \\ \beta_{ij}(t) + 0.2^{*t}/_{NC} & t > 100 \end{cases} \quad (17)$$

where t is the current number of iterations, and NC is the total number of iterations.

Here, when UDPO is selected as one of the similar colonies, the rest must be one of the ACS or MMAS, we only reward the path parameters of ACS or MMAS colonies. The main reasons are as follows: 1. UDPO has no reward parameters; 2. Controlling variables make the algorithm more robust and improve algorithm performance in each TSP instance. When the similar colonies are ACS and MMAS, we reward the parameters of common path in two populations.

2) ADAPTIVE ADJUSTMENT OF THE COMMUNICATION FREQUENCY

Reward is a positive feedback. If exchange of the information between two similar colonies is too frequent, it will make them same and harm variation. So, it is necessary to control the communication frequency. Therefore, dynamic feedback between the similar colonies is used to adaptively adjust the interaction interval and reduce the probability that the algorithm falls into local optima. Every w iterations, the algorithm will make a judgment. According to the Eq.(18), exchange probability P_{PCCACO} determines whether two similar colonies exchange information. It represents the proportion of the common path in the total path. When the exchange probability is greater than or equal to 0.5, it indicates that the diversity of PCCACO has been reduced, so the exchange between colonies will not proceed; otherwise communicate between the similar colonies. Therefore, the communication frequency of the entire PCCACO algorithm is shown in Eq.(18) and Eq.(19).

$$P_{PCCACO} = S_{(i,j)} \Big/ V^T_{population\,i}[\text{k}] \quad (18)$$

$$w = \begin{cases} 0 & P_{PCCACO} \ge \frac{1}{2} \\ 1 & P_{PCCACO} < \frac{1}{2} \end{cases} \quad (19)$$

where i, j are the two similar colonies, $S_{(i,j)}$ is the common path between two similar colonies, $V^T_{population\,i}[k]$ is the iteration-best path found by *k-th* ant in *T-th* iteration in population i. 0 and 1 is a binary variable. w equaling 1 means communication.

Although reward can speed up the convergence of the algorithm, it also reduces the algorithm's variation. So, adjusting the communication frequency adaptively according to Eq.(18) and Eq.(19) can ease the negative impact of rewards and improve the robustness of the algorithm. It also can balance the communication and learning mechanism between the two populations which makes the ants move steadily toward the optimal solution.

3) INITIALIZATION OF PARAMETERS

MMAS has such a high diversity partially because MMAS can reinitialize all pheromones when algorithm stagnates. We are inspired by this. When PCCACO falls into local optima and stagnate, the values of α and β are initialized to return to the initial value, so the influence of the parameters on the solution is weakened, and introduce a chance to obtain a better path.

C. CITIES-DROPOUT

Dropout is first proposed in neural networks. It mainly reduces the large number of calculations caused by too many nodes in the neural network and reduces the overfitting of the network. Overfitting is similar to the concept of local optima, so this article introduces the idea of dropout into ACO. Because of the excessive and slow calculation caused by so many cities in large-scale TSPs, this paper proposes a method of cities-dropout. In each iteration, combining with the accessible city set (*Allowed*), the number of computing cities required by each ant is greatly reduced. As shown in Fig 2, center point is the city where the ant is located. Drawing a circle with radius r in the center of the city, the cities in the *Allowed* are divided into two types: in the circle or out of the circle. The cities outside the circle are abandoned and their transfer probability is no longer calculated, which greatly reduces the number of cities calculated each time. Because the cities selected in this paper is large and concentrated, and the radius is relatively large, even if a city node outside the radius is calculated, it will not be selected as the next city. The method of cities-dropout does not affect the accuracy of

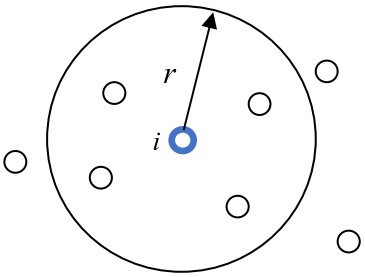

FIGURE 2. Current city and other cities. The radius of r_2 is used when the circle of radius r_1 has no optional city in the *Allowed* city table.

the solution, but can greatly reduce the running time of the program. Radius r follows Eq.(20) and (21):

$$r_1 = \frac{1}{3}\max(d_{i1}, d_{i2} \ldots\ldots d_{in}) + \min(d_{i1}, d_{i2} \ldots..d_{in}) \quad (20)$$

$$r_2 = \max(d_{i1}, d_{i2} \ldots\ldots d_{in}) \quad (21)$$

where i is the city where the ant is currently location, d_{in} is the distance between the *i-th* and *n-th* cities.

D. ALGORITHM FRAMEWORK

Algorithm 1 PCCACO Algorithm for TSP

1.Initialize the pheromone and the parameters
2.Calculate the distance between cities
3.While termination condition is not satisfied do
4. Construct ant solutions with cities-dropout
5. Update pheromones for MMAS, ACS, UDPO
6. Calculate the *P* earson correlation coefficient
7. Calculate P_{PCCACO} with Eq.(1)
8. If the communication condition is met (Eq.(19))
9. Find the common path using Eq.(13), Eq.(14) and Eq(15)
10. Reward the parameters with Eq(16) and Eq(17)
11. end-if
12. If (algorithm is stagnant)
13. Initialization the value of parameters
13. NC = NC + 1
14. END while

IV. EXPERIMENT AND SIMULATION

This experiment is simulated in the environment of MATLAB R2014a. In order to verify the performance of improved algorithm, we choose the various middle-large scale TSP instances, such as Eil76, ch130, KroB150, KroB200, Tsp225, lin318 and compare with ACS, MMAS algorithm and other multi-colonies Ant Colony algorithms. Each algorithm is performed 10 times and the number of ants, heuristic values are adjusted in various TSP instance. The parameters used in this paper are shown in Table 1.

Section A shows a comparative analysis of PCCACO, ACS, MMAS and the distribution of common paths in some TSP instances. Experiments show that PCCACO is better than the other two algorithms both in convergence speed and optimal solution. In the early stage, PCCACO can converge quickly, and can also jump out of local optima in the later stage, thus improving the quality of the solution.

Section B shows the comparison experiment between PCCACO algorithm and other multi-colonies Ant Colony Optimization algorithms. The experimental data of the algorithm come from the corresponding papers. Comparisons show that the optimal solution in this paper is better than the other multi-colonies ant algorithm in some TSP instances.

Section C shows the impact of the cities-dropout on the experimental running time. Experiments have shown that cities-dropout can significantly reduce the runtime of algorithm.

A. COMPARISON OF THREE ALGORITHMS

In order to compare the performance of MMAS, ACS with PCCACO, this paper selects middle-large scale TSP instances for analysis, for instance ch130, KroB150, KroA200, KroB200, Tsp225, lin318. The experimental analysis is carried out from several directions, for example Optimal solution (Opt), Average solution (Mean), Error rate, and convergence iteration (convergence). Experimental data is in Table2. We use Eq.(22) to measure the difference between each ACO and the optimal solution of the TSP instances.

$$Error\ rate = (L_{avg} - L_{\min}/L_{\min}) \times 100\% \quad (22)$$

where L_{avg} is the average cost of the best tour lengths, $L_{\min}$ is the optimal solution for each instance.

It can be seen from the results in Table 2 that PCCACO is superior to ACS and MMAS in all scale TSP instances in all metrics. On the small and medium scale instance (like Eil51 instance, Fig. (3)), in the early period, PCCACO has the fastest convergence speed compared with other ACO algorithms maybe due to the exchange of information and its reward strategy among populations. The optimal solution in the TSP instance was found after 53 iterations by PCCACO, while the ACS algorithm and MMAS algorithm did not find the optimal solution. Because MMAS reinitializes all pheromones when algorithm is stagnant, so the solution it found is better than ACS, but they all fall into local optima. For large scale TSP instance (such as the lin318 instance, Fig. (3)), due to the large scale of the city, the optimal solution is difficult to find, PCCACO has a slight better convergence than ACS, and MMAS is the slowest. Due to the adaptive communication frequency and the initialization of parameters, PCCACO continuously jumps out of local optima and increases the quality of the solution.

1) COMMON PATH REWARD

The red and bold black lines in Fig.5 and Fig.6 are the paths that have been rewarded during all the iterations. Black is the path with more rewards and the red part is the rewarded path. It can be seen from the figures that the cities with bold

TABLE 2. Performance comparison of PCCACO, MMAS, ACS in different TSP instances.

Instance	Opt	ACO	Best	Worst	Mean	Error rate	Convergence
Eil51	426	PCCACO	**426**	**430**	**427**	**0.23**	**53**
		MMAS	428	436	431	1.41	812
		ACS	429	437	430	0.93	634
berlin52	7542	PCCACO	**7542**	**7542**	**7542**	**0**	**58**
		MMAS	7542	7542	7542	0	200
		ACS	7542	7542	7542	0	304
Eil76	538	PCCACO	**538**	**543**	**539**	**0.18**	317
		MMAS	539	549	542	0.74	434
		ACS	540	547	543	0.93	**205**
KroA100	21282	PCCACO	**21282**	**21651**	**21383**	**0.47**	**148**
		MMAS	21291	22051	21516	1.10	804
		ACS	21304	22214	21558	1.30	198
Eil101	629	PCCACO	**629**	**637**	**631**	**0.31**	**153**
		MMAS	631	643	634	0.79	719
		ACS	632	639	636	1.11	392
ch130	6110	PCCACO	**6129**	**6215**	**6173**	**1.03**	**327**
		MMAS	6176	6331	6277	2.73	879
		ACS	6221	6378	6291	2.96	1874
KroA150	26524	PCCACO	**26654**	**27136**	**26715**	**0.72**	**458**
		MMAS	26746	27813	27251	2.74	1254
		ACS	26793	28014	26938	1.56	1029
KroB150	26130	PCCACO	**26130**	**26732**	**26241**	**0.42**	**526**
		MMAS	26176	27283	26562	1.65	1678
		ACS	26198	27164	26487	1.37	1275
KroA200	29368	PCCACO	**29391**	**29806**	**29485**	**0.40**	448
		MMAS	29495	31894	30435	3.63	1120
		ACS	29561	32483	30732	4.64	**367**
KroB200	29437	PCCACO	**29541**	**30483**	**29878**	**1.50**	**428**
		MMAS	29723	32604	31078	5.57	1117
		ACS	29819	32811	31489	6.97	734
d198	15780	PCCACO	**15814**	**18522**	**16463**	**4.32**	**433**
		MMAS	15915	19583	17605	11.5	1634
		ACS	15943	18626	17382	10.15	843
Tsp225	3916	PCCACO	**3937**	**4034**	**3981**	**1.65**	531
		MMAS	4025	4119	4063	3.75	849
		ACS	3978	4073	4025	2.78	**417**
lin318	42029	PCCACO	**42461**	43957	**42933**	**2.15**	**582**
		MMAS	42800	**43792**	43049	2.43	1478
		ACS	42936	44018	43310	3.04	734

black parts are more dispersed, and the results of transition probability are much different. It can be considered that there are fewer cities for ants to choose; while the red part has a high density of distribution within a specific range, and ants are easy to fall into local optima. Both of the solution found in Fig.5 and Fig.6 are consistent with the solution given in the TSP instances. So, it proves that the common path we described above have a higher probability of being part of the optimal solution.

2) THEORETICAL JUSTIFICATION

According to Eq.2, Eq.16 and Eq.17, the larger the values of α and β between two cities, the easier it be selected. Parameters have positive feedback on the algorithm. On the small and medium scale instance (like Eil51, berlin52), because the number of cities is small, the optimal solution is easier to find, the algorithm is not easy to fall into local optima and reward of the common path makes the algorithm converge quickly. For large-scale TSP instances such as Tsp225, lin318, etc.,

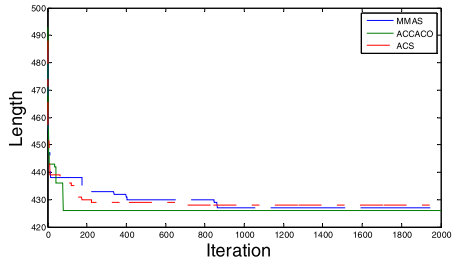

FIGURE 3. Convergence curves in the Eil51 instance.

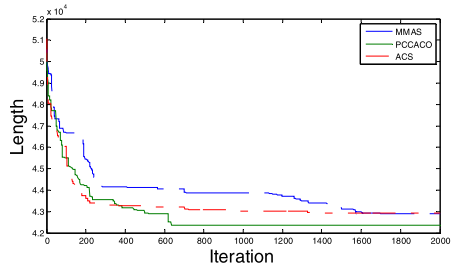

FIGURE 4. Convergence curve of the lin318 instance.

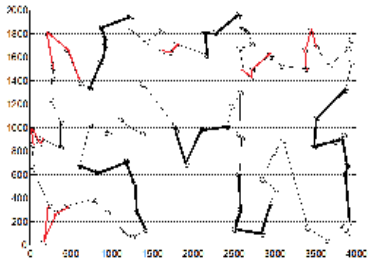

FIGURE 5. Rewarded common path of KroA100.

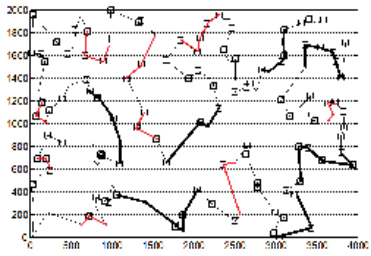

FIGURE 6. Rewarded common path of KroB150.

due to more cities, the optimal solution is more difficult to be found and the fast convergence brought by the reward will reduce the accuracy of the algorithm. So adaptive communication is beneficial to maintain the diversity of the algorithm. If algorithm still can't jump out of the local optima, the values of α and β are initialized to return to the initial value. This method reduces the impact of parameters on the results.

3) OPTIMAL SOLUTION

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

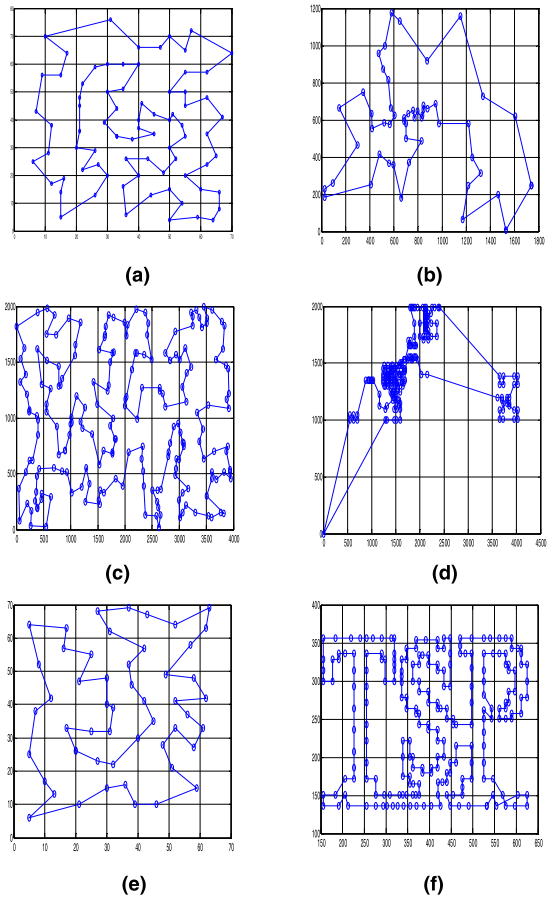

FIGURE 7. Best tours for each TSP instance found by PCCACO. (a) Eil76. (b) berlin52. (c) KroB200. (d) d198. (e) Eil51. (f) Tsp225.

B. COMPARISON WITH OTHER MUTIL- COLONIES ANT COLONY ALGORITHMS

The improved algorithm in this paper is also compared with other multi-colonies Ant Colony Optimization algorithms. Table 3 shows the comparison of PCCACO with other multi-colonies ant colony algorithms. Data reference to its corresponding literature. The number in parentheses after the algorithm name indicates the position in the reference. *Avg* is the average tour length and *Error*(%) is the relative deviation.

As can be seen from Table 3, PCCACO outperforms other algorithms in terms of the average and error rate in some TSP instances. Especially on the berlin52 and Eil76, the average solution of the 15 experiments is very close to the optimal solution of the TSP instances, and the error rate is very low.

C. CITIES-DROPOUT

1) ABLATION EXPERIMENT ON CITIES-DROPOUT

The cities-dropout method proposed in this paper can significantly reduce the total time of program operation on large-scale TSP instances. This paper sets up an ablation experiment. For the PCCACO algorithm, a method uses

TABLE 3. Comparison with other mutil-colonies ant colony algorithms.

ACO		Eil51	berlin52	Eil76	KroA100	Eil101	KroA200
PCCACO	Avg	426.73	**7542**	**538.25**	21419.33	631.52	**29485.29**
	Error(%)	0.23	**0.00**	**0.04**	0.47	0.31	**0.40**
PACO-3Opt	Avg	426.35	**7542**	539.85	21326.80	**630.55**	29644.50
(2018) [9]	Error(%)	0.08	**0.00**	0.34	0.21	**0.25**	0.94
HHACO	Avg	429.41	-	541.84	21463.39	-	-
(2017) [18]	Error(%)	0.70	-	0.58	0.85	-	-
PSO-ACO-3Opt	Avg	**426.45**	7543.20	**538.30**	21445.10	632.70	29646.05
(2015) [8]	Error(%)	**0.11**	0.02	**0.06**	0.77	0.59	0.95
ACO with ABC	Avg	443.39	7543.37	557.98	22,435.31	683.39	-
(2015) [23]	Error(%)	4.08	0.03	3.71	5.42	8.65	-
WFA with 3-Opt	Avg	426.65	**7542**	541.22	**21282.80**	633.50	29646.50
(2013) [15]	Error(%)	0.14	**0.00**	0.60	**0.00**	0.72	0.95
IVRS + 2opt	Avg	431.10	7547.23	-	21,498.61	648.67	-
(2012) [26]	Error(%)	1.20	0.07	-	1.02	3.13	-
CGAS	Avg	-	7634.00	542.00	21,437.00	-	29,946.00
(2015) [25]	Error(%)	-	1.24	0.74	0.73	-	1.97
ACO + 2opt	Avg	439.20	7556.58	-	23441.80	672.67	-
(2012) [26]	Error(%)	3.11	0.19	-	10.15	6.90	-
ACO with Tagushi	Avg	435.40	7635.40	565.50	21,567.10	655.00	-
(2013) [27]	Error(%)	2.21	1.24	5.11	1.34	4.13	-
SA ACO PSO	Avg	427.27	7542	540.20	21,370.30	635.23	29738.73
(2011) [28]	Error(%)	0.30	0.00	0.41	0.41	0.99	1.26
HACO	Avg	431.20	7560.54	-	-	-	-
(2012) [29]	Error(%)	1.22	0.19	-	-	-	-

TABLE 4. Running time comparison.

Instance	NC	Ant number	PCCACO-1 t/s	PCCACO-2 t/s
Eil51	2000	32	**1328.59**	1700.49
berlin52	2000	34	**743.12**	1083.45
Ch130	2000	73	**3731.44**	4339.26
KroB150	2000	100	**6371.48**	8619.62
KroA200	2000	133	**7213.32**	9490.44
KroB200	2000	133	**7844.51**	11058.39
lin318	2000	212	**19038.33**	24019.45
d198	2000	132	**7637.71**	9782.26

cities-dropout, we call it PCCACO-1; another does not use cities-dropout method, we call it PCCACO-2. Table 4 shows the running time difference between PCCACO-1 and PCCACO-2 in different TSP instances. *NC* is the maximum number of iterations.

Here we have to explain that different programming languages and different models of CPU or GPU will cause differences in program runtime. Since our program is conducted in MATLAB, its running time is significantly longer than C++. However, using cities-dropout in same situation reduce running time of the program. Table 4 shows that as the number of cities in the TSP instances increases, the running time increases. In same TSP instances, PCCACO-1 runs significantly much quickly than PCCACO-2 due to the reduction in the number of calculated cities.

2) EFFECTS OF DIFFERENT RADII OF CITIES-DROPOUT

We also want to verify the effect of different cities-dropout radii on TSP instances, but because the TSP instances are different and the Ant Colony Optimization is a probabilistic algorithm, the result of the solution is not fixed every time, so this work is very difficult. We can take the limit of the radius, when the radii equal $\min(d_{i1}, d_{i2} \ldots . d_{in})$, it can be seen as a greedy approach, and the experimental results are very bad. When the radii equal $\max(d_{i1}, d_{i2} \ldots . d_{in})$, it means cities-drop unused. During this time interval, we tried several different radii of cities-dropout. It can be concluded that when the radius is small, the running time of the program will be shortened; when the radius becomes larger, the running time of the program will gradually increase. However, it is difficult to say how large the radius is the best because the paths of all optimization algorithms are probabilistic choices and are related to the distribution of different TSP instances. But the radii selected in this paper are relatively large, and TSP instances we experiment are medium-large scales, which the cities are very dense, therefore it will not affect the construction of the solution.

V. CONCLUSION

In this paper, a new path construction operator is proposed as a single population, which is combined with MMAS and ACS to form a three-colonies Ant Colony Optimization algorithm. The algorithm introduces the Pearson correlation coefficient as the evaluation criterion, selecting two colonies with higher similarity, and rewarding the parameters of the common path in the two colonies to speed up the convergence. According to the dynamic feedback of the diversity among colonies, we can adaptively control the frequency of information exchange between two colonies to ensure the ant colony algorithm to find better solutions. When the algorithm stagnates, the parameters of the reward are reinitialized and make the algorithm more easily jump out of the local optima. Finally, the concept of dropout in neural network is introduced, and an idea of cities-dropout is proposed in path construction, which reduces the running time of the algorithm. The experimental results show that the PCCACO proposed in this paper has better superiority on TSP than the traditional single-population ant colony algorithm and other multi-population ant colony optimization.

As future work, we will continue to study the impact of parameters on the experiment and use a lot of experiments to find out how large the radius is the best.

APPENDIX

See Table 5.

TABLE 5. Abbreviations of algorithm.

Algorithm	Abbreviation
Particle Swarm Optimization	PSO
Genetic Algorithm	GA
Simulated Annealing	SA
Ant System	AS
Ant Colony System	ACS
MAX-MIN Ant System	MMAS
Artificial Bee Colony	ABC
Rank-based Ant System	(RbAS)
Unit Distance-Pheromone Operator	UDPO
Invasive Weed Optimization	IWO
Water Flow-like Algorithm	WFA
2-Optimization	2-Opt
3-Optimization	3-Opt

ACKNOWLEDGEMENT

The author would like to thank his seniors Mingle Xu and Jia Chen, who had given him some advice to make the article be much better.

REFERENCES

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. Int. Conf. Neural Netw.*, Dec. 1995, pp. 1942–1948.
[2] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Boston, MA, USA: Kluwer, 1989.
[3] S. Kirkpatrick and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1987.
[4] M. Dorigo, V. Maniezzo, and A. Colorni, "The 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.
[5] 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.
[6] T. Stützle and H. H. Hoos, "MAX-MIN ant system," *Future Generat. Comput. Syst.*, vol. 16, no. 8, pp. 889–914, Jun. 2000.
[7] D. Gomez-Cabrero and D. N. Ranasinghe. (2018). "Fine-tuning the ant colony system algorithm through particle swarm optimization." [Online]. Available: https://arxiv.org/abs/1803.08353
[8] M. Mahi, 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.
[9] G. aban, M. Mostafa, B. O. Kaan, 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.
[10] H. Chen *et al.*, "Ant colony optimization with tabu table to solve TSP problem," in *Proc. 37th Chin. Control Conf. (CCC)*, Apr. 2018, pp. 2523–2527.
[11] D. S. Deif and Y. Gadallah, "An ant colony optimization approach for the deployment of reliable wireless sensor networks," *IEEE Access*, vol. 5, pp. 10744–10756, 2017.
[12] I. Ellabib, P. Calamai, and O. Basir, "Exchange strategies for multiple ant colony system," *Inf. Sci.*, vol. 177, no. 5, pp. 1248–1264, Mar. 2007.
[13] 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.
[14] Y. Zhou *et al.*, "A discrete invasive weed optimization algorithm for solving traveling salesman problem," *Neurocomputing*, vol. 151, no. 3, pp. 1227–1236, Mar. 2015.
[15] Z. A. Othman *et al.*, "Performance water flow-like algorithm for TSP by improving its local search," *Int. J. Advancements Comput. Technol.*, vol. 5, no. 14, p. 126, Oct. 2013.
[16] M. D. Toksari, "A hybrid algorithm of ant colony optimization (ACO) and iterated local search (ILS) for estimating electricity domestic consumption: Case of Turkey," *Int. J. Electr. Power Energy Syst.*, vol. 78, pp. 776–782, Jun. 2016.
[17] C. Twomey, T. Stützle, M. Dorigo, M. Manfrin, and M. Birattari, "An analysis of communication policies for homogeneous multi-colony ACO algorithms," *Inf. Sci.*, vol. 180, no. 12, pp. 2390–2404, Jun. 2010.
[18] M.-L. Xu, X.-M. You, and S. Liu, "A novel heuristic communication heterogeneous dual population ant colony optimization algorithm," *IEEE Access*, vol. 5, pp. 18506–18515, 2017.
[19] S. C. Chu, J. F. Roddick, and J. S. Pan, "Ant colony system with communication strategies," *Inf. Sci.*, vol. 167, nos. 1–4, pp. 63–76, Dec. 2014.
[20] L. Wang and J. Shen, "Multi-phase ant colony system for multi-party data-intensive service provision," *IEEE Trans. Services Comput.*, vol. 9, no. 2, pp. 264–276, Sep. 2016.
[21] 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.
[22] E. Liao and C. Liu, "A hierarchical algorithm based on density peaks clustering and ant colony optimization for traveling salesman problem," *IEEE Access*, vol. 6, pp. 38921–38933, 2018.
[23] M. Gunduz and M. S. Kiran, and E. Özceylan, "A hierarchic approach based on swarm intelligence to solve the traveling salesman problem," *Mathematics*, vol. 23, pp. 215–235, Mar. 2015.
[24] D. Karaboga and B. Gorkemli, "A combinatorial artificial bee colony algorithm for traveling salesman problem," in *Proc. Int. Symp. Innov. Intell. Syst. Appl. (INISTA)*, Apr. 2011, pp. 50–53.
[25] G. Dong, W. W. Guo, and K. Tickle, "Solving the traveling salesman problem using cooperative genetic ant systems," *Expert Syst. Appl.*, vol. 39, no. 5, pp. 5006–5011, Apr. 2012.
[26] 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, Apr. 2012.
[27] M. Peker, B. en, and P. Y. Kumru, "An efficient solving of the traveling salesman problem: the ant colony system having parameters optimized by the Taguchi method," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 21, pp. 2015–2036, May 2013.
[28] S.-M. Chen and C.-Y. Chien, "Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques," *Expert Syst. Appl.*, vol. 38, no. 12, pp. 14439–14450, 2011.

[29] W. Junqiang and O. Aijia, "A hybrid algorithm of ACO and delete-cross method for TSP," in *Proc. Int. Conf. Ind. Control Electron. Eng.*, Jun. 2012, pp. 1694–1696.
[30] J. H. Zhao, Z. Liu, and M. T. Daoa, "Reliability optimization using multiobjective ant colony system approaches," *Rel. Eng. Syst. Saf.*, vol. 92, pp. 109–120, 2017.
[31] J. Benesty *et al.*, "Pearson correlation coefficient," *Noise Reduction Speech Process.*, vol. 1, pp. 1–4, Sep. 2009.
[32] J. Adler and I. Parmryd, "Quantifying colocalization by correlation: The Pearson correlation coefficient is superior to the Mander's overlap coefficient," *Cytometry A*, vol. 77, no. 8, pp. 733–742, Aug. 2010.

HONGWEI ZHU was born in 1993. 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.

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

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