Arabian Journal for Science and Engineering (2022) 47:1657–1674
https://doi.org/10.1007/s13369-021-06033-4

RESEARCH ARTICLE-COMPUTER ENGINEERING AND COMPUTER SCIENCE

Multi-Colony Ant Optimization Based on Pheromone Fusion Mechanism of Cooperative Game

Yadong Mo[1] · Xiaoming You[1] · Sheng Liu[2]

Received: 16 April 2021 / Accepted: 28 July 2021 / Published online: 10 August 2021

Abstract

In this paper, we propose a multi-colony ant optimization based on pheromone fusion mechanism of cooperative game (CGMACO) to balance the convergence speed and diversity of the algorithm. Firstly, the heterogeneous multi-colony is composed of ant colony system (ACS) and Max–Min ant system (MMAS), and these two classical colonies coordinate together to improve the solution quality. Secondly, the cooperative game model determines which sub-colonies can interact with each other based on evaluating each union's payoff, while the pheromone fusion mechanism decides what information can exchange by regulating the pheromone matrix of each subpopulation. Those two methods can greatly diversify the solution of algorithm. In addition, the information entropy is also introduced to control the interaction frequency, which enhances the adaptability of the algorithm. Finally, the experimental results of the large-scale TSP instances show that the improved algorithm can improve the accuracy of the solution without affecting the convergence speed and better than the existing intelligent algorithms.

Keywords Ant colony algorithm · Cooperative game · Pheromone fusion · Information entropy · TSP

1 Introduction

Travelling salesman problem (TSP) is a classical combinatorial optimization problem. The problem can be mainly described as follows: a travelling merchant traverses all the cities of the country without repeating and finally comes back to the start point. The loop obtained by the salesman is required to be shortest, so-called the minimum Hamiltonian circuit. There are many methods to solve TSP, such as genetic algorithm [1], particle swarm optimization [2], grey wolf optimization algorithm [3], ant colony algorithm [4, 5]. And current research shows that ant colony algorithm can solve the TSP problem well.

✉ Xiaoming You
yxm6301@163.com

Yadong Mo
529634794@qq.com

Sheng Liu
ls6601@163.com

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

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

Ant colony optimization (ACO) is a classical swarm intelligence algorithm proposed by Italian scholar M. Dorigo who was inspired by ants foraging in nature [4]. The main idea is that ants can use their own pheromone updating mechanism to effectively return and forth between food source and nest. After the ant system algorithm was proposed, it attracted much attention and brought largely discussion about the improvement of the algorithm. In order to improve the solution accuracy of the ant system algorithm, Dorigo [5] also put forward the ant colony system algorithm (**ACS**). In ACS, only the global optimal ant can be allowed to deposit pheromone in each iteration, and other ants will diminish the level of pheromone on the tour they visited in terms of local pheromone updating rule. This mechanism can strengthen the positive feedback effect of the optimal information and speed up the convergence of the algorithm. However, it also makes the algorithm easily fall into local optimum. In order to overcome this problem, Stützle et al. [6] proposed the Max–Min ant system algorithm (**MMAS**). MMAS restricts the accumulation and volatilization of pheromone by limiting it within a fixed interval, which can avoid algorithm stagnation to some certain extent. Thus, the population diversity can be improved. However, the algorithm will be difficult to converge when the solutions distribute dispersedly.

Although these improved references have acquired some achievements in the traditional ant colony algorithms, the main problem, how to balance the relationship between the convergence speed and diversity, has not been solved well. Therefore, more scholars try to solve it based on their research fields. Sangeetha et al. [7] used a pheromone enhancement mechanism to increase the pheromone concentration on the better path that reduced useless search and saved the time cost. Ye et al. [8] introduced the negative feedback pheromone strategy to guide ant colony search for unknown space to avoid too many ants selecting the same area, which can expand the search space and enhance the diversity of the algorithm. Ning et al. [9] put forward a pheromone update mechanism based on the current optimal path. The method increased the pheromone value of different paths between the best-so-far optimal path and the current optimal path, which can speed up the convergence of the algorithm. Tseng et al. [10] divided the ant colony into two groups and the cooperation between two kinds of ants improved the accuracy of the solution. Besides, the parameter setting is another conundrum for the ACO. To solve this problem, Mahi et al. [2] applied the particle swarm optimization algorithm to optimize the parameters of the ACO, which improved the stability of the algorithm. Olivas et al. [11] introduced the fuzzy control system to select the appropriate parameters for the ACO algorithm and enhanced the accuracy of the solution. Tuani et al. [12] proposed a novel adaptive parameter adjustment mechanism to improve the adaptability of the algorithm. In addition, other improved ACO algorithms have been widely used in various fields, such as robot path planning problem [13], network routing problem [14], image detection [15], vehicle scheduling problem [16], data mining [17].

However, due to the limitation of the single population, the improvements often weaken one characteristic of the colony to strengthen another. For example, it will increase the search time to expand more areas or will diminish the solution accuracy to accelerate the convergence. In order to balance the relationship between the convergence speed and diversity of the algorithm further, the multi-population gradually attracts many scholars' attention. Gambardella [18] proposed the concept of the multi-ant colony algorithm for the first time. They adopted two colonies of ACS to solve vehicle scheduling problems with time window. Chu et al. [19] proposed seven interaction strategies to control the communication among the homogenous colonies. Twomey et al. [20] analyzed the homogenous multi-ant colony with different communication policies and proposed migrant integration strategy for the interaction. The cooperation on homogenous populations will only amplify the single feature in terms of their same characteristics, while the heterogeneous populations can take full advantage of each other. Dong et al. [21] combined the ant colony algorithm with genetic algorithm in a novel way to solve the TSPs successfully. Zhang et al. [22] used two heterogeneous ant colonies to diversify the solution of algorithm by exchanging the pheromone information. Wang et al. [23] applied multi-ant algorithm with local search to solve the vehicle routing problem, which enhanced the solution accuracy by comparing and exchanging the global optimal solution of each colony.

According to the above references, multi-colony algorithms can balance the convergence speed and search abilities of the ACO better than the single colony algorithms. However, the interaction mechanism among sub-colonies is relatively simple, which leads to the adaptability of multi-colony algorithm underperformance. To deal with these issues, some cross-discipline methods, such as game theory or information theory, are applied to improve the performance of the multi-colony algorithm. Yang et al. [24] introduced game theory to control the coordination among heterogenous populations and improved the stability of the algorithm. Li et al. [25] applied the information entropy to adapt the communication among populations more accurately. In this paper, we focus on balancing the relationship between the convergence speed and the diversity of the algorithm. And from the above theories, the multi-ant colony algorithm based on pheromone fusion mechanism of the cooperative game is proposed to solve large-scale TSP instances. The main contributions and innovations of this research are as follows.

Firstly, the pheromone fusion mechanism that regulates the pheromone distribution of each sub-colony is introduced to realize the information exchange among multiple populations effectively. It fuses the pheromone matrix of other subpopulations while remains the original population information, which improves the efficiency of communication. Thus, the diversity of the algorithm is enhanced.

Secondly, the cooperative game model is proposed to help the population select appropriate communication objects by finding the Pareto optimal combination. If the Pareto optimal combination belongs to the cooperative union, the profit distribution strategy will be applied, otherwise, the pheromone smoothing mechanism can be triggered. In the profit distribution strategy, the profits will be distributed into members reasonably by adding the pheromone on the public paths among populations to accelerate the convergence speed of the algorithm. In the pheromone smoothing mechanism, the pheromone matrix will be reinitialized to help the algorithm jump out of the local optimum effectively.

Finally, the information entropy is introduced to control the communication frequency, which is called adaptive communication strategy. In this strategy, the information entropy is used to evaluate the diversity of the population, and we control the communication frequency among populations by measuring their information entropy state to improve the adaptability of the algorithm.

In addition, the contents of this paper are as follows: Sect. 2 introduces ACS, MMAS algorithm and information entropy briefly. Section 3 reports he working principle of CGMACO, including adaptive communication strategy, pheromone fusion mechanism and cooperative game model. Section 4 analyses the performance of proposed algorithm with different strategies and compares CGMACO with the traditional ant colony algorithm and other intelligent algorithms. Section 5 summarizes and prospects this research.

2 Related Work

2.1 Ant Colony System

Ant colony system (ACS) was proposed by Italian scholar M. Dorigo in 1996 [5]. It introduced a special state transition rule called pseudo-random proportionality rule. This rule allows each ant to choose its path in a roulette mode with a probability of 1-q0, where q0 is a parameter between [0,1]. The state transition formula is as follows.

$$P^k{}_{ij} = \begin{cases} \frac{\tau^{\alpha}{}_{ij} \cdot \eta_{ij}{}^{\beta}}{\sum_{l \in \text{allowed}} \tau^{\alpha}{}_{il} \cdot \eta_{il}{}^{\beta}} & \text{if j} \in \text{allowed}^k \\ 0 & \text{else} \end{cases} \tag{1}$$

where the τ_{ij} represents the pheromone value from city i to city j (vertex (i, j)); $\eta_{ij} = 1/d_{ij}$ denotes the heuristic information on the vertex (i, j) which d_{ij} is the cost of the vertex (i, j); *allowed* stores a set of cities that the ant k is not visit; α and β are two weight parameters determining the confluence of pheromone value and heuristic information. In addition, $S = argmax(\tau_{ij} \cdot \eta_{ij}{}^{\beta})$ is another transition formula that the ants positioned city i move to city j when the random number q is lower than q0.

After finishing a transition from city i to city j, each ant applies a local pheromone update rule to decrease the attraction of the edge (i, j). The formula is as follows:

$$\tau_{ij} \leftarrow (1-\xi) \cdot \tau_{ij} + \xi \cdot \tau_0 \tag{2}$$

where $0 < \xi < 1$ is a pheromone evaporation rate; $\tau_0 = 1/(n \cdot l_n)$ is the initial pheromone level, where n is the city number and l_n is the tour length created by the nearest neighbour heuristic algorithm.

When all ants complete tour construction, only the globally best tour is allowed to add pheromone which called global pheromone updating rule. The formula is written as:

$$\tau_{ij} \leftarrow (1-\rho) \cdot \tau_{ij} + \rho \cdot \Delta\tau^{\text{bs}}{}_{ij} \tag{3}$$

where $0 < \rho < 1$ is pheromone evaporation rate; $\Delta\tau^{\text{bs}}{}_{ij} = 1/L_{gb}$ is the number of increasing pheromone, and L_{gb} is the length of the best tour.

2.2 Max–Min Ant system

In order to solve the easy stagnation of the traditional ant colony algorithm, Stützle proposed the Max–Min ant system (MMAS) [6]. In MMAS, the pheromone is updated by alternating iteration-best tour with best-so-far tour in the early run time, and the updating rule is defined as same as formula (3). What's more, the pheromone trail of each path is limited to the specified range $[\tau_{\min}, \tau_{\max}]$. If $\tau_{ij} < \tau_{\min}$, then $\tau_{ij} = \tau_{\min}$; if $\tau_{ij} > \tau_{\max}$, then $\tau_{ij} = \tau_{\max}$. And it also reinitializes the pheromone matrix to avoid the stagnation. The maximum and minimum values of pheromones are set as follows:

$$\tau_{\max} = (1/\rho) \cdot (1/L_{gb}) \tag{4}$$

$$\tau_{\min} = \tau_{\max}/(2n) \tag{5}$$

where n is the city number; L_{gb} is the length of the iteration-best tour.

2.3 Information Entropy

Entropy was originally used to measure the disorder state of thermodynamic system in the field of physics. Later, American scholar Shannon introduced it into the field of information theory and put forward the conception of information entropy. For now, information entropy has not only improved greatly in theory [26–28], but also achieved good results in many practical applications [29–31]. And it shows the effectiveness of information entropy as a measure of discrete system. The formula is as follows:

$$E(P) = -\sum_{x \in X} P(x) \log p(x) \tag{6}$$

where **X** is the solution of the problem, and **P (x)** is the probability of x, and $\sum_{x \in X} P(x) = 1$.

3 Proposed Algorithm

In this research, a heterogeneous multi-colony ant colony algorithm based on cooperative game theory is proposed to balance the convergence and the diversity of the algorithm. Based on regarding each subpopulation as an independent agent and the premise of individual rationality, cooperative evolution is realized by game decision mechanism among subpopulations and Fig. 1 shows the interactive model.

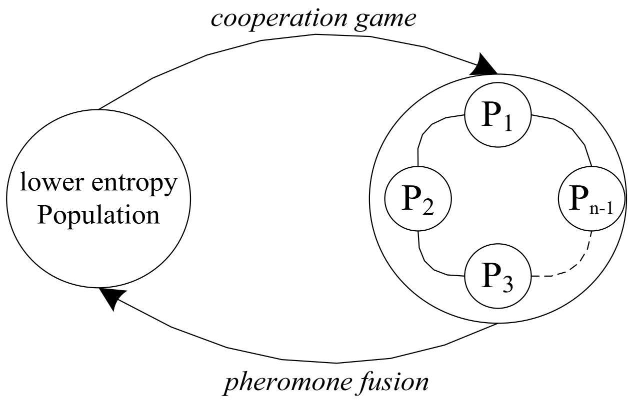


Fig. 1 Dynamic interactive game model

And this part is organized as follows. Section 3.1 introduces the self-adaptive communication strategy based on information entropy. Section 3.2 provides the pheromone fusion strategy in detail. Section 3.3 is dedicated to applying the cooperative game theory to the multi-ACO algorithm. Section 3.4 is the algorithm description.

3.1 Self-adaptive Communication Strategy

For multi-colony algorithm, it is necessary to have an appropriate method to control the interaction frequency of various subpopulations. With the running of algorithm, there will be more and more unbalanced distribution among various pheromone matrices. Thus, interaction with each other is needed to increase the diversity. According to the part 2.3, the information entropy is applied to measure the diversity of the algorithm so as to control the communication frequency more accurately, which the formula is as follows:

$$P_i(t) = n/m \tag{7}$$

$$E(Pt) = -\sum_{i=1}^{m} P_i(t) log p_i(t) \tag{8}$$

where $\boldsymbol{P_i(t)}$ is the proportion that the i-th tour trail selected by $\boldsymbol{n}$ ants when $\boldsymbol{M}$ ants generate $\boldsymbol{m}$ paths in this iteration, $\boldsymbol{E(Pt)}$ is the information entropy of the population in t-th iteration which demonstrates that if the tour difference of the population is higher, the information entropy of the population will be larger, and vice versa. In another word, the higher information entropy, the better diversity of the population. Therefore, by comparing the information entropy of each subpopulation with the set threshold, the communication frequency among populations can be controlled more accurately, and thus, the adaptability of interaction is improved.

Table 1 The pheromone matric under cooperative game

	C (P_j)	D (P_j)
C (P_k)	Ph_{11}	Ph_{12}
D (P_k)	Ph_{21}	Ph_{22}

3.2 Pheromone Fusion Strategy

In this part, the pheromone fusion strategy is proposed to increase the diversity of the population when the population falls into local optimum. If $E(p_i) < E^*(P)$, where the $\boldsymbol{E^*(P)}$ is the threshold parameter, we will consider that the diversity of this population is poor. Under this circumstance, communicating with other populations is needed to be adopted. In this paper, communication strategy is the pheromone fusion which is realized by using the weighted coefficients of information entropy. The formula is as follows:

$$Ph_i \leftarrow (1 - \sum_{j \neq i}^{n-1} w_j) \cdot Ph_i + \sum_{j \neq i}^{n-1} w_j \cdot Ph_j \tag{9}$$

where $\boldsymbol{Ph_i}$ is the pheromone matrix of population i, $\boldsymbol{Ph_j}$ is the pheromone matrix of population j; $\boldsymbol{W_j}$ is the pheromone contribution of population j to population i, which the formula can be written as follows:

$$w_j = \frac{E(p_j)}{\sum_{j=1}^{n} E(p_j)} \tag{10}$$

where $\boldsymbol{E(P_j)}$ is the information entropy of the population j.

3.3 Cooperative Game Model

3.3.1 Cooperative Game in Multi-ACO

In order to improve the performance of the pheromone fusion mechanism proposed above further and make full use of the heterogeneous populations, we build the cooperative game model based on cooperative game theory. In the model, two decisions that cooperation (C) or defection (D) allows subpopulations to select when they receive signals that need to communicate. In this paper, the cooperation and defection rules are given by (11)

$$w_j = \begin{cases} w_j \textit{ if } p_j \textit{ is cooperation} \\ 0 \;\; \textit{if } p_j \textit{ is defection} \end{cases} \tag{11}$$

The formula (11) illustrates that if the population j chooses to participate in cooperation (C), the weight coefficient is $\boldsymbol{w_j}$; if the population j chooses to defection (D), the weight coefficient $\boldsymbol{w_j}$ is 0. And the different pheromone matrices obtained by those selections are shown in Table 1. Therefore, there are three types of alliance structure including that full union structure which all colonies cooperate such as $\boldsymbol{Ph_{11}}$ in Table 1, sub-union structure which not all colonies are cooperative such as $\boldsymbol{Ph_{12}}$ and $\boldsymbol{Ph_{21}}$ in Table 1 and single-player structure which all colonies are not cooperative such as $\boldsymbol{Ph_{22}}$ in Table 1.

In addition, there are three basic parts of the game theory including players, strategy sets and the payoff of each strategy. In this paper, the players are the subpopulations, and the pheromone matrices in Table 1 denote the strategy sets. Meanwhile, we define $\boldsymbol{V_i}$, which is given by (12), as the corresponding payoff of strategy *i* to select the best pheromone matrix for the lower information entropy subpopulation.

$$V_i = f(p_i) \cdot A_i \tag{12}$$

and $f(p_i) = \frac{E(p_i)}{E(p)_{gb}}$, $A_i = \frac{L_{gb}}{L_i}$where $\boldsymbol{E(p)_{gb}}$ represents the value of global optimal information entropy in all strategies, and $\boldsymbol{E(P_i)}$ denotes the information entropy of strategy i; $\boldsymbol{L_{gb}}$ is the global optimal path, and $\boldsymbol{L_i}$ is the current optimal solution under strategy i. From the equal (12), the larger value of $\boldsymbol{A_i}$ denotes the higher solution quality of the strategy and the higher $\boldsymbol{f(P_i)}$ reflects the better population diversity. Therefore, the higher value of the payoff $\boldsymbol{V_i}$ is, the better solution quality would be.

3.3.2 Profit Distribution Strategy

When the maximum payoff $\boldsymbol{V}$ belongs to the cooperative union, the payoff distribution strategy is proposed to distribute the payoff obtained by the union to the participants reasonably. In another hand, if the profit of cooperative union is higher than non-union, the population will choose cooperation under the premise of individual rationality. And in this case, in order to maintain the stability of the cooperative alliance, it is necessary to distribute the profits reasonably among the members of the union. As we known, a reasonable distributed mechanism can promote participants to take part in the union more actively and make the alliance structure more stable, thus, they can obtain a higher collective profit. In this paper, we define the increasing profits after cooperation game as the pheromone to reward the common paths among participants, which the formula is given by (13)-15 and the public path is shown as in Fig. 2.

$$\Delta\tau_{\text{profit}} = \frac{V_{\text{new}} - V_{\text{before}}}{L_{\text{new}}} \tag{13}$$

$$\Delta\tau^k_{\text{profit}} = \frac{E(p_k)}{\sum_{k\in K} E(p_k)} \cdot \Delta\tau_{\text{profit}} \tag{14}$$

$$\tau^k_{\text{public}} = (1-\rho)\cdot\tau + \rho\cdot\Delta\tau + \Delta\tau^k_{\text{profit}} \tag{15}$$

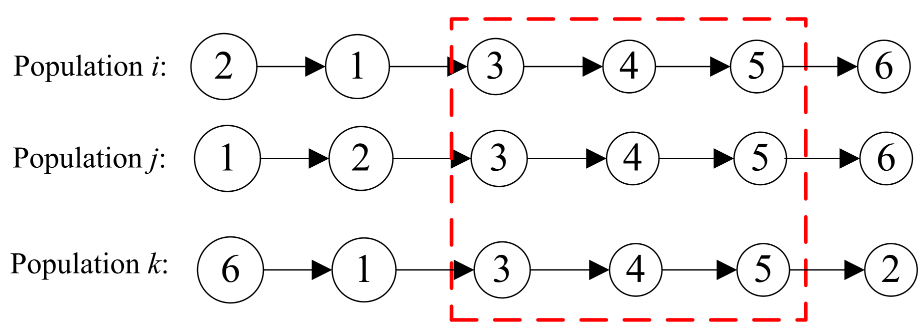
where $\boldsymbol{\Delta\tau_{profit}}$ is the increasing profits of the colony after cooperative game which denotes the payoff of the union, $\boldsymbol{V_{before}}$ is the profit of the population without pheromone fusion, $\boldsymbol{V_{new}}$ is the maximum profit in all unions after pheromone fusion, and $\boldsymbol{L_{new}}$ is the length of new tour created by cooperation game; $\boldsymbol{\Delta\tau^k_{profit}}$ is the profit of kth population, and $\boldsymbol{E(p_k)}$ is the information entropy of the kth colony in the union.

From the formula (13) and (14), we can see that the public path between lower entropy population and other participant populations will be rewarded added pheromone in the union, which makes the cooperative profits higher than non-cooperation, thus it can ensure the effectiveness of cooperation and enhance the collective rationality of the group. In addition, the path selected by many populations will also belong to the optimal tour largely [6]. Therefore, rewarding the pheromone to those paths can narrow the useless research of the ant colony and accelerate the convergence speed of the algorithm. And the formula (15) is the pheromone updating rule of the population k on public paths.

3.3.3 Pheromone Smoothing Mechanism

What's more, a rare situation that the payoff of non-cooperation is higher needs to be considered. This means that the profits created by the pheromone fusion mechanism are less than these under non-fusion mechanism. In this case, we first judge whether the population finds a better solution. And the pheromone matrix of the population

Fig. 2 Public path



would not be changed when a better solution is found, which denotes that the population has better potential, otherwise, it shows that the pheromone fusion mechanism is invalid and the algorithm has been stagnant. To deal with this issue, the pheromone smoothing mechanism (**PSM**) is introduced to help the population jump out of the local optimum, which the formula is as follows:

$$\tau_{ij}^{i} = \frac{\tau_{\min}^{i} + \tau_{ij}^{i}}{2} \tag{16}$$

where $\tau_{\min}^{i}$ and τ_{ij}^{i} are the minimum concentration of pheromone and the value of pheromone on edge (i, j) in population i, respectively. And if the population i is the ACS, the $\tau_{\min}^{i}$ is equal to τ_0. Otherwise, it is equivalent to $\tau_{\min}$ in MMAS. The formula (16) reflects a novel reinitialization pheromone method that the larger gap between the pheromone value on the path and the minimum pheromone trail is, the more pheromone evaporation will be. However, the edges with minimum pheromone would not be volatilized causing that the pheromone matric reduces to $\tau_{\min}$ with different degrees. Thus, it can reduce the maximum pheromone trail while would not lost the information of suboptimal path.

3.4 Algorithm Description

Informally, the algorithm proposed in this research can be described as follows: When one population happens that its entropy is lower than the threshold value, this population will launch the cooperative game model right now, and other populations will choose cooperation or defection due to pheromone fusion mechanism by equal (9)-equal (11) to create the strategy sets shown in Table 1. Then, this population with lower information entropy will produce the payoff matrix, which is shown in following Table 2, based on the different pheromone matrices in Table 1.

where $\boldsymbol{V_{11}}$ is the payoff under the full union structure, $\boldsymbol{V_{12}}$ and $\boldsymbol{V_{21}}$ are the payoff under the sub-union structure, and $\boldsymbol{V_{22}}$ is the payoff under the single union structure. Thus, the Pareto optimality of this game is:

$$V_{\text{pareto}} = \max(V_{11}, V_{12}, V_{21}, V_{22}) \tag{17}$$

Next, we start to judge which union is the Pareto optimality. If the $V_{\text{pareto}} = V_{22}$, which means the pheromone fusion mechanism is failure, the pheromone smoothing mechanism will be applied to help the algorithm get rid of local optimum, otherwise, the execute profit distribution strategy is introduced to

Table 2 The payoff matrices of cooperative game

	$C(P_j)$	$D(P_j)$
$C(P_k)$	V_{11}	V_{12}
$D(P_k)$	V_{21}	V_{22}

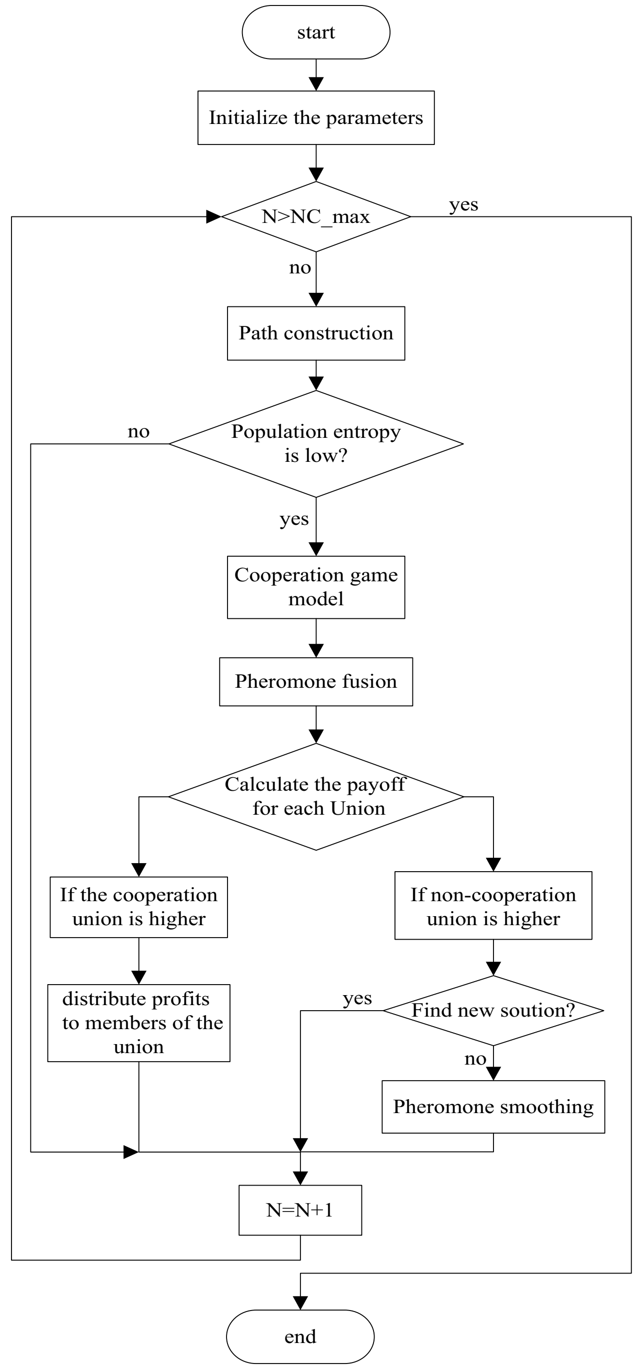


Fig. 3 The flow chart of CGMACO algorithm

speed up the convergence of the algorithm. And algorithm 1 and Fig. 3 are the framework and flowchart of the proposed algorithm respectively, which is shown as follows in detail.

Algorithm 1 CGMACO Algorithm for TSP

```
Procedure CGMACO ()
Begin:
initialize the pheromone and the parameters
calculate the distance between cities
calculate the initial pheromone matrix
While N<Nc do;
While m<M do; #m as ants' number
  construct ant solutions by equal (1)
  update the pheromone matrix by equal (2) and (3)
End while
Obtain each colony's information entropy by equal (7)-(8)
If the information entropy < threshold
  Execute the pheromone fusion by equal (9) and (10)
  Execute the cooperation game theory model by equal (11)
  calculate the payoff of each union by equal (12)
    If cooperative union payoff > non-cooperative union payoff
      Distribute the profits for the union members by equal (13) - (15)
    else
      Execute the smoothing pheromone mechanism by equal (16)
    End if
  End if
End while
Return best solution
```

Table 3 Experimental factors and levels of ACS

	α	β	ρ	ζ	q0
Level 1	1	2	0.1	0.1	0.6
Level 2	2	3	0.2	0.2	0.7
Level 3	3	4	0.3	0.3	0.8
Level 4	4	5	0.4	0,.4	0.9

4 Experiment and Simulation

In this part, Sect. 4.1 is the parameters setting in ACS, MMAS and E(P)*, which E(P)* is the threshold value of the information entropy. Section 4.2 analyses the performance of the different strategies we have proposed. Section 4.3 compares the proposed algorithm with the conventional ACO algorithms. Section 4.4 is the experimental comparison between CGMACO and the algorithms including

Table 5 Analysis of test results of ACS

	α	β	ρ	ζ	q0
K_1	**89,738.6**	90,710.6	**89,864.9**	92,373.6	90,132.5
K_2	89,793.1	92,144.8	89,985.7	89,969.8	90,277.1
K_3	90,648.7	**89,744.1**	92,551.8	**89,750.6**	**90,094.5**
K_4	92,244.9	89,825.9	90,023.0	90,331.4	91,921.3
k_1	22,434.7	22,677.6	22,466.2	23,093.4	22,533.1
k_2	22,448.3	23,036.2	22,496.4	22,492.5	22,569.3
k_3	22,662.2	22,436.0	23,138.0	22,437.6	22,523.6
k_4	23,061.2	22,456.5	22,505.7	22,582.8	22,980.3
Max	23,061.2	23,036.2	23,138.0	23,093.4	22,980.3
Min	22,434.7	22,436.0	22,466.2	22,582.8	22,523.6
Range	626.5	600.2	678.1	510.6	456.7
Scheme	Level 1	Level 3	Level 1	Level 3	level3

$K_i(i = 1, 2, 3, 4)$ represents the sum of experimental results at each level; $k_i(i = 1, 2, 3, 4)$ represents the average of each level; Max and Min represent the maximum length and the minimum length, respectively; Range represents the difference between the maximum and the minimum; Scheme represents each factor

The optimal parameter of ACS algorithm is: $\alpha = 1,\ \beta = 4,\ \rho = 0.1,\ \xi = 0.3,\ q0 = 0.8$

Table 4 Orthogonal test scheme and test results of ACS

No	α	β	ρ	ζ	q0	Result
1	1	2	0.1	0.1	0.6	22,598.7
2	1	3	0.2	0.2	0.7	22,422.6
3	1	4	0.3	0.3	0.8	22,363.5
4	1	5	0.4	0.4	0.9	22,353.7
5	2	2	0.2	0.3	0.9	22,433.9
6	2	3	0.1	0.4	0.8	22,450.8
7	2	4	0.4	0.1	0.7	22,446.4
8	2	5	0.3	0.2	0.6	22,462.0
9	3	2	0.3	0.4	0.7	23,023.6
10	3	3	0.4	0.3	0.6	22,568.6
11	3	4	0.1	0.2	0.9	22,430.9
12	3	5	0.2	0.1	0.8	22,625.7
13	4	2	0.4	0.2	0.8	22,654.4
14	4	3	0.3	0.1	0.9	24,702.8
15	4	4	0.2	0.4	0.6	22,503.3
16	4	5	0.1	0.3	0.7	22,384.5

Table 6 Experimental factors and levels of MMAS

	α	β	ρ
Level 1	1	2	0.1
Level 2	2	3	0.2
Level 3	3	4	0.3
Level 4	4	5	0.4

Table 7 Orthogonal test scheme and test results of MMAS

No	α	β	ρ	Result
1	1	2	0.1	22,942.8
2	1	3	0.2	22,416.8
3	1	4	0.3	22,384.1
4	1	5	0.4	22,310.7
5	2	2	0.2	23,107.7
6	2	3	0.1	22,461.1
7	2	4	0.4	22,585.6
8	2	5	0.3	22,429.4
9	3	2	0.3	24,285.2
10	3	3	0.4	23,135.1
11	3	4	0.1	22,767.8
12	3	5	0.2	22,670.9
13	4	2	0.4	24,852.2
14	4	3	0.3	23,500.7
15	4	4	0.2	23,147.3
16	4	5	0.1	22,864.5

*Note: result represents the average value after 20 tests

Table 8 Analysis of test results of MMAS

	α	β	ρ
K_1	**22,513.6**	23,797.0	**22,759.0**
K_2	22,645.9	22,878.4	22,835.7
K_3	23,214.7	22,721.2	23,149.8
K_4	23,591.2	**22,568.8**	23,220.9
k_1	22,513.6	23,797.0	22,759.0
k_2	22,645.9	22,878.4	22,835.7
k_3	23,214.7	22,721.2	23,149.8
k_4	23,591.2	22,568.8	23,220.9
Max	23,591.2	23,797.0	23,220.9
Min	22,513.6	22,568.8	22,759.0
Range	1077.6	1228.2	461.9
Scheme	Level 1	Level 3	Level 1

$K_i(i = 1, 2, 3,4)$ represents the sum of experimental results at each level; $k_i(i = 1, 2, 3,4)$ represents the average of each level; Max and Min represent the maximum length and the minimum length, respectively; Range represents the difference between the maximum and the minimum; Scheme represents each factor

The optimal parameter of MMAS algorithm is: $\alpha = 1,\ \beta = 5,\ \rho = 0.1$

other improved ACO algorithms and swarm intelligence algorithms. Moreover, the experimental platform is the MATLAB R2019b in Windows 10 environment, the CPU, with 16 GB RAM memory capacity, is Intel(R) Core (TM) i7-10700F, and the experiments are applied to execute based on different scale TSP instances, each instance runs for 20 times independently.

4.1 Parameters Setting

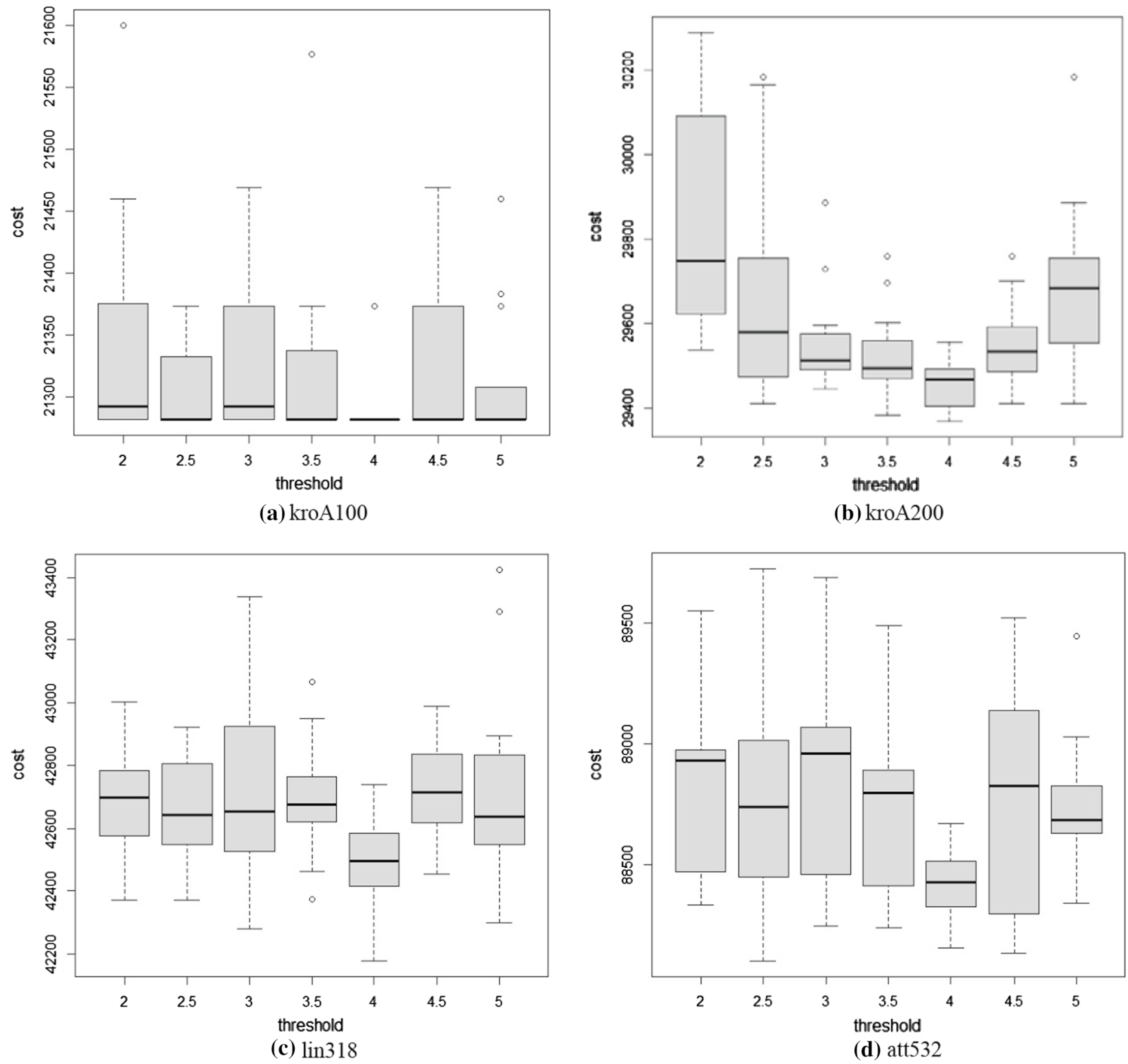
The first experiment of this part is to set the appropriate parameters of ACS and MMAS. In order to improve the performance of the proposed algorithm, we adopt the orthogonal tests that have four levels and sixteen parameter combinations to set the appropriate parameter for each colony. Besides, kroB100 instance is selected to carry out in the orthogonal experiment, and each combination of the parameters experiment is executed 20 times independently to ensure the reliability of the experiment. Tables 3, 4 and 5 are the experimental results of ACS, and Tables 6, 7 and 8 denote the results of MMAS.

The second experiment of this part is to select the suitable entropy threshold for CGMACO. Information entropy threshold (E(P)*) is also an important parameter in this research. If ***E(P) **** is too large, the communication frequency of subpopulations would be higher, which will make the multi-population degenerate into a single population. And if it is set too small, the insufficient interaction among sub-colonies would also decrease the diversity of the algorithm. In this research, we select the suitable value of E(P)* through the experiment based on four TSP instances such as kroA100, kroA200, lin318, att532. And Fig. 3 illustrates those specific experiments data. As it clears in it, the results have shown that the smallest fitness evaluation function value can be obtained under the parameter E(P)* = 4. Therefore, we set the parameter E(P)* = 4 in the following experiments (Fig. 4).

From the above experimental results, the final setting results of the algorithm parameters are shown in following Table 9, which the ρ denotes the global pheromone evaporation rate, ζ is the local pheromone evaporation rate, and M represents ant number.

4.2 Strategy Analysis

In this part, we analyse the effectiveness of three strategies proposed above including adaptive communication strategy based on information entropy, pheromone fusion strategy and pheromone smoothing mechanism. Strategy-1(S-1) is the algorithm that has pheromone fusion strategy and pheromone smoothing mechanism but does not use information entropy. Strategy-2 (S-2) is the algorithm that retains adaptive communication strategy and pheromone smoothing



(a) kroA100 **(b)** kroA200 **(c)** lin318 **(d)** att532

Fig. 4 Adjustment of the entropy threshold

Table 9 The parameter setting of the algorithm

Parameter	ACS	MMAS
α	1	1
β	4	(5)
ρ	0.1	0.1
ζ	0.3	–
q0	0.8	–
M	20	20
Iteration	2000	2000
E(P)*	4	4

mechanism but does not use pheromone fusion mechanism. Strategy-3 (S-3) represents the algorithm that has adaptive communication strategy and pheromone fusion strategy but does not use pheromone smoothing mechanism. In addition, in order to make the experimental algorithm run normally, we use fixed algebra communication strategy in S-1, here, we select it as 200 iterations and exchange the optimal solution between colonies in S-2. In experiment, kroB100, kroA200 and fl417 TSP instances are selected and analysed with three aspects including optimal solution error rate, worst solution and average solution. And each instance runs

Table 10 Experiment results with different strategies

TSP instance	Opt	Best cost				Worst cost				Average			
		S-1	S-2	S-3	CGMACO	S-1	S-2	S-3	CGMACO	S-1	S-2	S-3	CGMACO
kroB100	22,141	**22,141**	22,179	**22,141**	**22,141**	22,295	22,336	22,284	**22,275**	22,227.45	22,254.80	22,207.90	**22,201.20**
kroA200	29,368	29,382	29,409	29,382	**29,368**	29,728	29,759	29,760	**29,628**	29,500.10	29,545.00	29,512.60	**29,478.55**
fl417	11,861	11,932	11,942	11,929	**11,916**	12,184	12,143	12,141	**12,053**	12,002.15	12,004.10	11,987.95	**11,978.05**

20 times, 2000 iterations each time. The experimental results are shown in Table 10 and Fig. 5.

As we can see in the results, the performance of S-2, which the algorithm without pheromone fusion mechanism, is worst, while CGMACO, which the algorithm with all strategies, is best. And S-1 and S-3 have their own advantage on different instances. This is because the pheromone fusion mechanism can effectively take full advantage of the heterogeneous population and improve the diversity of the algorithm, due to regulating the pheromone distribution of each subpopulation. While the communication efficiency among subpopulations is greatly reduced without this mechanism, which it has been confirmed in experiments, thus the accuracy of the solutions reduces. The results of the algorithm without information entropy strategy and pheromone smoothing mechanism are better than the algorithm without pheromone fusion mechanism, but the quality of the solution is still lower than that of CGMACO.

4.3 Comparison with Traditional ACO Algorithm

In the first phase of the experiment, we compare CGMACO with traditional ACO algorithms. And Table 10 reveals the performance of proposed algorithm with ACS and MMAS based on 22 TSP instances. The evaluation criterions in experiment mainly include the best solution, the worst solution, mean solution, error rate and the standard deviation, which the error rate and standard deviation formula are as follows:

$$\text{error} = \frac{L_{\text{ACO}} - L_{\text{opt}}}{L_{\text{opt}}} \times 100\% \tag{18}$$

where $\boldsymbol{L_{ACO}}$ represents the optimal solution of each algorithm, and $\boldsymbol{L_{opt}}$ represents the standard optimal solution of the known test set.

$$\text{dev} = \sqrt{\frac{1}{N}\sum\nolimits_{i=1}^{N}(L_i - \overline{L})^2} \tag{19}$$

where $\boldsymbol{dev}$ denotes the standard deviation, $\boldsymbol{N}$ represents the number of times the algorithm runs, and $\boldsymbol{L_i}$ is the solution obtained by the algorithm in the i-th experiment.

As it clears in Table 11, in small-scale instances with city's scale from 51 to 200, both CGMACO and conventional ACOs can achieve better results in the error rate, but CGMACO has lower average solution and standard deviation than ACS and MMAS due to the cooperation among multi-populations, which proves that CGMACO has better stability than comparison algorithms. Moreover, more flexible interaction mechanism based on the self-adaptive communication strategy makes the information transmission among sub-colonies more adaptive, which can greatly diversify the solutions of algorithm. As shown in Table 11,

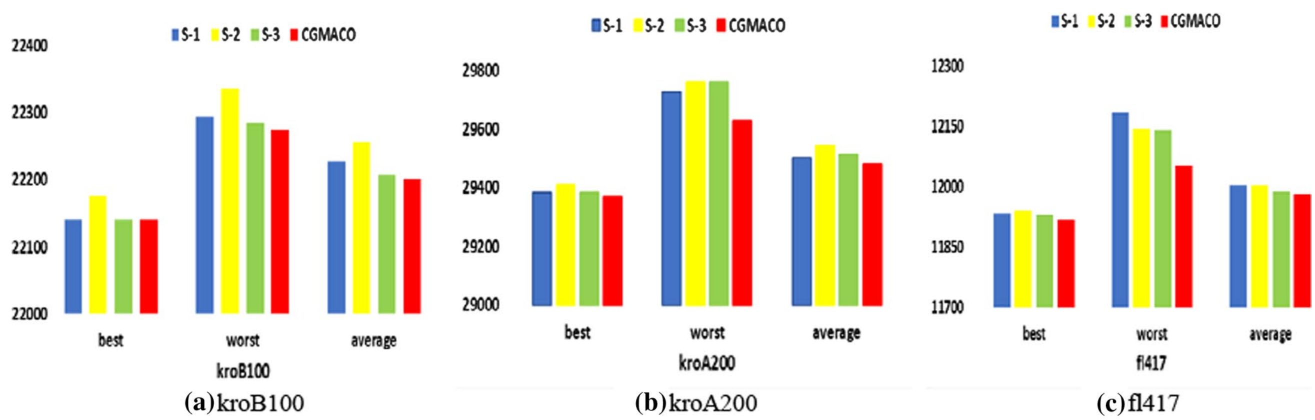


Fig. 5 Results comparison in different strategies

in middle-scale instances, such as tsp225, pr264 and a280, CGMACO gains the standard optimal solution, but the comparison algorithms do not get. Although our algorithm would not obtain the optimal solution in lin318, the error rate remains within 1%, which is more superior to the 1.81% and 2.42% obtained by ACS and MMAS, respectively. With the increase in city scale from fl417 to rl1304, the error rate of traditional ACOs is exceedingly more than 1% and gradually rising. While attributed to the pheromone fusion mechanism, we proposed in this research, the pheromone distribution of each colony can be regulated appropriately and the information exchanged among subpopulations becomes more effective, which makes the improved algorithm still control the error rate within 1%. Besides, the cooperative game model can further promote the efficiency of communication among sub-colonies, helping that CGMACO possesses lower mean solution and standard deviation solution and still enables strong stability in large-scale instances.

Figure 6 illustrates the obvious improvement of convergence speed and solution accuracy of the proposed algorithm. According to the population profits distribution strategy in CGMACO, high-quality solutions can be selected in the early algorithm stage which guides more ants to explore around the optimal solution and avoids much useless search, while ACS and MMAS have to seek the whole search space to complete the evolution causing that it is hard to convergence. As Fig. 6 shown, CGMACO has faster convergence than conventional ant colony algorithms. And in the later stage, due to the excessive accumulation of pheromones, single population ACOs easily fall into stagnation. However, with the help of pheromone smoothing mechanism, the pheromone matric is reinitialized helping CGMACO jump out of local optimum effectively. Taking rl1304 instance as an example, as shown in Fig. 6f, CGMACO can still obtain new better solution at about 1100 and 1600 iteration in terms of this mechanism. In a word, the CGMACO algorithm has faster convergence than ACS and MMAS without losing high-quality solutions.

Figure 7 demonstrates the optimal tour found by CGMACO in the simulate experiment.

4.4 Comparison with Other Algorithms

In the second phase of the experiment, we compare CGMACO with DBAL [32] and DSMO [33]in detail under error rate column, average solution and standard deviation solution. According to Table 12 and Fig. 8, it can conclude that the improved algorithm has strong competitiveness with comparison algorithm. In the experiment, we select the instances with city's scale from 51 to 1000. Both in small-scale TSP instances and large-scale TSP instances, CGMACO outperforms DSMO under three evaluation criterions, but CGMACO and DBAL have their own performance advantages in different TSP instances. For example, in lin318, the factors including error rate, average solutions and standard deviation in DBAL (0.1%, 42,268.1 and 128.24) are better than them in CGMACO (0.23%, 42666.1 and 181.32), respectively; however, our proposed algorithm is superior to DBAL in pr1002. In general, under the synergy of the pheromone fusion mechanism and cooperative game model, the efficiency of coordination among sub-colonies has been improved greatly. As we can see in Table 12, among 13 TSP instances experiments, CGMACO outperforms DBAL with 8 instances including eil51, st70, eil76, eil101, kroA100, pr264, pr439 and pr1002, which proves the excellent performance of CGMACO.

In the third phase of the experiment, we compare the CGMACO with other optimization algorithms. The comparison optimization algorithms mainly have single ant colony algorithms that include HAACO [12], PACO-3opt [34], DEACO [35], HMMA [36], multi-ant colony algorithms such as JCACO [22], NACO [24], LDTACO [25] and other swarm intelligence algorithms that include hybrid

Table 11 Performance compare CGMACO with ACS and MMAS

Instances	Optimal	Algorithm	Best	Worst	Mean	Error%	Dev
eil51	426	ACS	426	435	428.61	0.00	2.18
		MMAS	426	432	428.10	0.00	2.04
		CGMACO	**426**	**427**	**426.55**	**0.00**	**0.51**
eil76	538	ACS	538	551	544.93	0.00	4.16
		MMAS	538	552	542.20	0.00	4.71
		CGMACO	***538***	***541***	***538.60***	***0.00***	***1.09***
eil101	629	ACS	630	650	639.51	0.16	7.04
		MMAS	630	653	642.30	0.16	6.52
		CGMACO	**629**	**638**	**632.05**	**0.00**	**2.96**
kroA100	21,282	ACS	21,282	21,835	21,450.62	0.00	157.83
		MMAS	21,282	21,945	21,396.30	0.00	166.59
		CGMACO	**21,282**	**21,302**	**21,286.95**	**0.00**	**8.57**
kroB100	22,141	ACS	22,270	22,420	22,366.41	0.58	43.41
		MMAS	22,220	22,580	22,320.31	0.36	91.42
		CGMACO	**22,141**	**22,237**	**22,163.40**	**0.00**	**33.60**
ch130	6110	ACS	6162	6348	6255.00	0.85	58.16
		MMAS	6121	6359	6188.40	0.18	37.54
		CGMACO	**6110**	**6220**	**6152.00**	**0.00**	**21.55**
ch150	528	ACS	6548	6778	6600.50	0.31	54.02
		MMAS	6528	6641	6578.30	**0.00**	29.55
		CGMACO	**6528**	**6608**	**6553.00**	**0.00**	**13.65**
kroA150	26,524	ACS	26,765	27,945	27,267.20	0.91	332.13
		MMAS	26,665	27,251	27,010.80	0.53	177.04
		CGMACO	**26,524**	**27,056**	**26,717.25**	**0.00**	**157.52**
kroB150	26,130	ACS	26,244	27,269	26,699.11	0.44	302.76
		MMAS	26,196	27,034	26,443.91	0.25	213.36
		CGMACO	**26,130**	**26,417**	**26,292.15**	**0.00**	**72.56**
rat195	2323	ACS	2331	2373	2352.05	0.34	13.30
		MMAS	2340	2385	2353.50	0.73	13.56
		CGMACO	**2330**	**2348**	**2338.60**	**0.30**	**4.73**
d198	15,780	ACS	15,970	16,420	16,153.93	1.20	120.69
		MMAS	16,065	17,117	16,545.25	1.81	319.41
		CGMACO	**15,839**	**16,159**	**15,964.60**	**0.46**	**83.19**
kroA200	29,368	ACS	29,536	30,290	29,828.00	0.57	242.34
		MMAS	29,460	30,184	29,639.00	0.31	185.09
		CGMACO	**29,368**	**29,628**	**29,478.55**	**0.00**	**63.85**
kroB200	29,437	ACS	29,816	30,701	30,348.87	1.29	261.93
		MMAS	29,766	30,957	30,195.95	1.12	401.21
		CGMACO	**29,437**	**29,833**	**29,640.80**	**0.00**	**106.68**
tsp225	3916	ACS	3931	4061	3997.70	0.38	39.42
		MMAS	3923	4033	3984.35	0.18	31.62
		CGMACO	**3916**	**3970**	**3940.40**	**0.00**	**15.30**
pr264	49,135	ACS	49,198	51,702	49,734.35	0.13	671.45
		MMAS	**49,135**	51,927	49,715.00	**0.00**	786.85
		CGMACO	**49,135**	**49,245**	**49,167.95**	**0.00**	**36.04**
a280	2579	ACS	2594	2712	2636.65	0.58	36.48
		MMAS	2587	2683	2624.75	0.31	27.75
		CGMACO	**2579**	**2621**	**2595.89**	**0.00**	**13.93**
lin318	42,029	ACS	42,790	43,586	43,277.01	1.81	221.68
		MMAS	43,046	44,879	43,614.45	2.42	447.74
		CGMACO	**42,146**	**43,052**	**42,662.10**	**0.28**	**181.32**
fl417	11,861	ACS	12,031	12,404	12,185.91	1.43	100.58
		MMAS	12,006	12,363	12,174.15	1.22	94.02
		CGMACO	**11,916**	**12,178**	**11,991.15**	**0.46**	**51.83**
pr439	107,217	ACS	108,309	116,846	110,905.95	1.02	2402.69
		MMAS	107,929	114,244	110,826.90	0.66	1682.36
		CGMACO	**107,572**	**109,309**	**108,477.40**	**0.33**	**524.90**

Table 11 (continued)

Instances	Optimal	Algorithm	Best	Worst	Mean	Error%	Dev
att532	86,729	ACS	88,976	93,426	90,438.85	2.60	887.57
		MMAS	92,041	98,446	94,588.40	6.12	1626.73
		CGMACO	**87,541**	**90,284**	**88,794.40**	**0.94**	**459.54**
p654	34,643	ACS	35,032	37,318	35,476.70	1.12	535.75
		MMAS	36,257	37,923	36,996.80	4.66	518.65
		CGMACO	**34,762**	**35,539**	**35,157.85**	**0.34**	**207.25**
rl1304	252,948	ACS	264,904	277,443	270,350.80	4.73	3657.79
		MMAS	279,232	303,688	288,417.80	10.39	7138.96
		CGMACO	**254,199**	**261,890**	**260,248.50**	**0.49**	**2521.41**

ant colony particle swarm optimization algorithm called PSO-ACO-3opt [2], Discrete Bat Algorithm DBAL [32], Discrete Spider Monkey Optimization DSMO [33], Discrete Water Cycle Algorithm DWCA [37], Artificial Bee Colony algorithm ABC[38], Discrete Symbiotic Organisms Search algorithm DSOS [39] and Improved Discrete Bat Algorithm IBA [40]. Tables 12 and 13 show the specific experiment data, which the best is the optimal solution obtained by each algorithm and the error is the error rate column defined by equal (18). And the "-" in the table denotes that the comparison algorithm does not test the instance.

As we can see in Table 13, the results demonstrate that our proposed algorithm can find the standard optimal solutions in all small-scale TSP instances, which outperforms the comparison algorithms such as LDTACO, DSMO, DEACO and PACO-3opt. Moreover, in Table 14, CGMACO is also superior to the recent algorithms. In tsp225 and a280 instances, CGMACO can still obtain the standard optimal solution, while DSMO, NACO, JCACO would not get. And in fl417, pr439 and p654 instances, the superiority of CGMACO has been confirmed obviously, which the error rate CGMACO obtained is significantly lower than the comparison algorithms. These satisfactory experimental results are mainly ascribed to the strong search ability improved by cooperative game based on the pheromone fusion mechanism. Specifically, under the pheromone fusion mechanism, more useful information can be explored. And due to the cooperative game model, the beneficial pheromone distribution can be generated in subpopulations. These two methods can take full use of the advantage of heterogeneous populations. In short, our proposed algorithm, CGMACO, has strong competitiveness with the state-of-art algorithms and can obtain higher quality solutions, especially for large-scale TSP instances.

5 Conclusion

In this paper, we have proposed a novel ant colony algorithm, so-called multi-ACO based on pheromone fusion mechanism of cooperative game, to solve travelling salesman problems. In multiple populations, we select two ACS colonies and one MMAS colony. Two ACS subpopulations form the homogenous population, which can better amplify the convergence speed of ACS. In addition, we also add one MMAS subpopulation to form a heterogeneous population, which can effectively enhance the diversity of the ACS. The advantages of multiple populations complement to ensure the solution quality of the algorithm.

In addition, the pheromone fusion mechanism is applied to regulate the pheromone distribution of each subpopulation. It fuses the pheromone matrix of other subpopulations based on retaining the original population information, which can exchange the information among multiple

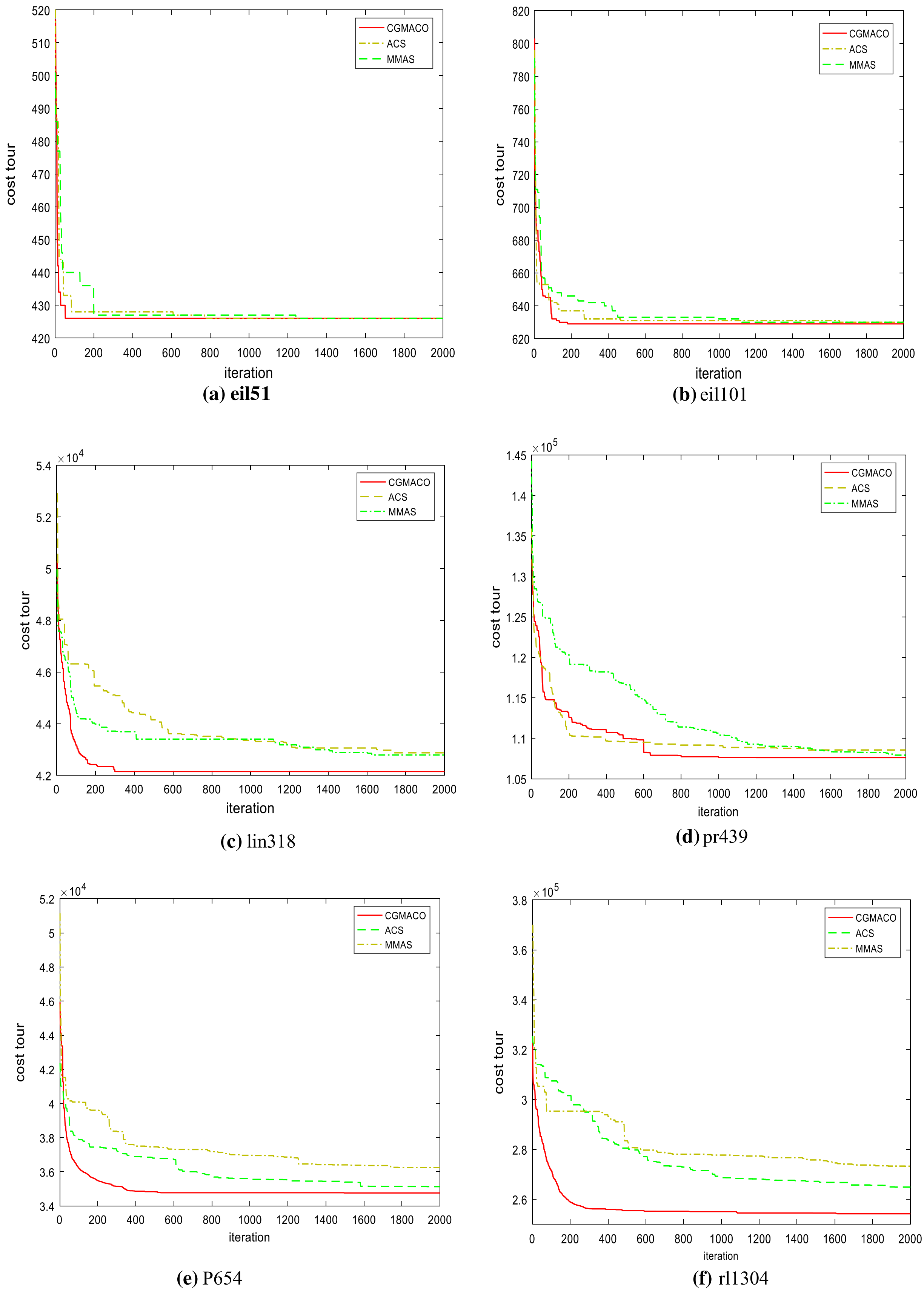
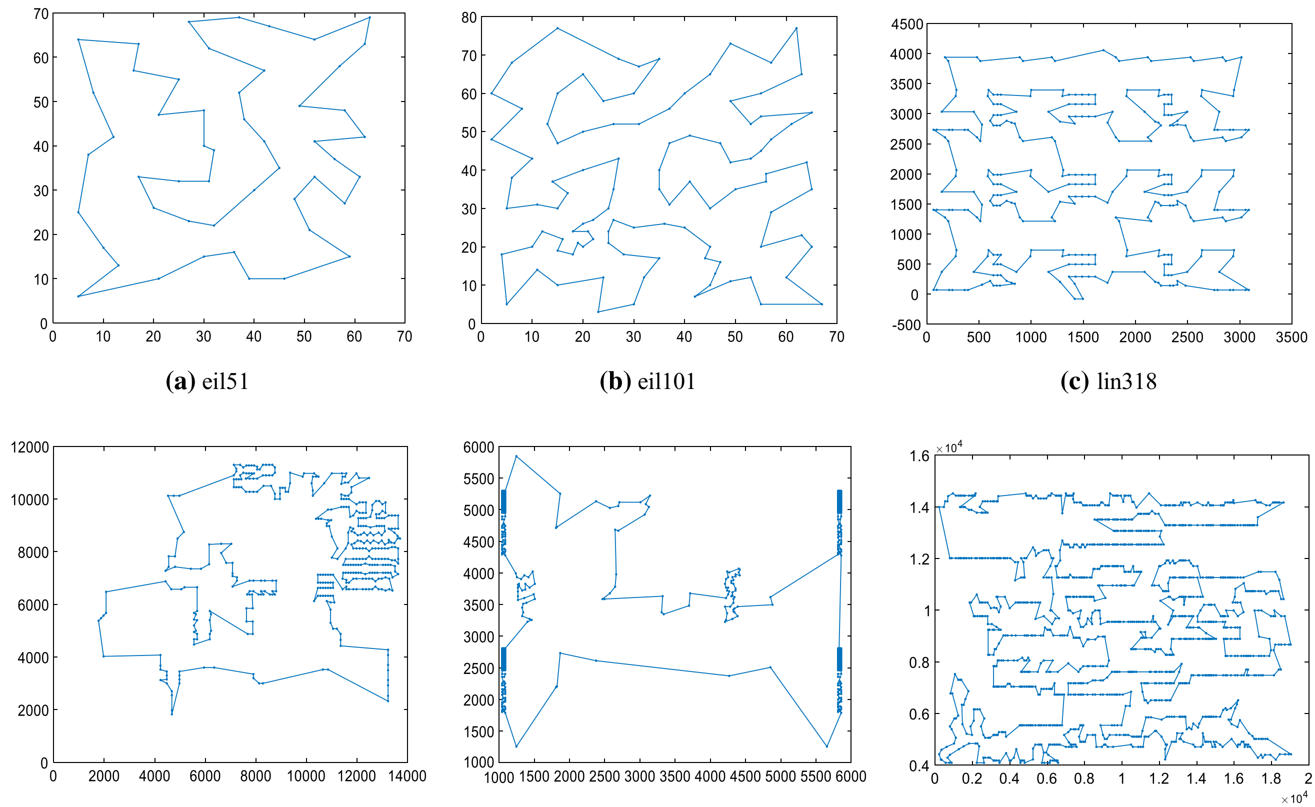


Fig. 6 Comparison the convergence of different algorithms



(a) eil51 (b) eil101 (c) lin318

(d) pr439 (e) p654 (f) rl1304

Fig. 7 Optimal tour found by CGMACO

Table 12 Compare CGMACO with DBAL and DSMO

TSP instance	Optimal	DBAL (2021)			DSMO (2020)			CGMACO		
		Error	Average	Std	Error	Average	Std	Error	Average	Std
eil51	426	**0.00**	426.70	0.55	0.67	436.96	4.73	**0.00**	**426.55**	**0.51**
Berline52	7542	**0.00**	**7542.00**	**0.00**	0.02	7633.60	85.4	**0.00**	**7542.00**	**0.00**
eil76	538	**0.00**	539.10	1.47	3.84	572.70	7.56	**0.00**	**538.61**	**1.09**
kroA100	21,282	**0.00**	21,287.11	9.02	0.07	22,024.30	508.89	**0.00**	**21,286.95**	**8.57**
kroB100	22,141	**0.00**	22,164.30	36.86	0.75	22,707.91	259.83	**0.00**	**22,163.40**	**33.60**
eil101	629	**0.00**	632.15	3.32	3.02	674.40	10.97	**0.00**	**632.05**	**2.96**
pr152	73,682	**0.00**	**73,723.00**	**62.19**	0.76	76,526.77	1663.08	**0.00**	73,983.85	195.45
kroA200	29,368	**0.00**	**29,424.55**	**61.26**	3.79	31,828.60	652.32	**0.00**	29,478.55	63.85
kroB200	29,437	**0.00**	**29,483.60**	**46.89**	4.34	31,781.60	487.39	**0.00**	29,640.8	106.68
pr264	49,135	**0.00**	49,175.90	92.06	–	–	–	**0.00**	**49,167.95**	**36.04**
lin318	42,029	**0.10**	**42,268.10**	**128.24**	4.97	45,460.30	660.47	0.35	42,662.10	181.32
pr439	107,217	0.36	**107,903.25**	**166.58**	4.56	116,379.20	2462.82	**0.33**	108,477.90	524.9
pr1002	259,047	4.21	271,473.05	734.10	–	–	–	**3.16**	**268,619.8**	**698.22**

Fig. 8 The average solution in each algorithm

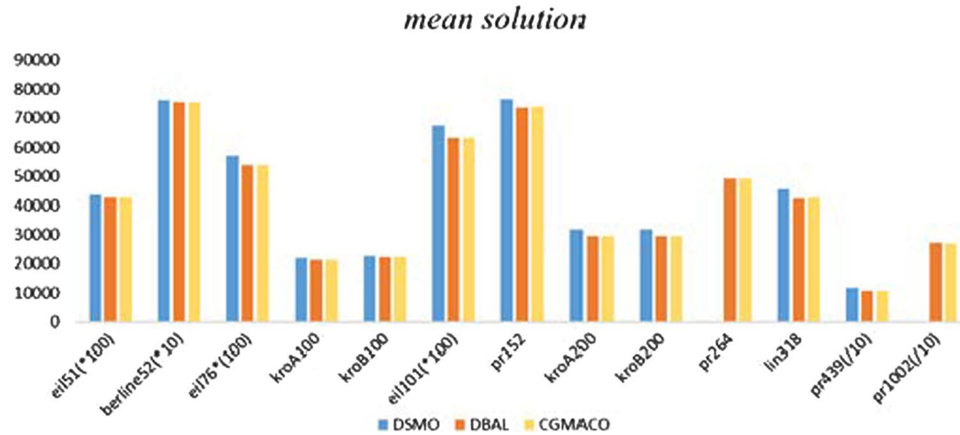


Table 13 Compare proposed algorithm with other algorithms in small-scale TSP instances

Algorithm	Instance	eil51	Berline52	st70	eil76	kroA100	eil101	kroA150	kroA200
Proposed	Best	***426***	***7542***	***675***	***538***	***21,282***	***629***	***26,524***	***29,368***
Algorithm	Error%	***0.00***	***0.00***	***0.00***	***0.00***	***0.00***	***0.00***	***0.00***	***0.00***
DBAL (2021)	Best	426	7542	675	538	21,282	629	–	29,368
	Error%	0.00	0.00	0.00	0.00	0.00	0.00	–	0.00
LDTACO (2021)	Best	426	–	–	538	21,282	629	26,524	29,380
	Error%	0.00	–	–	0.00	0.00	0.00	0.00	0.04
DSMO (2020)	Best	428	7544	677	558	21,298	648	27,591	30,481
	Error%	0.67	0.02	0.30	3.72	0.07	3.02	4.02	3.79
HAACO (2020)	Best	426	7542	675	538	21,282	629	–	29,483
	Error%	0.00	0.00	0.00	0.00	0.00	0.00	–	0.39
DEACO (2020)	Best	426	7542	675	541	21,282	629	26,572	29,368
	Error%	0.00	0.00	0.00	0.52	0.00	0.00	0.18	0.00
ABC (2019)	Best	427	7542	675	538	21,282	629	–	29,450
	Error%	0.23	0.00	0.00	0.00	0.00	0.00	-	0.28
PACO-3opt (2018)	Best	426	7542	676	538	21,282	629	–	29,368
	Error%	0.00	0.00	0.15	0.00	0.00	0.00	–	0.00
DWCA (2018)	Best	426	7542	675	543	21,282	639	–	–
	Error%	0.00	0.00	0.00	0.9	0.00	1.5	–	–
DSOS (2017)	Best	426	7542	675	542	21,282	640	–	29,477
	Error%	0.00	0.00	0.00	0.74	0.00	1.75	–	0.37
IBA (2016)	Best	426	7542	675	539	21,282	634	–	–
	Error%	0.00	0.00	0.00	0.19	0.00	0.79	–	–
PSO-ACO-3opt (2015)	Best	426	7542	676	538	21,301	631	–	29,468
	Error%	0.00	0.00	0.15	0.00	0.09	0.32	–	0.34

Table 14 Compare proposed algorithm with other algorithms in large-scale TSP instances

Algorithm	Instance	tsp225	a280	lin318	fl417	pr439	p654
Proposed algorithm	Best	***3916***	***2579***	***42,146***	***11,916***	***107,512***	***34,762***
	Error%	***0.00***	***0.00***	***0.28***	***0.46***	***0.34***	***0.28***
LDTACO	Best	3926	2579	42,327	11,987	108,377	–
(2021)	Error%	0.26	0.00	0.71	1.09	1.08	–
DSMO (2020)	Best	4013	–	44,118	12,218	112,105	–
	Error%	2.48	–	4.97	3.01	4.56	–
NACO	Best	3922	2588	42,258	11,947	107,878	–
(2020)	Error%	0.15	0.35	0.54	0.74	0.62	–
HAACO	Best	–	–	–	11,960	108,730	–
(2020)	Error%	–	–	–	0.83	1.41	–
JCACO	Best	3935	2590	42,399	11,969	108,375	–
(2019)	Error%	0.48	0.42	0.88	0.91	1.08	–
PACO-3opt	Best	–	–	–	11,972	108,482	35,027
(2018)	Error%	–	–	–	0.94	1.12	1.24
HMMA	Best	4074	2879	45,349	12,543	114,095	37,043
(2015)	Error%	5.56	11.33	7.90	5.07	6.41	6.93

populations more effectively. The experimental results show that the pheromone fusion mechanism has been proved to be effective and it can fully exploit the characteristics of each subpopulation and complement advantages among the heterogeneous sub-colonies.

The adaptive communication strategy and cooperative game model are used to further control the pheromone fusion mechanism. The former method based on information entropy can make the communication frequency among populations more adaptively, and the latter can help the population select appropriate communication objects by evaluating the payoff of each union. From the experiment in large-scale TSPs, it illustrates that the improved algorithm can improve the accuracy of solution without affecting the convergence speed of the population and balance the convergence speed and the diversity of the algorithm effectively.

In the future, more types of heterogeneous populations can be used in the solution construction and more pheromone fusion mechanisms can be designed to regulate the pheromone distribution among populations. In addition, except for the evaluation criteria under information entropy in this paper, more methods based on statistics or machine learning can be also introduced to control the interaction frequency of the population. Finally, the game mechanism we proposed in this research also has some certain practical value in the application of ant colony algorithm.

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

References

1. Liu, F.; Zeng, G.: Study of genetic algorithm with reinforcement learning to solve the TSP. Expert Syst. Appl. **36**, 6995–7001 (2009). https://doi.org/10.1016/j.eswa.2008.08.026
2. Mahi, M.; Baykan, Ö.K.; Kodaz, H.: A new hybrid method based on particle swarm optimization, ant colony optimization and 3-opt algorithms for traveling salesman problem. Appl. Soft Comput. J. **30**, 484–490 (2015). https://doi.org/10.1016/j.asoc.2015.01.068
3. Panwar, K.; Deep, K.: Discrete grey wolf optimizer for symmetric travelling salesman problem. Appl. Soft Comput. **105**, 107298 (2021). https://doi.org/10.1016/j.asoc.2021.107298
4. Dorigo, M.; Maniezzo, V.; Colorni, A.: Ant system: optimization by a colony of cooperating agents. IEEE Trans. Syst. Man, Cybern. Part B **26**, 29–41 (1996). https://doi.org/10.1109/3477.484436
5. Dorigo, M.; Gambardella, L.M.: Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans. Evol. Comput. **1**, 53–66 (1997). https://doi.org/10.1109/4235.585892
6. Stützle, T.: Hoos HH (2000) MAX–MIN ant system. Futur. Gener. Comput. Syst. **16**, 889–914 (2000). https://doi.org/10.1016/S0167-739X(00)00043-1
7. Sangeetha, V.; Krishankumar, R.; Ravichandran, K.S.; Kar, S.: Energy-efficient green ant colony optimization for path planning in dynamic 3D environments. Soft Comput. **25**, 4749–4769 (2021). https://doi.org/10.1007/s00500-020-05483-6
8. Ye, K.; Zhang, C.; Ning, J.; Liu, X.: Ant-colony algorithm with a strengthened negative-feedback mechanism for constraint-satisfaction problems. Inf. Sci. (Ny) **406**, 29–41 (2017). https://doi.org/10.1016/j.ins.2017.04.016
9. Ning, J.; Zhang, Q.; Zhang, C.; Zhang, B.: A best-path-updating information-guided ant colony optimization algorithm. Inf. Sci. (Ny) **433**, 142–162 (2018). https://doi.org/10.1016/j.ins.2017.12.047
10. Tseng, H.E.; Chang, C.C.; Lee, S.C.; Huang, Y.M.: Hybrid bidirectional ant colony optimization (hybrid BACO): an algorithm for disassembly sequence planning. Eng. Appl. Artif. Intell. **83**, 45–56 (2019). https://doi.org/10.1016/j.engappai.2019.04.015

11. Olivas, F.; Valdez, F.; Castillo, O.; Gonzalez, C.I.; Martinez, G.; Melin, P.: Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems. Appl. Soft Comput. J. **53**, 74–87 (2017). https://doi.org/10.1016/j.asoc.2016.12.015
12. Tuani, A.F.; Keedwell, E.; Collett, M.: Heterogenous adaptive ant colony optimization with 3-opt local search for the travelling salesman problem. Appl. Soft Comput. **97**, 106720 (2020). https://doi.org/10.1016/j.asoc.2020.106720
13. Miao, C.; Chen, G.; Yan, C.; Wu, Y.: Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm. Comput. Ind. Eng. **156**, 107230 (2021). https://doi.org/10.1016/j.cie.2021.107230
14. Chatterjee, S.; Das, S.: Ant colony optimization based enhanced dynamic source routing algorithm for mobile Ad-hoc network. Inf. Sci. (Ny) **295**, 67–90 (2015). https://doi.org/10.1016/j.ins.2014.09.039
15. Raveendra, K.; Vinothkanna, R.: Hybrid ant colony optimization model for image retrieval using scale-invariant feature transform local descriptor. Comput. Electr. Eng. **74**, 281–291 (2019). https://doi.org/10.1016/j.compeleceng.2019.02.006
16. Wang, Y.; Wang, L.; Chen, G.; Cai, Z.; Zhou, Y.; Xing, L.: An improved ant colony optimization algorithm to the periodic vehicle routing problem with time window and service choice. Swarm Evol. Comput. (2020). https://doi.org/10.1016/j.swevo.2020.100675
17. Hong, T.P.; Tung, Y.F.; Wang, S.L.; Wu, Y.L.; Wu, M.T.: A multilevel ant-colony mining algorithm for membership functions. Inf. Sci. (Ny) **182**, 3–14 (2012). https://doi.org/10.1016/j.ins.2010.12.019
18. Gambardella, L.M.: MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows. New Ideas Optim. (1999)
19. Chu, S.-C.; Roddick, J.F.; Pan, J.-S.: Ant colony system with communication strategies. Inf. Sci. (Ny) **167**, 63–76 (2004). https://doi.org/10.1016/j.ins.2003.10.013
20. Twomey, C.; Stützle, T.; Dorigo, M.; Manfrin, M.; Birattari, M.: An analysis of communication policies for homogeneous multicolony ACO algorithms. Inf. Sci. (Ny) **180**, 2390–2404 (2010). https://doi.org/10.1016/j.ins.2010.02.017
21. Dong, G.; Guo, W.W.; Tickle, K.: Solving the traveling salesman problem using cooperative genetic ant systems. Expert Syst. Appl. **39**, 5006–5011 (2012). https://doi.org/10.1016/j.eswa.2011.10.012
22. Zhang, D.; You, X.; Liu, S.; Yang, K.: Multi-colony ant colony optimization based on generalized jaccard similarity recommendation strategy. IEEE Access. **7**, 157303–157317 (2019). https://doi.org/10.1109/ACCESS.2019.2949860
23. Wang, Y.; Wang, L.; Peng, Z.; Chen, G.; Cai, Z.; Xing, L.: A multi ant system based hybrid heuristic algorithm for vehicle routing problem with service time customization. Swarm Evol. Comput. **50**, 100563 (2019). https://doi.org/10.1016/j.swevo.2019.100563
24. Yang, K.; You, X.; Liu, S.; Pan, H.: A novel ant colony optimization based on game for traveling salesman problem. Appl. Intell. **50**, 4529–4542 (2020). https://doi.org/10.1007/s10489-020-01799-w
25. Li, S.; You, X.; Liu, S.: Multiple ant colony optimization using both novel LSTM network and adaptive Tanimoto communication strategy. Appl. Intell. (2021). https://doi.org/10.1007/s10489-020-02099-z
26. Deng, Y.: Uncertainty measure in evidence theory. Sci. China Inf. Sci. **63**, 1–19 (2020). https://doi.org/10.1007/s11432-020-3006-9
27. Deng, Y.: Information volume of mass function. Int. J. Comput. Commun. Control (2020). https://doi.org/10.15837/ijccc.2020.6.3983
28. Xue, Y.; Deng, Y.: Tsallis eXtropy. Commun. Stat - Theory Methods (2021). https://doi.org/10.1080/03610926.2021.1921804
29. Zhao, J.; Liang, J.M.; Dong, Z.N.; Tang, D.Y.; Liu, Z.: Accelerating information entropy-based feature selection using rough set theory with classified nested equivalence classes. Pattern Recognit. **107**, 107517 (2020). https://doi.org/10.1016/j.patcog.2020.107517
30. Sabirov, D.S.: Information entropy of mixing molecules and its application to molecular ensembles and chemical reactions. Comput. Theor. Chem. **1187**, 112933 (2020). https://doi.org/10.1016/j.comptc.2020.112933
31. Zhu, H.; Wang, Y.; Du, C.; Zhang, Q.; Wang, W.: A novel odor source localization system based on particle filtering and information entropy. Rob. Auton. Syst. **132**, 103619 (2020). https://doi.org/10.1016/j.robot.2020.103619
32. Saji, Y.; Barkatou, M.: A discrete bat algorithm based on Lévy flights for Euclidean traveling salesman problem. Expert Syst. Appl. **172**, 114639 (2021). https://doi.org/10.1016/j.eswa.2021.114639
33. Akhand, M.A.H.; Ayon, S.I.; Shahriyar, S.A.; Siddique, N.; Adeli, H.: Discrete spider monkey optimization for travelling salesman problem. Appl. Soft Comput. J. **86**, 105887 (2020). https://doi.org/10.1016/j.asoc.2019.105887
34. Gülcü, Ş; Mahi, M.; Baykan, Ö.K.; Kodaz, H.: A parallel cooperative hybrid method based on ant colony optimization and 3-Opt algorithm for solving traveling salesman problem. Soft Comput. **22**, 1669–1685 (2018). https://doi.org/10.1007/s00500-016-2432-3
35. Ebadinezhad, S.: DEACO: adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem. Eng. Appl. Artif. Intell. **92**, 103649 (2020). https://doi.org/10.1016/j.engappai.2020.103649
36. Yong, W.: Hybrid Max-Min ant system with four vertices and three lines inequality for traveling salesman problem. Soft Comput. **19**, 585–596 (2015). https://doi.org/10.1007/s00500-014-1279-8
37. Osaba, E.; Ser, J.D.; Sadollah, A.; Bilbao, M.N.; Camacho, D.: A discrete water cycle algorithm for solving the symmetric and asymmetric traveling salesman problem. Appl. Soft Comput. J. **71**, 277–290 (2018). https://doi.org/10.1016/j.asoc.2018.06.047
38. Khan, I.; Maiti, M.K.: A swap sequence based artificial bee colony algorithm for traveling salesman problem. Swarm Evol. Comput. **44**, 428–438 (2019). https://doi.org/10.1016/j.swevo.2018.05.006
39. Ezugwu, A.E.S.; Adewumi, A.O.: Discrete symbiotic organisms search algorithm for travelling salesman problem. Expert Syst. Appl. **87**, 70–78 (2017). https://doi.org/10.1016/j.eswa.2017.06.007
40. Osaba, E.; Yang, X.S.; Diaz, F.; Lopez-Garcia, P.; Carballedo, R.: An improved discrete bat algorithm for symmetric and asymmetric traveling salesman problems. Eng. Appl. Artif. Intell. **48**, 59–71 (2016). https://doi.org/10.1016/j.engappai.2015.10.006