Complex & Intelligent Systems (2022) 8:4679–4696
https://doi.org/10.1007/s40747-022-00716-7

ORIGINAL ARTICLE

# Multi-colony ant optimization with dynamic collaborative mechanism and cooperative game

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

Received: 8 July 2021 / Accepted: 11 March 2022 / Published online: 8 April 2022

## Abstract

Ant Colony Optimization easily falls into premature stagnation when solving large-scale Travelling Salesmen Problems. To address this problem, a multi-colony ant optimization with dynamic collaborative mechanism and cooperative game is proposed. Firstly, *Ant Colony System* and *Max–Min Ant System* form heterogeneous colonies. Secondly, to diversify the solutions of the algorithm, the Shapley value in the cooperative game is applied to share the information by distributing the pheromone payoff of the sub-colonies. In addition, the dynamic collaborative mechanism that contains two methods is designed to enhance the co-evolution of the heterogeneous populations. One, called public path recommendation strategy, is proposed to improve the astringency of *Max–Min Ant System*. The other is the pheromone fusion mechanism to regulate the pheromone distribution of *Ant Colony System* when the algorithm falls into stagnation, which can help the algorithm jump out of the local extremum effectively. Finally, the results demonstrate that the proposed methodology can improve the accuracy of solution effectively in solving large-scale TSP instances and has strong competitiveness with other swarm intelligent algorithms.

**Keywords** Ant colony algorithm · Cooperative game · Shapley value · Dynamic collaborative mechanism · TSP

## Introduction

Traveling salesman problem (TSP) is a classical combinatorial optimization problem that can be described by the weight-directed graph $G = (\upsilon, \Omega, d)$ where $\upsilon = (1, 2, 3..., n)$ denotes the set of the nodes or cities, $\Omega = \{(i, j)|(i, j) \in \upsilon \times \upsilon\}$ represents the set of the edges, and $d : \Omega \mapsto \mathbb{N}$ is the weight function about each edge $(i, j)$. The goal of the problem is to seek the shortest route visiting every city or node exactly once. The well-known methods to solve TSP include genetic algorithm [1], whale optimization algorithm [2], grey wolf optimization algorithm [3], ant colony optimization [4], etc. Due to the advantage of the positive feedback mechanism and distributed computing, ant colony optimization shows excellent performance in solving TSP.

Ant System algorithm (AS), originally proposed by Italian scholar M.Dorigo in the 1990s, is the classical meta heuristic algorithm which imitated ants foraging in nature [5]. The main idea is that ants can communicate with each other effectively by depositing pheromone on the paths between food source and nest. After the AS algorithm proposed, Dorigo [6] also put forward the *Ant Colony System* algorithm (ACS) to overcome its premature stagnation problem. There are two main improvements called global updating rule and local updating rule in ACS. According to the global updating rule, only the optimal ant can be allowed to deposit pheromone in each iteration while other ants would diminish the pheromone trail on the tour that they visited in terms of the local updating rule. However, this mechanism can accelerate the convergence of the algorithm effectively at the cost of falling into the local extremum easily yet. In addition, Stützle et al. [7] introduced the *Max–Min Ant System* algorithm (MMAS) to diversify the solution of the ACO. With the upper and lower restriction in the pheromone matrix, MMAS can avoid algorithm premature stagnation to some certain extent but will be hardly converged when the solutions distribute dispersedly.

✉ Yadong Mo
529634794@qq.com

Xiaoming You
yxm6301@163.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

To improve the performance of the traditional ACO further, many researchers have improved the algorithm in various aspects. Sangeetha et al. [8] improved ACO with a pheromone enhancement mechanism. It strengthened the attraction of the better path so that reduced useless search. Whereas excessively narrowing the search space would decrease the diversity of the algorithm. Ye et al. [9] used the negative feedback pheromone strategy to expand the search space, which achieved exploration of the ant colony in more unknown areas and avoided the premature stagnation of the algorithm. Comparing the best-so-far path with the iteration-best path, Ning et al. [10] dynamically increased the pheromone concentration on the different edges to enhance the diversity of the algorithm. Although these improvements that explored more searching space could diversify the solutions, the convergence speed of the ant colony would be apt to slow down. To accelerate the speed rate of the algorithm, Guan et al. [11] incorporated an automatic updating mechanism into the ant colony algorithm for constraint satisfaction problem (CSP). The main idea of the improvement is to optimize the suboptimal solutions rather than all solutions so that the convergence speed of the algorithm is improved. Liu et al. [12] adaptively updated the pheromone with iteration-best solution, that is called the dynamic weighted pheromone update mechanism, to accelerate the convergence rate of the algorithm. While the algorithm would easily fall into premature stagnation due to the too-fast convergence speed. In addition, the rational assignment of the parameter for ACO is also a baffling dilemma. To address this conundrum, Mahi et al. [13] incorporated the particle swarm optimization algorithm to optimize the parameters of the ACO, which improved the robustness of the algorithm. Olivas et al. [14] introduced the fuzzy control system to assign the appropriate parameters for the ACO algorithm so that a high-quality solution of the algorithm is achieved. Tuani et al. [15] proposed a novel adaptive parameter adjustment mechanism to improve the adaptability of the algorithm. According to these methods [2, 12, 13], the adaptability of the algorithm has been improved to some extent, but the problems that these algorithms solved are relatively simple and it is difficult to deal with more complex problems. Besides, the ant colony algorithm has achieved promising results in other optimization problems, which embraces scheduling problem [16, 17], robot path planning problem [18], geotechnical problems [19], network routing problem [20], and image segmentation [21], etc.

However, although these methods in the single population above have achieved some progress, the improvements of the single population often strengthen one characteristic as well as weaken another. For example, it will explore more search areas at the expend of increasing search time [9, 10] or will diminish the solution accuracy to accelerate the convergence [11, 12]. In order to balance the functions of the convergence speed and diversity of the algorithm further, the multi-colony algorithm gradually attracts many scholars' attention. Gambardella et al. [22] studied the multi ant colony algorithm for the first time. They used two ACS colonies to solve vehicle scheduling problems with time windows. Chu et al. [23] tested the performance of seven interaction strategies to control the communication of the homogenous colonies. Twomey et al. [24] analyzed the homogenous multi-ant colony with different communication policies and proposed migrant integration strategy for the interaction. The cooperation on homogenous populations usually only amplifies the single feature in terms of their same characteristics whereas the heterogeneous populations can take full advantage of each other. Dong et al. [25] combined the ant colony algorithm with the genetic algorithm in a novel way to solve the TSPs successfully. Zhang et al. [26] applied two heterogeneous ant colonies to diversify the solution of the algorithm by exchanging the pheromone information. To solve the Vehicle Routing Problem well, Wang et al. [27] incorporated local search into the multi ant algorithm and exchanged the global optimal solution of each colony, which improved the solution accuracy.

On the basis of these above references [22–27], the multi-ACO has more competitive than the single colony algorithm in balancing the convergence rate and diversity of the population, while the efficiency of this algorithm also needs to be improved due to the insufficient interaction mechanism. To overcome these shortcomings, some cross-discipline methods are applied to improve the performance of the multi colony algorithm. Yang et al. [28] incorporated the game theory to solve the decision conflicts of each sub-colony, which can promote the efficiency of the multi-colony algorithm. Li et al. [29] used the recommendation algorithm to adapt the communication among populations more accurately.

Compared with a single colony algorithm, the multi-ACO algorithm can achieve better results in small or middle scale TSP instances due to the coordination among multiple colonies. Whereas the communication mechanism, the key of the multi-colony algorithm, is relatively simple, causing that the satisfied results can be hardly achieved when solving large-scale TSP instances. To address this issue, the multi-colony ant optimization with dynamic collaborative mechanism and cooperative game is proposed to strengthen the interactive mechanism and take full advantage of the heterogenous populations. The main contributions and innovations of this research are as follows.

First, to improve the solution accuracy of the algorithm in solving large-scale TSP instances, the cooperative game based on the Shepley value is introduced. It redistributes the pheromone payoff to each participant with respect of their contribution after each iteration so that more useful information can be explored during the searching process and a diversification of the solution can be achieved.

Second, the dynamic collaborative mechanism is applied to improve the interactive mechanism of the heterogenous populations and the mechanism contains two methods. One is called the public path recommendation strategy for MMAS population, which recommends the public paths of ACS populations to MMAS population and improves the astringency of the population. The other is the pheromone fusion mechanism based on information entropy. The aim of this method is to help ACS population jump out of the local extremum by regulating the pheromone distribution when the algorithm falls into stagnation.

Finally, the remainder of this paper are as follows. Section "Related work" reports briefly the basic ACS, MMAS algorithm, information entropy and Shapley value. Section "Proposed algorithm" describes the DCM-ACO algorithm, including dynamic collaborative mechanism and cooperative game model. Section "Experiment analysis" gives the analysis about the performance of the proposed strategies and compares DCM-ACO with the traditional ant colony algorithm and other intelligent algorithms. Section "Conclusion" summarizes and prospects this research.

## Related work

### Ant Colony system

*Ant Colony system* (ACS) algorithm mainly contains two-state transition rules. One rule, that the ants construct their paths, is called roulette mode if the random number q is higher than the constant parameter $q0$, where $q$ and $q0$ are between [0,1]. The state transition formula is as follows.

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

where the $\tau_{ij}$ 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; $\alpha$ and $\beta$ are two weight parameters determining the influence of pheromone value and heuristic information. In addition, $S = \text{argmax}\left(\tau_{ij} \cdot \eta_{ij}^{\beta}\right)$ is another transition formula that the ants positioned city $i$ move to city $j$ when $q < q0$.

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 $\boldsymbol{l_n}$ is the tour length created by the nearest neighbour heuristic algorithm.

When all ants complete path construction, the global pheromone updating rule is triggered to add pheromone on the global best tour. The formula is written as:

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

$$\Delta\tau_{ij}^{\text{bs}} = 1/L_{\text{gb}} \tag{4}$$

where $0 < \rho < 1$ is pheromone evaporation rate; $\Delta\tau_{ij}^{\text{bs}}$ is the value of increasing pheromone and $L_{\text{gb}}$ is the length of the best tour.

### Max-Min Ant system

To improve the diversity of the traditional ant colony algorithm, Stützle proposed the *Max–Min Ant System* (MMAS) [7]. The pheromone of MMAS is updated by an alternating iteration-best tour with best-so-far tour in the early run time, which the updating rule is defined as same as formula (3). Besides, the pheromone matrix of the ant colony 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:

$$\tau_{\max} = (1/\rho) \cdot \left(1/L_{\text{gb}}\right) \tag{5}$$

$$\tau_{\min} = \tau_{\max}/(2n) \tag{6}$$

where $n$ is the city number; $L_{\text{gb}}$ is the length of the global-best tour.

### Information entropy

Information Entropy was the firstly proposed by the American scholar *Shannon*, which greatly promoted the progress of the information theory [30]. For now, information entropy has been applied in many fields and has achieved good results [31–33]. And it shows the effectiveness of information entropy as a measure of the discrete system. the formula is as follows:

$$E(P) = -\sum_{x \in X} P(x) \log p(x) \tag{7}$$

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

### Shapley value

Shapley value, a way of profit distribution, was originally proposed by scholar *Lloyd Shapley* in 1953 [34]. The main idea of this method is that the income of each member in the union is proportional to their own contribution. And it has promoted the further development of the cooperative game theory, which the formula is given as follows.

$$\varphi_i(v) = \sum_{s \subset K} w(s)[v(s) - v(s - i)] \tag{8}$$

$$w(s) = \frac{(s-1)!(k-s)!}{k!} \tag{9}$$

where the $K$ denotes the union, $v$ represents the payoff function of each sub-union, *w(s)* is the weight function of each union and $\boldsymbol{\varphi_i(v)}$ is the payoff function of the $i$-th player in the union.

## Proposed algorithm

In this research, we focus on improving the interactive mechanism and solving the large-scale TSP instances well. The proposed algorithm including cooperative game and dynamic collaborative mechanism is designed. First, in the cooperative game model, a novel pheromone accumulation is designed among ACS sub-colonies. By applying Shapley value, the ant colonies updated the pheromone with respect to their contribution, which more useful knowledge about the problem can be explored during the searching process. Secondly, the dynamic collaborative mechanism, including two methods, is used to improve the interactive mechanism of heterogeneous colonies. By using the public path recommendation strategy, the solution that belongs to the ACS colonies would be strengthened in the MMAS, which can accelerate the convergence speed of the MMAS. While the pheromone matrix of the ACS can be reset when the searching falls into stagnation due to the pheromone fusion mechanism. These two methods can take full advantage of heterogeneous colonies. And the interactive model is shown in Fig. 1.

Besides, this part is organized as follows. Section "The cooperative game strategy" is dedicated to applying the Shapley value to the multi-ACO algorithm based on the cooperative game theory. Section" Dynamic collaborative mechanism" provides the dynamic collaborative mechanism in detail that includes pheromone fusion mechanism and the public path recommendation strategy. Section "Algorithm description" is the algorithm description.

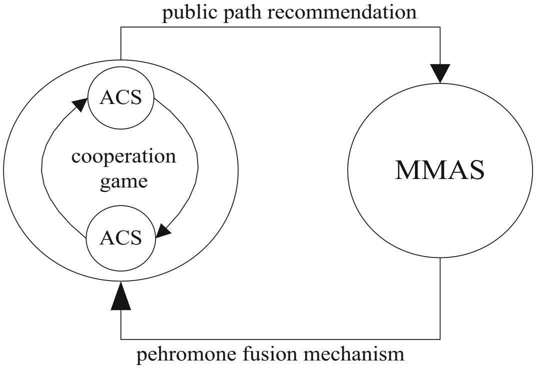


**Fig. 1** Dynamic interactive game model

### Cooperative game strategy

The cooperative game strategy among ACS populations is mainly to ensure the accuracy of the algorithm. In contrast to other intelligent algorithms, the advantage of the ant colony algorithm is mainly depending on the positive feedback mechanism which performs the solution optimization through accumulating useful information after each iteration. Corresponding to this mechanism in ACO, the global pheromone update rule, as shown in Eqs. (3), (4), plays a key role in ACS. However, too fast convergence speed that created by this mechanism would easily bring premature stagnation of the algorithm. Therefore, a reasonable regulation of pheromone accumulation is the key to the ant colony algorithm.

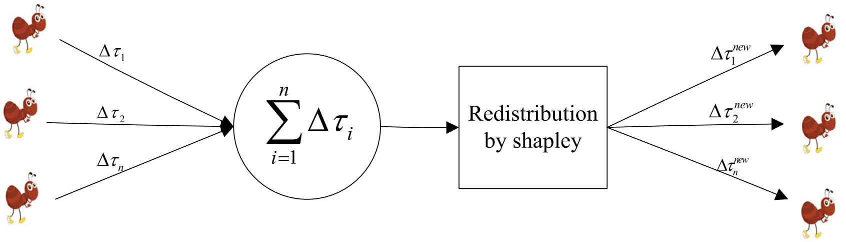
In this research, the Shapley value based on the cooperative game is introduced to regulate the pheromone accumulation after each iteration. Specifically, before performing the global update rule, we collect the pheromones of each subpopulation so that obtain the total pheromone income $b$, as shown in the following formula (10). And then the Shapley formula is introduced to allocate those total income to each member with respect of their contribution, which is shown in Fig. 2. The aim of this improvement is to regulate the pheromone accumulation of each subpopulation so that makes the pheromone allocation more reasonable, which achieves a diversification of research. The specific distribution formulas are constructed as follows.

$$b = \sum_{i=1}^{k} \Delta\tau^{i}_{\mathrm{ACS}} \tag{10}$$

$$\mathrm{Contribution}_i = \mathrm{sol}_i \cdot H(p_i) \tag{11}$$

and $\mathrm{sol}_i = \frac{\mathrm{length}_{\min}}{\mathrm{length}_i}$, $H(p_i) = \frac{E(p_i)}{E(p_{\max})}$. where $\mathrm{length}_{\min}$ and $E(p_{\max})$ are the minimum length and the maximum value of information entropy respectively among $k$ colonies of ACS, and $sol_i$ denotes the tour length ratio, $H(p_i)$ represents the

**Fig. 2** Cooperative game strategy



diversity ratio of each subpopulation. It can be concluded that the contribution of the colony is the value between [0,1], which the closer the value is to 1, the larger the contribution of the population. And the pheromones are redistributed as follows.

$$\pi_i = \frac{\text{Contribution}_i}{\sum_{i=1}^{k}\text{Contribution}_i} \tag{12}$$

$$\Delta\tau_i^{\text{new}} = \pi_i \cdot b \tag{13}$$

where $\Delta\tau_i^{\text{new}}$ is the new pheromone revenue of the $i$-th ACS population.

From equal (11), it can be concluded that in the initial stage of the algorithm, each population is in an exploration state causing that the difference of diversity among sub-colonies is not obvious, thus the contribution of the population is proportional to the quality of its solution. While in the late stage that is called the exploitation stage, the difference of the solution quality has not competitiveness at this stage, so that the contribution of the population is mainly depending on the diversity, corresponding to the information entropy in this research. Besides, the role of the pheromone redistribution strategy, as shown in equal (12), is to regulate the pheromone accumulation of each sub-colony with respect to their contribution in different stages. The aim is to reward the elite population and punish the poor performance population and thus to diversify the solution of the algorithm effectively.

## Dynamic collaborative mechanism

In this part, we apply a dynamic collaborative mechanism to take full advantages of the heterogenous population. And the mechanism contains two methods: one is the pheromone fusion mechanism which fuses the pheromone of ACS population and MMAS population based on information entropy to help the algorithm get rid of local minima, and the other is public path recommendation strategy that recommends the public paths from ACS population to MMAS population to improve the convergence rate of the MMAS population.

### Pheromone fusion mechanism

**Diversity description** On the basic of part 2.3, the information entropy is worthy of measuring the diversity of the algorithm. And the diversity of the population based on information entropy can be described as follows:

$$P_i(t) = n/m \tag{14}$$

$$E(\text{Pt}) = -\sum_{i=1}^{m} P_i(t)\log p_i(t) \tag{15}$$

where $P_i(t)$ is the proportion that the $i$-th tour trail selected by $n$ ants when $M$ ants generate $m$ paths in this iteration, $E(\text{Pt})$ 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 detecting the information entropy of each subpopulation, the state of the algorithm can be described more accurately, thus the adaptability of interaction is improved.

**Pheromone fusion strategy** This strategy proposed in this part is main for ACS colonies. Due to the pheromone update mechanism of MMAS, the pheromone distribution in the population has been limited in a certain range, which prevents the population from falling into local optimum effectively. While ACS will easily fall into premature stagnation in terms of the global updating rule although it can speed up the convergence of the algorithm. Under these circumstances, the pheromone fusion strategy is proposed to regulate the pheromone distribution in ACS and diversify the solution of the population, which helps ACS get rid of the local extremum. When the entropy of the ACS colony is lower than the threshold $(E(p_i) < E^*(P)$, where the $E^*(P)$ is the threshold parameter), that means the diversity of the population has been decreased, then the pheromone fusion strategy is triggered. And the formula is as follows.

$$\text{ph}_{\text{ACS}}^{i} \leftarrow (1-\omega_i)\cdot \text{ph}_{\text{ACS}}^{i} + \omega_i \cdot \text{ph}_{\text{MMAS}} \tag{16}$$

where $\mathrm{Ph}_i$ is the pheromone matrix of population $i$, $\mathrm{Ph}_j$ is the pheromone matrix of population $j$; $W_j$ is the pheromone contribution of population $j$ to population $i$, which the formula can be written as follows.

$$\omega_i = \frac{E(p)^i_{\text{ACS}}}{E(P)^i_{\text{ACS}} + E(P)_{\text{MMAS}}} \tag{17}$$

where $E(P_i)$ is the information entropy of the population $i$.

### Public path recommendation strategy

The strategy in this section is mainly for MMAS population. Although the MMAS population usually has good diversity due to the pheromone restriction, the slow convergence speed is a significant disadvantage of the colony. To address this issue, we propose a public path recommendation strategy, as shown in Fig. 3, to recommend the excellent paths from ACS population to MMAS population, which improves the astringency of the population. Generally speaking, it can be considered that the path selected by multiple populations together is a part of the optimal global path or there is an optimal path near this path. Therefore, we recommend the common paths among ACS populations to MMAS population and reward the pheromone on them when the convergence speed of MMAS population is lower. Thus, it will avoid the useless search of population, save the cost of the solution and accelerate the convergence speed of the algorithm. And the formulas are constructed as follows.

$$\mathrm{con}_t = \frac{\mathrm{iter}_{\mathrm{opt}}}{\mathrm{iter}_t} \tag{18}$$

where $\mathrm{con}_t$ represents the convergence rate of the population at the $t$-th iteration, $\mathrm{iter}_{\mathrm{opt}}$ is the number of iterations corresponding to the current optimal solution of the population, and $\mathrm{iter}_t$ denotes the number of the $t$-th iteration. From the formula, its range is [0,1] and the closer its value is to 1, the faster convergence speed of the population. And the public path recommendation strategy is triggered when the convergence rate of the MMAS is lower than threshold ($\mathrm{con}_t < \mathrm{con}^*$), which can be written as follows.

$$\tau_{\text{public}} = \frac{1}{n} \cdot e^{-\text{iter}} \tag{19}$$

$$\tau_{ij}^{\text{public}} \leftarrow \tau_{ij}^{\text{public}} + \tau_{\text{public}} \tag{20}$$

where $\tau_{\text{public}}$ is the added pheromone of the common route between ACS populations, $n$ is the city number, iter is the number of the iteration. and in formular (14), $\tau_{ij}^{\text{public}}$ denotes the pheromone on the paths that belongs to the common part of ACS populations.

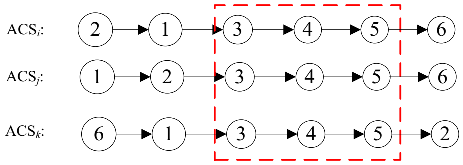


**Fig. 3** Public path

From the formula (13) and (14), we can see that the pheromones are rewarded decreasing with the run of iteration, which accelerates the convergence speed of MMAS population in the early state and improve the diversity in the late stage of the algorithm.

## Algorithm description

The proposed algorithm in this research mainly contains two ACS populations and one MMAS population and each subpopulation construct the path in parallel. Firstly, initializing the parameters among ACS and MMAS colonies in the initial stage of the algorithm. Then, the dynamic collaborative mechanism is triggered to judge the state of each colony. If the information entropy of ACS population is lower than the threshold, the pheromone fusion mechanism is introduced to regulate their pheromone distribution and help the algorithm jump out of the local optimum. Otherwise, the cooperative game strategy is applied to re-update the global pheromone among the ACS populations by the Sharpley value, which enhances the diversity of the algorithm. Meanwhile, the public path recommendation strategy is utilized when the convergence rate of MMAS population is too low. And it recommends the public paths among ACS populations to MMAS population to accelerate its convergence. And the algorithm flowchart is shown in Fig. 4 in detail.

**Algorithm 1** DCM-ACO algorithm for TSP

```
Procedure DCM-ACO ()
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)
  obtain the information entropy of ACS by equal (15)
  If the information entropy < threshold
    execute the pheromone fusion by equal (16)
  else
    execute the cooperative game by equal (10)-(13)
    update the pheromone of ACS by equal (3)
  End if
  obtain the convergence speed of MMAS by equal (7)
  If the convergence speed < threshold
    execute the public path recommendation strategy by equal (19) - (20)
  else
    update the pheromone of MMAS by equal (3)
  End if
End while
End while
Return best solution
```

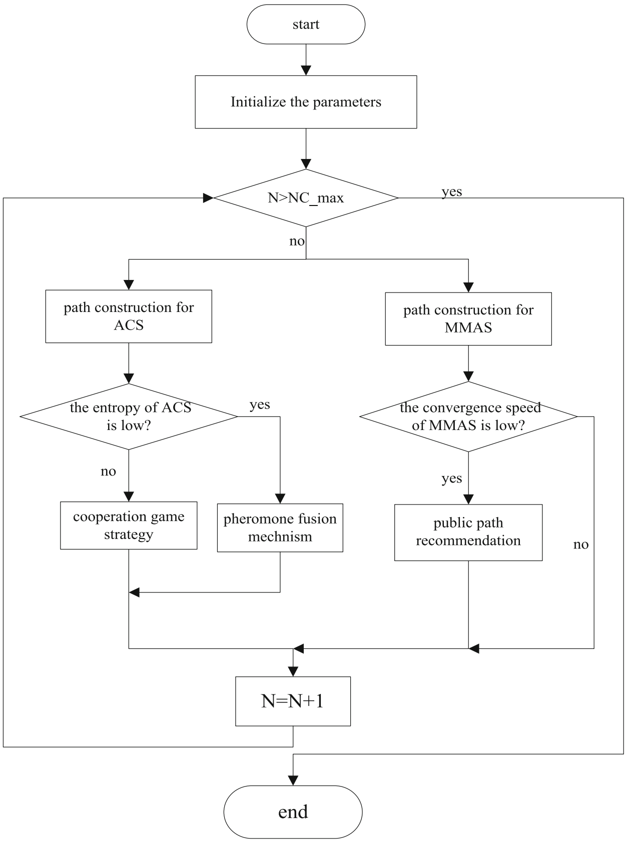

start
Initialize the parameters
N>NC_max
yes
no
path construction for ACS
path construction for MMAS
the entropy of ACS is low?
yes
no
the convergence speed of MMAS is low?
yes
no
cooperation game strategy
pheromone fusion mechnism
public path recommendation
N=N+1
end

**Fig. 4** The flow chart of DCM-ACO algorithm

## Experiment analysis

In this part, we implement the simulation experiment to verify the performance of the proposed algorithm. 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. In addition, the rest of this part is as follows. Section "Parameters setting" selects the appropriate parameters for ACS, MMAS and DCM-ACO. Section "Strategy analysis" is given the analysis about the strategy we have proposed. Section "Comparison with traditional ACO algorithm" compares the proposed algorithm with the conventional ACO algorithms. Section 4.4 is the experimental comparison between DCM-ACO and other swarm intelligence algorithms. Section "Statistical test analysis" is the statistical test analysis of the experimental results.

### Parameters setting

The first part of this section is the traditional parameters setting. And we apply the orthogonal tests that contain four levels and five factors to set the appropriate value for those parameters. Besides, the kroB100 instance is adopted to carry out for each colony in the orthogonal test. Each combination of the parameters experiment is tested 20 times independently to ensure the reliability of the experiment. Tables 1, 2, 3 are the experimental results of ACS and Tables 4, 5, 6 denote the results of MMAS.

According to the sum of experimental results at each level $H_i$ and the average of each level $h_i$, the parameters $\alpha$, $\beta$, $\rho$ and $q0$ obtain more excellent results at Level 1, Level 3, Level 1, Level 3, and Level 3 respectively in the orthogonal test of ACS, which are shown the bold numbers in Table 3. While in the orthogonal test of MMAS, the experimental results of the factors $\alpha$, $\beta$ and $\rho$ at Level 1, Level 4 and Level 1 respectively, corresponding to the bold numbers in Table 6, are superior to them at other Levels. Therefore, the optimal parameter of ACS algorithm is:$\alpha = 1$, $\beta = 4$, $\rho = 0.1$, $\xi = 0.3$, $q0 = 0.8$, the optimal parameter of MMAS algorithm is: $\alpha = 1$, $\beta = 5$, $\rho = 0.1$

The second part of this parameter experiment is the newly proposed parameter setting which concludes the information entropy threshold ($E(p)*$) and convergence rate threshold (con*). In this paper, we select the suitable value of the newly proposed parameter through the experiment of the different values. Besides, the kroB100 and kroA200 instances are selected to carry out the experiment, which the results are shown in Figs. 5 and 6. From the experimental results, the minimum fitness function is obtained under the case that $E(p)* = 4$ and con$* = 0.8$.

From the above experimental results, the final setting results of the algorithm parameters are shown in the following Table 7, which the $\rho$ denotes the global pheromone evaporation rate, $\zeta$ is the local pheromone evaporation rate and $M$ represents the number of ants.

### Strategy analysis

In the first phase of the experiment, we analyse the effectiveness of three strategies proposed above including cooperative game mechanism, pheromone fusion mechanism and public path recommendation strategy. LOST-1 is the algorithm that has a pheromone fusion mechanism and public path recommendation strategy but does not use a cooperative game mechanism. LOST-2 is the algorithm that retains cooperative game mechanism and public path recommendation strategy but does not use pheromone fusion mechanism. LOST-3 represents the algorithm that has a cooperative game mechanism and pheromone fusion mechanism but does not use a public path recommendation strategy. In the experiment, kroB100 and lin318 TSP instances are selected and analysed with three aspects including optimal solution error rate, worst solution and average solution. And each instance runs 20 times, 2000

**Table 1** Experimental factors and levels of ACS

| | $\alpha$ | $\beta$ | $\rho$ | $\zeta$ | $q0$ |
|---|---|---|---|---|---|
| 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 |

**Table 2** Orthogonal test scheme and test results of ACS

| No | $\alpha$ | $\beta$ | $\rho$ | $\zeta$ | $q0$ | 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 3** Analysis of test results of ACS

| | $\alpha$ | $\beta$ | $\rho$ | $\zeta$ | $q0$ |
|---|---|---|---|---|---|
| $H_1$ | **89,738.6** | 90,710.6 | **89,864.9** | 92,373.6 | 90,132.5 |
| $H_2$ | 89,793.1 | 92,144.8 | 89,985.7 | 89,969.8 | 90,277.1 |
| $H_3$ | 90,648.7 | **89,744.1** | 92,551.8 | **89,750.6** | **90,094.5** |
| $H_4$ | 92,244.9 | 89,825.9 | 90,023.0 | 90,331.4 | 91,921.3 |
| $h_1$ | **22,434.7** | 22,677.6 | **22,466.2** | 23,093.4 | 22,533.1 |
| $h_2$ | 22,448.3 | 23,036.2 | 22,496.4 | 22,492.5 | 22,569.3 |
| $h_3$ | 22,662.2 | **22,436.0** | 23,138.0 | **22,437.6** | **22,523.6** |
| $h_4$ | 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 | 22,437.6 | 22,523.6 |
| Range | 626.5 | 600.2 | 678.1 | 655.8 | 456.7 |
| Scheme | Level 1 | Level 3 | Level 1 | Level 3 | Level 3 |

Bold values indicate the best results between the comparison algorithms and the proposed algorithm

Note: $H_i (i = 1, 2, 3, 4)$ represents the sum of experimental results at each level; $h_i (i = 1, 2, 3, 4)$ 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

**Table 4** Experimental factors and levels of MMAS

| | $\alpha$ | $\beta$ | $\rho$ |
|---|---|---|---|
| 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 5** Orthogonal test scheme and test results of MMAS

| No | $\alpha$ | $\beta$ | $\rho$ | 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 6** Analysis of test results of MMAS

| | α | β | ρ |
|---|---|---|---|
| $H_1$ | **90,054.3** | 95,187.8 | **91,036.1** |
| $H_2$ | 90,583.7 | 91,513.7 | 91,342.6 |
| $H_3$ | 92,858.9 | 90,884.7 | 92,599.3 |
| $H_4$ | 94,364.7 | **90,275.3** | 92,883.5 |
| $h_1$ | **22,513.6** | 23,797.0 | **22,759.0** |
| $h_2$ | 22,645.9 | 22,878.4 | 22,835.7 |
| $h_3$ | 23,214.7 | 22,721.2 | 23,149.8 |
| $h_4$ | 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 4 | Level 1 |

Bold values indicate the best results between the comparison algorithms and the proposed algorithm

Note: $H_i (i = 1, 2, 3, 4)$ represents the sum of experimental results at each level; $h_i (i = 1, 2, 3, 4)$ 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.

iterations each time. The experimental results are shown in Table 8 and Fig. 7.

From the experimental results, in kroA100 instance, all the experimental groups gain the optimal solution, and in lin318 instance the optimal solution of the DCM-ACO outperforms other algorithms. Besides, the average solution and worst solution of LOST-3 are superior to the LOST-1 and LOST-2, which proves that the cooperative game strategy and the pheromone fusion mechanism can enhance the stability of the algorithm and improve the solution quality. But from Fig. 7, the convergence speed of LOST-3 is slower than other comparison algorithms showing that the public path recommendation strategy can accelerate the convergence of the population. In a word, the three novel methods we proposed above can improve the performance of the algorithm in various aspects.

## Comparison with traditional ACO algorithm

In this part, we select different TSP instances to verify the performance of the proposed algorithm in this research compared with traditional ACOs and the analysis aspects 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:

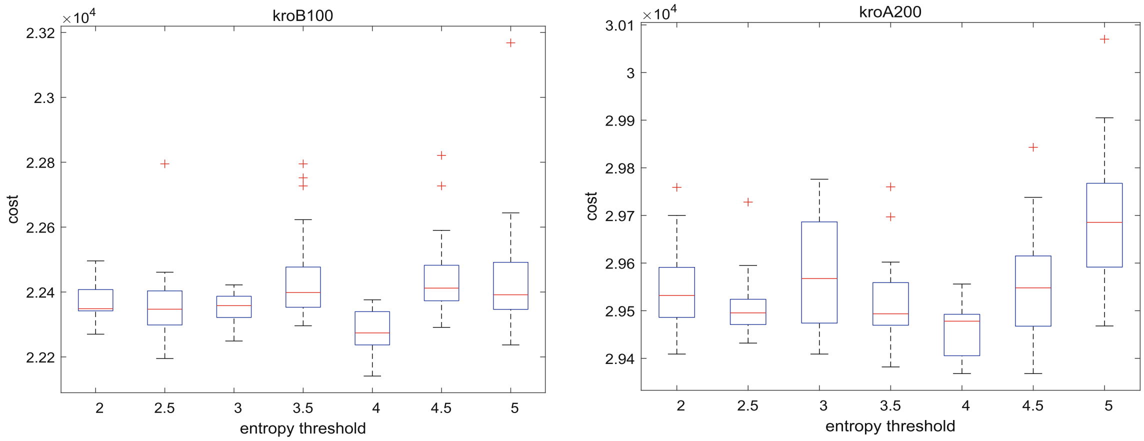
$$\text{Error} = \frac{L_{\text{ACO}} - L_{\text{opt}}}{L_{\text{opt}}} \times 100\% \tag{21}$$

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

$$\text{std} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(L_i - \overline{L}\right)^2} \tag{22}$$

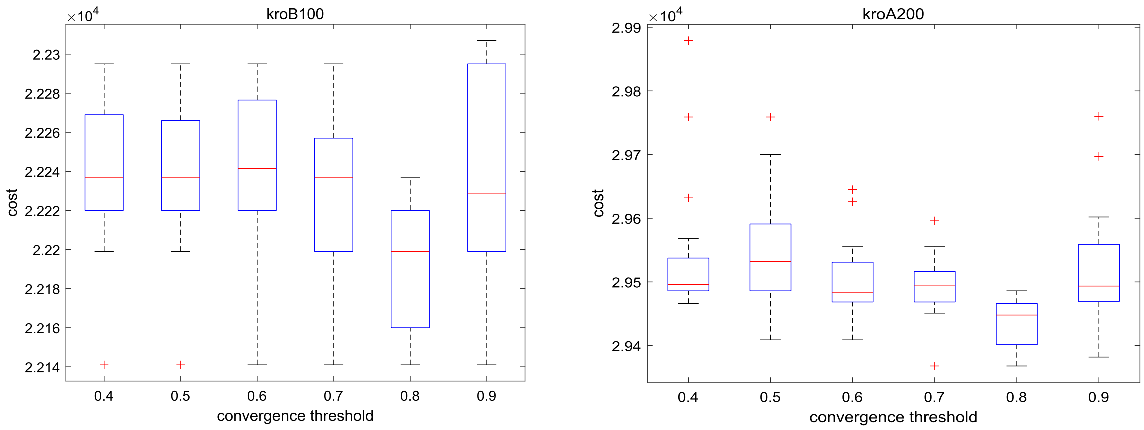
where std is the standard deviation, $N$ denotes the number of times the algorithm runs, and$L_i$ represents the solution obtained by the algorithm in the $i$-th experiment.

Table 9 shows the experimental results of ACS, MMAS and DCM-ACO algorithm. In small-scale instances of less than 300 cities, although the gap of the error rate between DCM-ACO and the comparison algorithm is not obviously, the standard deviation in DCM-ACO is superior to ACS and MMAS due to the coordination of multiple colonies, showing that our proposed algorithm has a more stable ability. When solving middle-scale instances (city scale within 600), DCM-ACO still remains with higher solution accuracy. And in large-scale TSP instances from 1000 to 2000 cities, the search space is so complex that traditional ant colony algorithms can hardly deal with well, corresponding



(a) kroB100 (b) kroA200

**Fig. 5** Adjustment of the entropy threshold



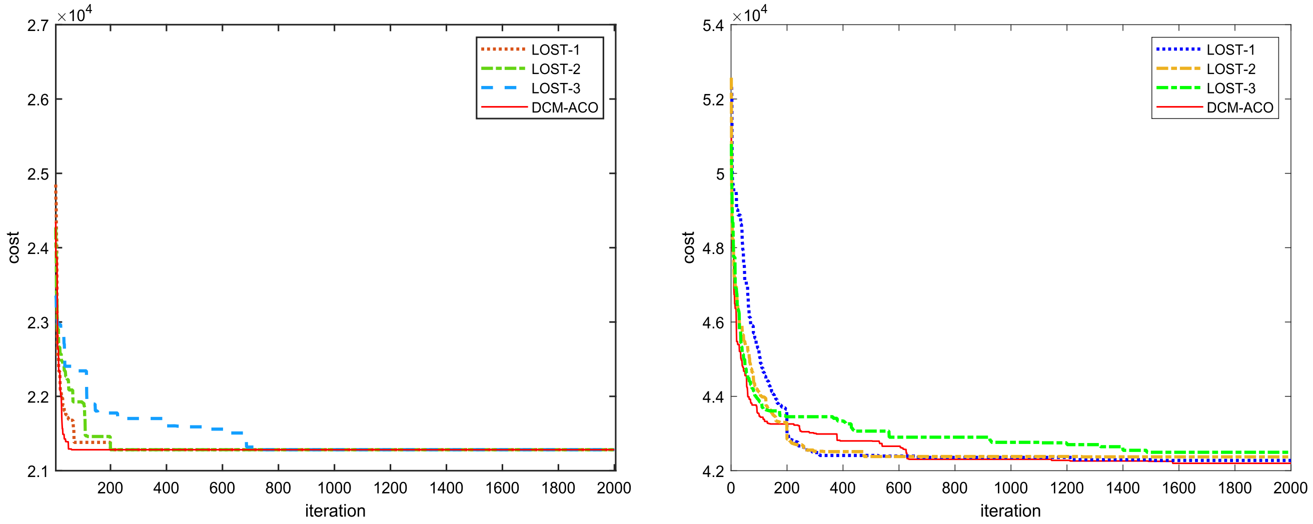
(a) kroB100. . (b) kroA200

**Fig. 6** Adjustment of the convergence threshold

to the huge error rate of ACS and MMAS in Table 9. However, the error computed based on DCM-ACO is controlled within 2%. These excellent results are mainly ascribed to the cooperative game strategy. With the help of this strategy, the pheromone accumulation of populations after each iteration updates more adaptively with respect to their contributions, so that a diversification of the search is achieved. As a result, the performance of the algorithm is improved especially for large-scale instances. For example, in d2103 instance, ACS and MMAS have huge deviations that the error rate is 4.00% and 5.73% respectively while the error rate found by DCM-ACO is just 1.87%, which greatly outperforms the traditional ACO algorithms. Besides, attribute to the dynamic collaborative mechanism, the interactive frequency of heterogeneous populations is more adaptive and the efficiency of the communication among sub-colonies can be improved greatly so that a strong robustness of the algorithm is acquired. And from the average solution and standard deviation, all the values of these factors obtained by DCM-ACO from small-scale instances to large-scale instances are lower than that based on the comparison algorithms, which proves the strong stability of the proposed algorithm for TSP.

**Table 7** The parameter setting of the algorithm

| Parameter | ACS | MMAS |
|---|---|---|
| $\alpha$ | 1 | 1 |
| $\beta$ | 4 | 5 |
| $\rho$ | 0.1 | 0.1 |
| $\zeta$ | 0.3 | – |
| $q0$ | 0.8 | – |
| $M$ | 20 | 20 |
| Iteration | 2000 | 2000 |
| $E(P)$* | 4 | – |
| Con* | – | 0.8 |



(a) kroA100

(b) lin318

**Fig. 7** Comparison results in different strategy

**Table 8** Experiment results with a different strategy

| Instance | Algorithm | Best | Worst | Average |
|---|---|---|---|---|
| kroA100 | LOST-1 | **21,282** | 21,600 | 21,347.40 |
| | LOST-2 | **21,282** | 21,577 | 21,328.47 |
| | LOST-3 | **21,282** | **21,373** | 21,307.60 |
| | DCM-ACO | **21,282** | **21,373** | **21,294.13** |
| lin318 | LOST-1 | 42,276 | 43,407 | 42,890.20 |
| | LOST-2 | 42,454 | 43,423 | 42,885.75 |
| | LOST-3 | 42,372 | 43,338 | 42,776.35 |
| | DCM-ACO | **42,179** | **43,002** | **42,651.00** |

Bold values indicate the best results between the comparison algorithms and the proposed algorithm

Figure 8 illustrates the evolution of the best solution calculated with ACS, MMAS and DCM-ACO in the optimization process for pr439, p654 and fl1400. Due to the cooperative game strategy, the higher quality of the solution can be strengthened in the early stage, while the information with better diversity plays an important role in the late stage. It means that the improved algorithm has better exploration capacity in the initial stage and owns the strong exploitation ability in the late stage. As shown in Fig. 8, DCM-ACO outperforms ACS and MMAS in the initial stage. And there is a tremendous gap between the results calculated by these three algorithms. According to the dynamic collaborative mechanism, the advantages of heterogeneous populations can be fully exploited. Due to the public recommendation strategy, the convergence of MMAS can be accelerated and with the help of the pheromone fusion mechanism, ACS can jump out of the local optima effectively. As we can see in Fig. 8, for each instance, DCM-ACO completes the convergence in a short cycle whereas ACS and MMAS converge to the best solution in a larger number of iterations, and the solution accuracy is also superior to the comparison algorithm. In contrast to ACS and MMAS, DCM-ACO can accelerate the convergence without losing high-quality solutions.

In a word, from those results, it can be concluded that DCM-ACO can balance the convergence speed and the diversity of the algorithm. In addition, to verify the authenticity of the experimental results, the optimal tours found by CGMACO are shown in Fig. 9.

## Comparison with other algorithms

The first phase of this section is the comparison of the proposed algorithm with SOS-ACO [4] and PSO-ACO-3opt [13], which SOS-ACO is the ACO with hybrid symbiotic organisms search algorithm and PSO-ACO-3opt is the hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms. And the bold number in Tables 10 and 11 denotes the minimum value of each index calculated by these algorithms.

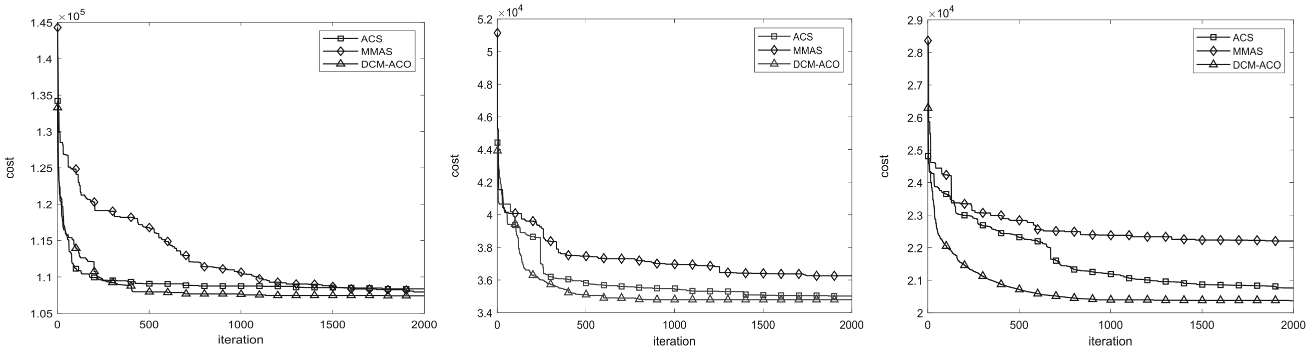
**Table 9** Experimental results of DCM-ACO

| No | TSP | Optimal | Algorithm | Best | Worst | Average | Error (%) | Std (%) |
|---|---|---|---|---|---|---|---|---|
| | | | ACS | **426** | 435 | 428.6 | **0.00** | 2.18 |
| 1 | eil51 | 426 | MMAS | **426** | 432 | 428.1 | **0.00** | 2.04 |
| | | | **DCM-ACO** | **426** | **427** | **426.5** | **0.00** | **0.51** |
| | | | ACS | **538** | 551 | 544.9 | **0.00** | 4.16 |
| 2 | eil76 | 538 | MMAS | **538** | 552 | 542.2 | **0.00** | 4.71 |
| | | | **DCM-ACO** | **538** | **541** | **538.7** | **0.00** | **1.06** |
| | | | ACS | **21,282** | 21,835 | 21,450.6 | **0.00** | 157.83 |
| 3 | kroA100 | 21,282 | MMAS | **21,282** | 21,945 | 21,396.3 | **0.00** | 166.59 |
| | | | **DCM-ACO** | **21,282** | **21,373** | **21,290.3** | **0.00** | **20.30** |
| | | | ACS | 22,270 | 22,496 | 22,366.4 | 0.58 | 53.72 |
| 4 | kroB100 | 22,141 | MMAS | 22,220 | 22,580 | 22,320.3 | 0.36 | 90.47 |
| | | | **DCM-ACO** | **22,141** | **22,275** | **22,167.9** | **0.00** | **38.13** |
| | | | ACS | 6162 | 6348 | 6245.0 | 0.85 | 57.94 |
| 5 | ch130 | 6110 | MMAS | 6121 | 6259 | 6188.4 | 0.18 | 37.35 |
| | | | **DCM-ACO** | **6110** | **6218** | **6151.4** | **0.00** | **20.65** |
| | | | ACS | 6548 | 6778 | 6600.5 | 0.31 | 54.02 |
| 6 | ch150 | 6528 | MMAS | **6528** | 6641 | 6578.3 | **0.00** | 29.55 |
| | | | **DCM-ACO** | **6528** | **6563** | **6546.4** | **0.00** | **11.86** |
| | | | ACS | 26,244 | 27,269 | 26,699.1 | 0.43 | 292.44 |
| 7 | kroB150 | 26,130 | MMAS | 26,196 | 27,034 | 29,443.9 | 0.25 | 214.31 |
| | | | **DCM-ACO** | **26,130** | **26,463** | **26,294.5** | **0.00** | **75.35** |
| | | | ACS | 29,536 | 30,290 | 29,827.9 | 0.57 | 233.57 |
| 8 | kroA200 | 29,368 | MMAS | 29,460 | 30,184 | 29,639.0 | 0.31 | 213.40 |
| | | | **DCM-ACO** | **29,368** | **29,608** | **29,494.6** | **0.00** | **59.89** |
| | | | ACS | 29,737 | 30,701 | 30,343.6 | 1.01 | 272.47 |
| 9 | kroB200 | 29,437 | MMAS | 29,721 | 30,957 | 30,195.9 | 0.96 | 405.92 |
| | | | **DCM-ACO** | **29,437** | **30,081** | **29,653.2** | **0.00** | **136.01** |
| | | | ACS | 49,198 | 51,702 | 49,734.4 | 0.13 | 671.45 |
| 10 | pr264 | 49,135 | MMAS | **49,135** | 51,927 | 49,715.0 | **0.00** | 786.85 |
| | | | **DCM-ACO** | **49,135** | **49,203** | **49,163.4** | **0.00** | **25.01** |
| | | | ACS | 2589 | 2712 | 2636.6 | 0.38 | 35.77 |
| 11 | a280 | 2579 | MMAS | 2587 | 2683 | 2624.8 | 0.31 | 27.06 |
| | | | **DCM-ACO** | **2579** | **2608** | **2596.2** | **0.00** | **8.68** |
| | | | ACS | 42,790 | 43,586 | 43,277.0 | 1.81 | 221.68 |
| 12 | lin318 | 42,029 | MMAS | 43,046 | 44,879 | 43,614.5 | 2.42 | 447.74 |
| | | | **DCM-ACO** | **42,179** | **42,915** | **42,638.9** | **0.35** | **177.35** |
| | | | ACS | 12,031 | 12,404 | 12,185.9 | 1.43 | 100.58 |
| 13 | fl417 | 11,861 | MMAS | 12,006 | 12,363 | 12,174.1 | 1.22 | 94.02 |
| | | | **DCM-ACO** | **11,901** | **12,042** | **11,955.6** | **0.33** | **29.20** |
| | | 107,217 | ACS | 108,309 | 116,846 | 110,905.9 | 1.02 | 2402.69 |
| 14 | pr439 | | MMAS | 107,929 | 114,244 | 110,826.9 | 0.66 | 1682.36 |
| | | | **DCM-ACO** | **107,400** | **109,158** | **108,408.8** | **0.17** | **475.95** |
| | | | ACS | 35,032 | 37,318 | 35,476.7 | 1.12 | 535.75 |
| 15 | p654 | 34,643 | MMAS | 36,257 | 37,923 | 36,996.8 | 4.66 | 518.65 |
| | | | **DCM-ACO** | **34,795** | **35,496** | **34,927.7** | **0.43** | **161.82** |
| | | | ACS | 279,388 | 293,489 | 285,565.9 | 3.40 | 3496.78 |

**Table 9** (continued)

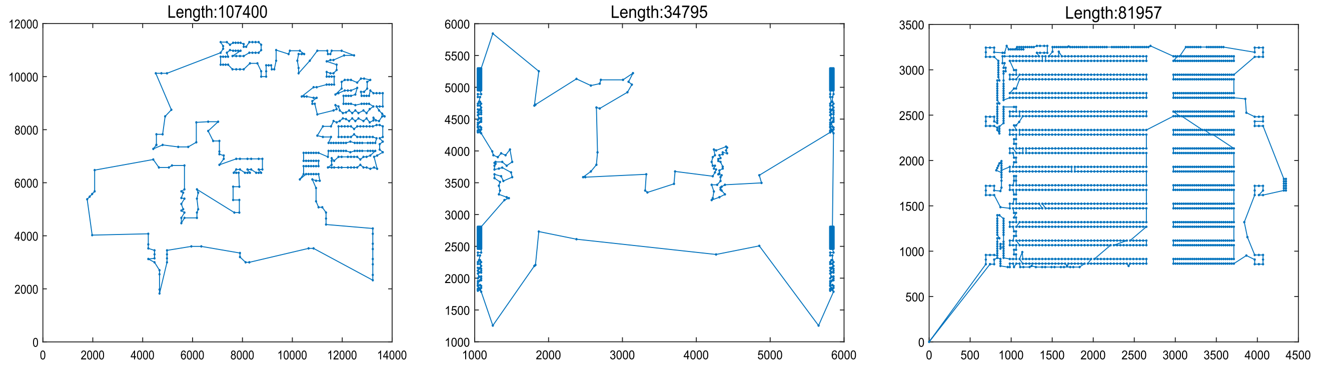
| No | TSP | Optimal | Algorithm | Best | Worst | Average | Error (%) | Std (%) |
|---|---|---|---|---|---|---|---|---|
| 16 | rl1323 | 270,199 | MMAS | 285,858 | 309,420 | 298,623.8 | 5.80 | 5855.98 |
| | | | **DCM-ACO** | **273,707** | **279,855** | **276,716.7** | **1.29** | **1663.25** |
| | | | ACS | 20,760 | 23,121 | 21,503.8 | 3.15 | 566.40 |
| 17 | fl1400 | 20,127 | MMAS | 22,204 | 23,718 | 22,963.6 | 10.32 | 440.49 |
| | | | **DCM-ACO** | **20,368** | **20,904** | **20,629.8** | **1.19** | **143.72** |
| | | | ACS | 83,671 | 87,556 | 85,748.2 | 4.00 | 1147.31 |
| 18 | d2103 | 80,450 | MMAS | 85,062 | 89,832 | 87,043.3 | 5.73 | 1462.16 |
| | | | **DCM-ACO** | **81,957** | **84,382** | **82,853.4** | **1.87** | **546.47** |

Bold values indicate the best results between the comparison algorithms and the proposed algorithm



(a) pr439 (b) p654 (c) fl1400

**Fig. 8** Comparison the convergence of different algorithms



(a) pr439 (b) p654 (c) d2103

**Fig. 9** Optimal tour found by DCM-ACO

In the experiment of the comparison with SOS-ACO, we analyse the experimental results under three factors that include the best solution, average solution and standard deviation, which the results are shown in Table10. In eil51, eil76 and kroA100 instances, both DCM-ACO and SOS-ACO find the optimal solution whereas DCM-ACO is superior to the SOS-ACO in the rest of instances. Moreover, the average solution computed with DCM-ACO outperforms that computed by SOS-ACO in all instances. Besides, DCM-ACO has a better standard deviation than SOS-ACO in the 5 instances based on 8 instances. In the comparison experiment of PSO-ACO-3opt, three indexes that contain the best solution, worst solution and average solution are selected to analyse the experimental results which are given in Table 11. From Table 11, the average solution calculated by PSO-ACO-3opt is superior to DCM-ACO in eil51 and eil76. However, DCM-ACO has a better average solution than PSO-ACO-3opt in 6 other instances, such as kroA100, ch150, kroA200, pr264,

**Table 10** Compare DCM-ACO with SOS-ACO

| TSP | Optimal | DCM-ACO | | | SOS-ACO | | |
|---|---|---|---|---|---|---|---|
| | | Best | Average | Std | Best | Average | Std |
| eil51 | 426 | **426** | **426.5** | **0.55** | **426** | 428.1 | 1.14 |
| eil76 | 538 | **538** | **538.6** | **1.11** | **538** | 541.7 | 2.49 |
| kroA100 | 21,282 | **21,282** | **21,289.2** | **8.57** | **21,282** | 21,290.1 | 12.56 |
| ch150 | 6528 | **6528** | **6546.4** | 11.86 | 6558 | 6571.2 | **10.52** |
| kroA200 | 29,368 | **29,368** | **29,494.6** | **59.89** | 29,413 | 29,520.7 | 69.31 |
| pr264 | 49,135 | **49,135** | **49,163.4** | **25.01** | 49,135 | 49,250.7 | 107.60 |
| lin318 | 42,029 | **42,179** | **42,638.9** | 177.35 | 42,473 | 42,762.7 | **169.69** |
| pr439 | 107,217 | **107,400** | **108,408.8** | 475.95 | 107,978 | 108,873.8 | **350.89** |

Bold values indicate the best results between the comparison algorithms and the proposed algorithm

**Table 11** Compare DCM-ACO with PSO-ACO-3opt

| TSP | Optimal | DCM-ACO | | | PSO-ACO-3opt | | |
|---|---|---|---|---|---|---|---|
| | | Best | Worst | Average | Best | Worst | Average |
| eil51 | 426 | **426** | **427** | 426.5 | **426** | 428 | **426.4** |
| eil76 | 538 | **538** | 541 | 538.6 | **538** | **539** | **538.3** |
| kroA100 | 21,282 | **21,282** | **21,373** | **21,289.2** | 21,301 | 21,554 | 21,445.1 |
| ch150 | 6528 | **6528** | **6563** | **6546.4** | 6538 | 6622 | 6563.9 |
| kroA200 | 29,368 | **29,368** | **29,595** | **29,494.6** | 29,468 | 29,957 | 29,646.0 |
| fl417 | 11,861 | **11,901** | 12,042 | **11,955.5** | 11,947 | **12,003** | 11,980.4 |
| pr439 | 107,217 | **107,400** | **109,158** | **108,408.8** | 108,530 | 109,341 | 108,873.8 |
| p654 | 107,217 | **34,795** | **34,927.7** | **34,927.7** | 35,052 | 35,145 | 35,098.2 |

Bold values indicate the best results between the comparison algorithms and the proposed algorithm

**Table 12** Compare proposed algorithm with other algorithms in small-scale TSP instances

| Algorithm | | eil51 | eil76 | kroA100 | ch150 | kroA200 | pr264 |
|---|---|---|---|---|---|---|---|
| Proposed | Best | 426 | 538 | 21,282 | 6528 | 29,368 | 49,135 |
| Algorithm | Error% | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| LDTACO | Best | 426 | 538 | 21,282 | 6528 | 29,380 | – |
| (2021) | Error% | 0.00 | 0.00 | 0.00 | 0.00 | 0.04 | – |
| DBAL | Best | 426 | 538 | 21,282 | – | 29,368 | 49,135 |
| (2021) | Error% | 0.00 | 0.00 | 0.00 | – | 0.00 | 0.00 |
| PPACO | Best | 426 | 538 | 21,282 | 6528 | 29,368 | 49,135 |
| (2021) | Error% | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 | 0.00 |
| HAACO | Best | 426 | 538 | 21,282 | 6566 | 29,483 | – |
| (2020) | Error% | 0.00 | 0.00 | 0.00 | 0.58 | 0.39 | – |
| DSMO | Best | 428 | 558 | 21,298 | – | 30,481 | – |
| (2020) | Error% | 0.67 | 3.72 | 0.07 | – | 3.79 | – |
| NACO | Best | 426 | – | 21,282 | 6528 | 29,368 | 49,144 |
| (2020) | Error% | 0.00 | – | 0.00 | 0.00 | 0.00 | 0.02 |
| PACO-3opt | Best | 426 | 538 | 21,282 | 6570 | 29,368 | – |
| (2018) | Error% | 0.00 | 0.00 | 0.00 | 0.64 | 0.00 | – |
| DSOS | Best | 426 | 542 | 21,282 | 6542 | 29,477 | 50,454 |
| (2017) | Error% | 0.00 | 0.74 | 0.00 | 0.21 | 0.37 | 2.68 |
| HMMA | Best | 438 | 549 | 21,371 | 6654 | 29,999 | 50,554 |
| (2015) | Error% | 0.00 | 0.00 | 0.09 | 1.89 | 0.34 | 2.89 |

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

| Algorithm | | lin318 | fl417 | pr439 | p654 | rl1323 | fl1400 |
|---|---|---|---|---|---|---|---|
| Proposed | Best | 42,179 | 11,901 | **107,400** | **34,795** | **273,707** | ***20,368*** |
| Algorithm | Error% | 0.35 | 0.34 | **0.17** | **0.43** | **1.29** | ***1.19*** |
| DBAL | Best | **42,072** | **11,861** | 107,603 | – | – | – |
| (2021) | Error% | **0.10** | **0.00** | 0.36 | – | – | – |
| PACO-3opt | Best | – | 11,972 | 108,482 | 35,027 | – | – |
| (2018) | Error% | – | 0.94 | 1.12 | 1.24 | – | – |
| IVNS | Best | 43,924 | 12,180 | 11,175 | – | 295,607 | 21,040 |
| (2018) | Error% | 4.51 | 2.69 | 4.22 | – | 9.45 | 4.53 |
| HMMA | Best | 45,443 | 45,349 | 12,543 | 114,095 | – | 23,099 |
| (2015) | Error% | 8.12 | 7.90 | 5.07 | 6.41 | – | 14.77 |
| Reference [42] | Best | 42,487 | – | – | – | 277,642 | 20,593 |
| (2011) | Error% | 1.09 | – | – | – | 2.75 | 2.32 |

Bold values indicate the best results between the comparison algorithms and the proposed algorithm

**Table 14** Rank sum test between DCM-ACO and ACS

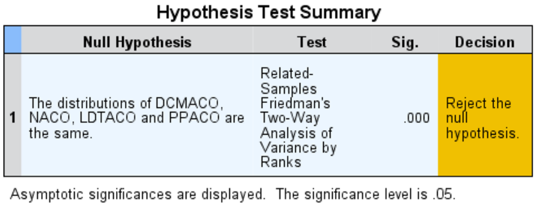
| TSP | *P*-value | sig | TSP | *P*-value | sig |
|---|---|---|---|---|---|
| eli51 | 9.90E−06 | YES | eil76 | 3.78E−08 | YES |
| kroA100 | 3.19E−06 | YES | kroB100 | 6.16E−06 | YES |
| ch130 | 3.61E−07 | YES | ch150 | 5.76E−07 | YES |
| kroB150 | 9.70E−06 | YES | kroA200 | 2.91E−07 | YES |
| kroB200 | 1.92E−07 | YES | pr264 | 7.46E−08 | YES |
| a280 | 6.95E−05 | YES | lin318 | 1.32E−07 | YES |
| fl417 | 9.10E−08 | YES | pr439 | 2.03E−05 | YES |
| p654 | 1.12E−06 | YES | rl1323 | 1.20E−06 | YES |
| fl1400 | 1.43E−07 | YES | d2103 | 9.17E−08 | YES |

**Table 15** Rank sum test between DCM-ACO and MMAS

| TSP | *P*-value | sig | TSP | *P*-value | sig |
|---|---|---|---|---|---|
| eli51 | 6.30E−03 | YES | eil76 | 3.85E−08 | YES |
| kroA100 | 1.02E−03 | YES | kroB100 | 5.85E−07 | YES |
| ch130 | 1.14 E−03 | YES | ch150 | 4.63 E−04 | YES |
| kroB150 | 1.54 E−02 | YES | kroA200 | 2.03 E−02 | YES |
| kroB200 | 6.01E−07 | YES | pr264 | 1.70E−05 | YES |
| a280 | 3.53E−04 | YES | lin318 | 1.23E−07 | YES |
| fl417 | 7.84E−08 | YES | pr439 | 4.53E−06 | YES |
| p654 | 6.79E−08 | YES | rl1323 | 6.80E−08 | YES |
| fl1400 | 6.75E−08 | YES | d2103 | 6.80E−08 | YES |

fl417, pr439 and p654. In addition, DCM-ACO outperforms the comparison algorithm in most instances under the best solution and the worst solution, especially in pr439 and p654. In total, the experimental results show that DCM-ACO has strong competitiveness with comparison algorithms for TSP.

In the second phase of this section, we compare the DCM-ACO with various other optimization algorithms. The comparison optimization algorithms are mainly improved ant colony algorithms that include HAACO [15], PACO-3opt [35], HMMA [36], NACO [28], LDTACO [29], PPACO [37] and other swarm intelligence algorithms that include IVNS [38], Discrete Bat Algorithm DBAL [39], Discrete Spider Monkey Optimization DSMO [40], Discrete Symbiotic Organisms Search algorithm DSOS [41] and reference [42]. Tables 12 and 13 show the specific experiment data in small-scale and large-scale TSP instance respectively. And



Hypothesis Test Summary

| | Null Hypothesis | Test | Sig. | Decision |
|---|---|---|---|---|
| 1 | The distributions of DCMACO, NACO, LDTACO and PPACO are the same. | Related-Samples Friedman's Two-Way Analysis of Variance by Ranks | .000 | Reject the null hypothesis. |

Asymptotic significances are displayed. The significance level is .05.

**Fig. 10** Hypothesis test summary

the Best denotes the best solution obtained by each algorithm, the error is the error rate defined by equal (21). And the "-" in the table denotes that the comparison algorithm does not test the instance.

In small-scale TSP instances, as shown in Table 12, our proposed algorithm finds all the standard optimal solutions from eil51 to pr264, which outperforms the comparison algorithms such as LDTACO, DSMO, PACO-3opt and HAACO. Moreover, in Table 13, which shows the experimental results in large-scale instances, DCM-ACO is also superior to the recent algorithms. For the lin318 and fl417 instances, DBAL find the minimum value over all the six competing algorithms. While in other instances such as pr439, p654, rl1323 and fl1400, DCM-ACO obtains better solutions over all the comparison algorithms. Especially for large-scale TSP instances, the solution computed by DCM-ACO is superior to that found by IVNS and reference [42] in rl1323. Besides, in fl1400, our algorithm also outperforms IVNS, HMMA and reference[42]. In the light of these results, DCM-ACO has strong competitiveness with the state-of-art algorithms for TSP, especially for large-scale instances.

## Statistical test analysis

To verify the effectiveness of the proposed algorithm, we perform the Wilcoxon rank-sum test in this section. In this test, $P$-value denotes the significance level, and if$P$-value < 0.05, the sig in Tables 14, 15 is YES, which means that the original hypothesis is rejected and the performance between our proposed method and the comparison algorithms is significantly different. Otherwise, the sig is NO, which implies that there is no significant difference among these algorithms. In this part, the original hypothesis is given as: the performance between the algorithms is no significant difference. From Tables 14, 15, all the values of $P$-value in two tests are less than 0.05, it can be concluded that the comparison experiments between DCM-ACO and traditional ant colony algorithms have statistical significance.

In addition, to require a fair conclusion further, we perform Friedman's test to verify the significant difference between our proposed method and other improved algorithms including NACO [28], LDTACO [29] and PPACO [37]. First, we set the null hypothesis $H_0$ is: DCM-ACO has no significant difference with NACO, LDTACO, and PPACO. Then 10 experimental data from the kroA200 and lin318 instances are selected to carry out Friedman's test under the SPSS25 software. The results are shown in Fig. 10 and Table 16. As we can see in Fig. 10, the decision is that rejection for the $H_0$ which means the proposed method has a significant difference with comparison algorithms. Specifically, from-Fig.11, the mean rank of DCM-ACO, NACO, LDTACO and PPACO is 1.27, 3.07, 2.03 and 2.93 respectively. Since the response rates differ at different frequencies, pairwise comparisons are necessary. Table 16 shows the results of the pairwise comparison test. From Table 16, the Adjust significance of DCM-ACO with NACO, LDTACO and PPACO is 0.011, 0.002, 0.001, respectively, which all the values are less than 0.05. It means that DCM-ACO is different from NACO, LDTACO and PPACO. In a word, the comparison experiments between DCM-ACO and other algorithms have statistical significance.

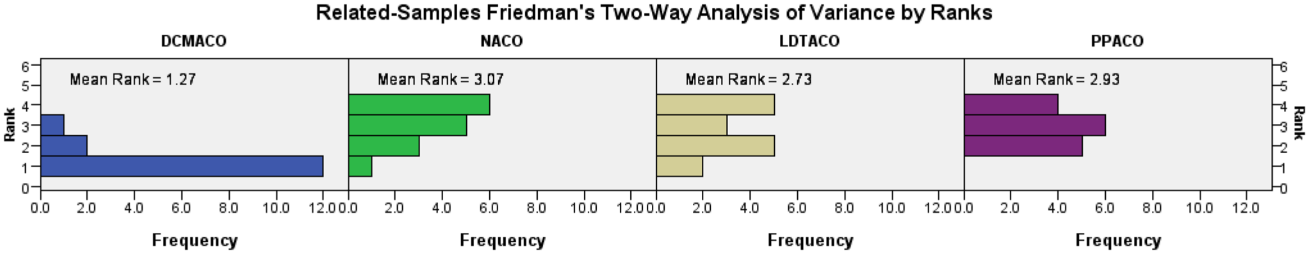


**Fig. 11** Related-samples Friedman's Two-way analysis of variance by ranks

**Table 16** Comparison test

| Sample1–Sample2 | Test statistic | Std. error | Std. test statistic | sig | Adj.Sig |
|---|---|---|---|---|---|
| DCM-ACO-NACO | – 1.47 | 0.47 | – 3.11 | 0.002 | 0.011 |
| DCM-ACO-LDTACO | – 1.67 | 0.47 | – 3.54 | 0.000 | 0.002 |
| DCM-ACO-PPACO | – 1.80 | 0.47 | – 3.82 | 0.000 | 0.001 |

## Conclusion

In this paper, we have proposed a novel ant colony algorithm, so-called multi-ACO with dynamic collaborative mechanism and cooperative game, to solve traveling salesman problems. In the proposed algorithm, two colonies of *Ant Colony System* (ACS) and one colony of *Max–Min Ant System* (MMAS) are formed heterogeneous population. And two ACS populations can better amplify the convergence speed of the algorithm. One MMAS subpopulation is added to enhance the diversity of the population effectively. The advantages of multiple populations complement to ensure the solution quality of the algorithm.

In addition, the cooperative game based on Shapley value is designed to regulate the pheromone accumulation among ACS subpopulations with respect to the contribution of each participant after each iteration. From the experimental results, it portrays that the cooperative game has been proved to be effective and it can adaptively control the pheromone accumulation and improve the diversity of the algorithm.

The dynamic collaborative mechanism is introduced to coordinate among heterogeneous populations. Two methods that pheromone fusion mechanism and public path recommendation strategy are in the mechanism. The former method based on information entropy can help ACS population get rid of the local optimal and the latter accelerates the convergence of the MMAS population effectively. The experiment results in different scale TSPs demonstrate that the improved algorithm can fully complement the advantages of the heterogeneous populations and balance the convergence speed and the diversity of the algorithm effectively, especially for large-scale TSP instances.

In the future, more types of collaborative mechanism can be designed in heterogeneous populations and more kinds of payoff distribution strategies based on game theory can be used to promote co-evolution among populations. In addition, except for the evaluation criteria under information entropy in this paper, other cross-discipline methods such as statistics or machine learning can also be attempted to improve the performance of the population further. Finally, the game mechanism we proposed in this research also has some certain practical value in the application of the ant colony algorithm.

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

## Declarations

**Conflict of interest** The authors declare no conflict of interest.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

## References

1. Dong X, Zhang H, Xu M, Shen F (2021) Hybrid genetic algorithm with variable neighborhood search for multi-scale multiple bottleneck traveling salesmen problem. Futur Gener Comput Syst 114:229–242. https://doi.org/10.1016/j.future.2020.07.008
2. Chen H, Li W, Yang X (2020) A whale optimization algorithm with chaos mechanism based on quasi-opposition for global optimization problems. Expert Syst Appl 158:113612. https://doi.org/10.1016/j.eswa.2020.113612
3. Panwar K, Deep K (2021) Discrete Grey Wolf Optimizer for symmetric travelling salesman problem. Appl Soft Comput 105:107298. https://doi.org/10.1016/j.asoc.2021.107298
4. Wang Y, Han Z (2021) Ant colony optimization for traveling salesman problem based on parameters optimization. Appl Soft Comput 107:107439. https://doi.org/10.1016/j.asoc.2021.107439
5. Dorigo M, Maniezzo V, Colorni A (1996) Ant system: optimization by a colony of cooperating agents. IEEE Trans Syst Man, Cybern Part B 26:29–41. https://doi.org/10.1109/3477.484436
6. Dorigo M, Gambardella LM (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. IEEE Trans Evol Comput 1:53–66. https://doi.org/10.1109/4235.585892
7. Stützle T, Hoos HH (2000) MAX–MIN Ant System. Futur Gener Comput Syst 16:889–914. https://doi.org/10.1016/S0167-739X(00)00043-1
8. Sangeetha V, Krishankumar R, Ravichandran KS, Kar S (2021) Energy-efficient green ant colony optimization for path planning in dynamic 3D environments. Soft Comput 25:4749–4769. https://doi.org/10.1007/s00500-020-05483-6
9. Ye K, Zhang C, Ning J, Liu X (2017) Ant-colony algorithm with a strengthened negative-feedback mechanism for constraint-satisfaction problems. Inf Sci (Ny) 406–407:29–41. https://doi.org/10.1016/j.ins.2017.04.016
10. Ning J, Zhang Q, Zhang C, Zhang B (2018) A best-path-updating information-guided ant colony optimization algorithm. Inf Sci (Ny) 433–434:142–162. https://doi.org/10.1016/j.ins.2017.12.047
11. Guan B, Zhao Y, Li Y (2021) An improved ant colony optimization with an automatic updating mechanism for constraint satisfaction problems. Expert Syst Appl 164:114021. https://doi.org/10.1016/j.eswa.2020.114021
12. Liu G, He D (2013) An improved ant colony algorithm based on dynamic weight of pheromone updating. In: 2013 ninth international conference on natural computation (ICNC), pp 496–500

13. Mahi M, Baykan ÖK, Kodaz H (2015) A new hybrid method based on Particle Swarm Optimization, Ant Colony Optimization and 3-Opt algorithms for Traveling Salesman Problem. Appl Soft Comput J 30:484–490. https://doi.org/10.1016/j.asoc.2015.01.068
14. Olivas F, Valdez F, Castillo O et al (2017) Ant colony optimization with dynamic parameter adaptation based on interval type-2 fuzzy logic systems. Appl Soft Comput J 53:74–87. https://doi.org/10.1016/j.asoc.2016.12.015
15. Tuani AF, Keedwell E, Collett M (2020) Heterogenous adaptive ant colony optimization with 3-opt local search for the travelling salesman problem. Appl Soft Comput 97:106720. https://doi.org/10.1016/j.asoc.2020.106720
16. Wang Y, Wang L, Chen G et al (2020) An improved ant colony optimization algorithm to the periodic vehicle routing problem with time window and service choice. Swarm Evol Comput. https://doi.org/10.1016/j.swevo.2020.100675
17. Martin E, Cervantes A, Saez Y, Isasi P (2020) IACS-HCSP: Improved ant colony optimization for large-scale home care scheduling problems. Expert Syst Appl. https://doi.org/10.1016/j.eswa.2019.112994
18. Miao C, Chen G, Yan C, Wu Y (2021) Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm. Comput Ind Eng 156:107230. https://doi.org/10.1016/j.cie.2021.107230
19. Yang Y, Wu W, Zhang J et al (2021) Determination of critical slip surface and safety factor of slope using the vector sum numerical manifold method and MAX-MIN ant colony optimization algorithm. Eng Anal Bound Elem 127:64–74. https://doi.org/10.1016/j.enganabound.2021.03.012
20. Shokouhifar M (2021) FH-ACO: Fuzzy heuristic-based ant colony optimization for joint virtual network function placement and routing. Appl Soft Comput 107:107401. https://doi.org/10.1016/j.asoc.2021.107401
21. Zhao D, Liu L, Yu F et al (2021) Chaotic random spare ant colony optimization for multi-threshold image segmentation of 2D Kapur entropy. Knowl-Based Syst 216:106510. https://doi.org/10.1016/j.knosys.2020.106510
22. Gambardella LM (1999) MACS-VRPTW: a multiple ant colony system for vehicle routing problems with time windows. New Ideas Optim 1999:5
23. Chu S-C, Roddick JF, Pan J-S (2004) Ant colony system with communication strategies. Inf Sci (NY) 167:63–76. https://doi.org/10.1016/j.ins.2003.10.013
24. Twomey C, Stützle T, Dorigo M et al (2010) An analysis of communication policies for homogeneous multi-colony ACO algorithms. Inf Sci (NY) 180:2390–2404. https://doi.org/10.1016/j.ins.2010.02.017
25. Dong G, Guo WW, Tickle K (2012) Solving the traveling salesman problem using cooperative genetic ant systems. Expert Syst Appl 39:5006–5011. https://doi.org/10.1016/j.eswa.2011.10.012
26. Zhang D, You X, Liu S, Yang K (2019) Multi-Colony Ant Colony Optimization Based on Generalized Jaccard Similarity Recommendation Strategy. IEEE Access 7:157303–157317. https://doi.org/10.1109/ACCESS.2019.2949860
27. Wang Y, Wang L, Peng Z et al (2019) A multi ant system based hybrid heuristic algorithm for vehicle routing problem with service time customization. Swarm Evol Comput 50:100563. https://doi.org/10.1016/j.swevo.2019.100563
28. Yang K, You X, Liu S, Pan H (2020) A novel ant colony optimization based on game for traveling salesman problem. Appl Intell 50:4529–4542. https://doi.org/10.1007/s10489-020-01799-w
29. Li S, You X, Liu S (2021) Multiple ant colony optimization using both novel LSTM network and adaptive Tanimoto communication strategy. Appl Intell. https://doi.org/10.1007/s10489-020-02099-z
30. Shannon CE (1948) A mathematical theory of communication. Bell Syst Tech J 27:379–423. https://doi.org/10.1002/j.1538-7305.1948.tb01338.x
31. Zhao J, Liang JM, Dong ZN et al (2020) Accelerating information entropy-based feature selection using rough set theory with classified nested equivalence classes. Pattern Recognit 107:107517. https://doi.org/10.1016/j.patcog.2020.107517
32. Sh. Sabirov D, (2020) Information entropy of mixing molecules and its application to molecular ensembles and chemical reactions. Comput Theor Chem 1187:112933. https://doi.org/10.1016/j.comptc.2020.112933
33. Zhu H, Wang Y, Du C et al (2020) A novel odor source localization system based on particle filtering and information entropy. Rob Auton Syst 132:103619. https://doi.org/10.1016/j.robot.2020.103619
34. Shapley LS (1952) A Value for n-Person Games
35. Gülcü Ş, Mahi M, Baykan ÖK, Kodaz H (2018) A parallel cooperative hybrid method based on ant colony optimization and 3-Opt algorithm for solving traveling salesman problem. Soft Comput 22:1669–1685. https://doi.org/10.1007/s00500-016-2432-3
36. Yong W (2015) Hybrid Max-Min ant system with four vertices and three lines inequality for traveling salesman problem. Soft Comput 19:585–596. https://doi.org/10.1007/s00500-014-1279-8
37. Pan H, You X, Liu S, Zhang D (2021) Pearson correlation coefficient-based pheromone refactoring mechanism for multi-colony ant colony optimization. Appl Intell 51:752–774. https://doi.org/10.1007/s10489-020-01841-x
38. Hore S, Chatterjee A, Dewanji A (2018) Improving variable neighborhood search to solve the traveling salesman problem. Appl Soft Comput J 68:83–91. https://doi.org/10.1016/j.asoc.2018.03.048
39. Saji Y, Barkatou M (2021) A discrete bat algorithm based on Lévy flights for Euclidean traveling salesman problem. Expert Syst Appl 172:114639. https://doi.org/10.1016/j.eswa.2021.114639
40. Akhand MAH, Ayon SI, Shahriyar SA et al (2020) Discrete Spider monkey optimization for travelling salesman problem. Appl Soft Comput J 86:105887. https://doi.org/10.1016/j.asoc.2019.105887
41. Ezugwu AES, Adewumi AO (2017) Discrete symbiotic organisms search algorithm for travelling salesman problem. Expert Syst Appl 87:70–78. https://doi.org/10.1016/j.eswa.2017.06.007
42. Chen SM, Chien CY (2011) Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques. Expert Syst Appl 38:14439–14450. https://doi.org/10.1016/j.eswa.2011.04.163