Dynamic ant colony algorithm based on cw saving Value

YaoJiang-Gu¹, Jian Gao

Shanghai University, Baoshan District, Shanghai, China
Shanghai University, Baoshan District, Shanghai, China

Abstract. As a swarm intelligence optimization algorithm, ant colony algorithm (ACO) has a good application in combinatorial optimization problems, in which traveling salesman problem (TSP) is an important application of ACO algorithm. It shows the powerful ability of ant colony algorithm to find short paths through graphics. However, there are obvious defects in the ant colony algorithm. When the scale of the ant colony is large, the convergence time of the algorithm becomes longer and the local optimal state is easy to fall into. In this paper, a dynamic pheromone ant colony optimization algorithm based on CW saving algorithm is proposed. Initially, a general path range is found by CW saving value algorithm, and the pheromone matrix can be reasonably configured, so that the ant colony algorithm can quickly get a better solution in the initial optimization. At the same time, the optimization scheme can be adjusted in real time according to the situation of path optimization. Large ant colony searches for other paths. Combined with 3-opt local search algorithm, the ant colony can find the optimal path more quickly. The experimental results show that the improved ant colony algorithm has better convergence speed and solution quality than other ant colony algorithms.

1 INTRODUCTION

With the rapid development of computer technology, the application field of intelligent computing methods is also more and more extensive. Heuristic optimization algorithm is an algorithm that imitates biological behaviour to solve problems inspired by the laws of nature. Inspired by nature and imitating its structure for invention and creation, this is bionics^[1].

Ant colony algorithm is an intelligent combinatorial optimization algorithm proposed by professor Dorigo in his doctoral thesis in 1992^[2]. The algorithm simulates the foraging behaviour of ants. Ants leave behind a chemical substance called pheromone during foraging. Ants follow the ant colony to find the best foraging path by sensing pheromone intensity^[3]. Research shows that ant colony algorithm has the advantages of high robustness, distributed computing, positive feedback and easy integration with other algorithms^[4]. However, when dealing with big data problems, the calculation time is long, and it is easy to fall into local optimal solution, even causing stagnation. In response to these problems, many scholars have put forward corresponding improvement methods^[5].

The core technology of ant colony algorithm lies in the choice of pheromone update mode and state transition probability. Among them, the construction of pheromone matrix is especially crucial. Pheromone guides artificial ants to choose their paths. The state transition of the ant colony is based on the pheromone matrix of the ant colony, so the reconstruction of the pheromone matrix can provide an improved ant colony algorithm, such as ACS algorithm

^[6], MMAS algorithm^[7], adaptive algorithm^[8], dynamic pheromone update algorithm^[9], hybrid behaviour ant colony algorithm^[10], etc. All the above methods improve the distribution of pheromone matrix by improving the way of pheromone update or adding restrictions to pheromones to improve the ability of path optimization. For example, Gao R^[11] adapts the fast convergence rate by candidate set strategy and adaptively adjusts pheromone strategy so that pheromone distribution is relatively uniform. Xu J M^[12] proposed a polymorphic ant colony algorithm with a variety of ant colonies and information hormones, which can improve search and convergence speed through different pheromone control mechanisms. In recent years, more and more attention has been paid to the method of combining ant colony algorithm with other algorithms. For example, Jie L I^[13] combines Ant Colony algorithm and genetic algorithm to solve the problem of Wireless Network location. In document^[14], Chen W proposes to combine particle swarm optimization with ant colony algorithm to achieve win-win results in search time and performance.

2 RELATED WORK

TSP problem belongs to NP hard problem. There are mainly two ways to solve this kind of problem at present. One is accurate algorithms such as dynamic programming method and branch and bound method. For the small-scale TSP problem, an accurate algorithm can be used to find the solution. However, when the ant colony is large in scale, it cannot be solved by accurate methods. In recent years, people have published many articles to solve the

^a Corresponding author: ¹ 1483260200@qq.com

TSP problem (Fanzhen Liu, Yong Quanzhou, Mostafa Mahaia), and many improved algorithms have made great progress. The following is a brief list of some.

Fanzhen Liu^[15] et al set the number of initial ant colonies and verified the relationship between the initial average optimal solution and the calculation cost through experiments. In the TSP problem, a strategy to initialize the ant population of ACO algorithm is proposed to obtain high quality optimal solution in a short period of time.

Jing qing Jiang^[16] and others adopted a hierarchical method to solve the large-scale TSP problem. The large scale traveling salesman problem is decomposed layer by layer into generalized traveling salesman problem (GTSP) by clustering analysis of cities, and each clustering is solved by ant colony algorithm. Then ant colony algorithm was used to solve each cluster. The whole solution will be obtained after each clustering result is obtained. It is verified that this algorithm can obviously improve the efficiency of solving large-scale ant colony algorithm.

Saban g ü LC ü^[17] et al. proposed a parallel collaborative hybrid algorithm PACO-3 opt. Based on the master-slave model, the algorithm divides the entire ant colony into several sub-ant colonies. Each ant colony shares information through GIS strategy. Once one sub-ant colony is stuck, other communities will help it to jump out of the stuck state. Each independent sub-ant colony is operated according to the ant colony optimization algorithm and the 3-opt algorithm, and the optimal solution is shared among the periods. After verification, the algorithm can get better results. It has higher parallel operation efficiency on distributed management system.

In this paper, a dynamic pheromone ant colony optimization algorithm based on CW saving value algorithm is proposed. through real-time monitoring of ant colony search paths, the pheromone matrix can be reasonably configured, the optimization scheme can be adjusted, premature phenomenon can be prevented in advance, and the possibility of ant colony searching other paths can be increased. Combining with the 3-opt local search algorithm, the convergence speed of the algorithm is improved while the global search capability is enhanced, and the optimal solution is searched to the greatest extent. At the end of this paper, the algorithm is verified by experiments, and compared with other algorithms, the results are satisfactory.

3 MATERIALS AND METHODS

3.1 ant colony system

This paper takes TSP problem in n cities on the plane as an example to verify the improvement of ant colony system algorithm (ACS).

3.1.1 state transition rules

In ant colony system, m ants construct TSP paths in parallel. At the initial moment, the ants were placed in randomly selected cities, and the path information elements were equal among the cities. Ant k in city i uses the

pseudo-random proportional rule of formula (1) to determine the city j to be visited next.

$$j = \begin{cases} \arg \max_{s \in N_i^k} \{ \tau_{is}^\alpha \cdot \eta_{is}^\beta \} & , q \leq q_0 \\ J & , q > q_0 \end{cases} \quad (1)$$

Among them, N_i^k refers to the collection of cities that ants k located in city i can reach directly, and J is the next city label selected by formula (2) in combination with roulette algorithm. $q \in (0,1)$ is a random number with uniform distribution, and $q_0 \in [0,1]$ is a fixed parameter.

$$P_{ij}^k = \begin{cases} \frac{\tau_{ij}^\alpha \cdot \eta_{ij}^\beta}{\sum_{s \in N_i^k} \tau_{is}^\alpha \cdot \eta_{is}^\beta} & , j \in N_i^k \\ 0 & , j \notin N_i^k \end{cases} \quad (2)$$

In formula (2), P_{ij}^k is the transition probability of ant k in city $i \rightarrow j$, τ_{ij} is the pheromone intensity of city $i \rightarrow j$, η_{ij} is the visibility of city $i \rightarrow j$, and is generally taken as $1/d_{ij}$, representing the predetermined heuristic information. α and β are the relative importance of pheromone and heuristic information respectively reflected in ant path selection.

3.1.2 local pheromone update

When each ant chooses a path, it will locally update the pheromone on that path.

$$\tau_{ij} = (1 - \zeta) \cdot \tau_{ij} + \zeta \cdot \tau_0 \quad , \quad \tau_0 \in (0,1) \quad (3)$$

In formula (3), ζ is a fixed parameter, τ_0 is the value of the initial pheromone, and the general value is $1/n * L_{nn}$. Among them, n represents the number of cities in the TSP problem, and L_{nn} represents the path length obtained by the nearest neighbor heuristic method. By updating local pheromone, the pheromone of this path is weakened, the possibility of ant colony searching for other paths is guaranteed, and local optimal solution is avoided.

3.1.3 global pheromone update

After all ants have traversed the city, they begin to update the global pheromone. The updating of global pheromones will only play a role in the path of the best ants. The global pheromone update formula is as follows:

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \rho \cdot \Delta \tau_{ij} \quad (4)$$

Where ρ is the pheromone volatilization factor, $\Delta \tau_{ij}$ is the pheromone increment, and the general value is $1/L_{gst}$. $1/L_{gst}$ is the length of the globally optimal path found by the algorithm so far.

3.2 CW algorithm

The CW mileage saving method (Clarke-Wright) is a set of heuristic algorithms proposed by Clarke and Wright to solve NP hard problem, i.e. inexact algorithms. The basic idea of the algorithm is:

1. Take one of the n cities as the distribution center. Connect the remaining points to the distribution center. Constitute line $1 - j - 1 (j = 2, 3, \dots, n)$, $n - 1$ lines with only one access point.
2. Calculate the savings value $S(i, j) = c_{1i} + c_{1j} - c_{ij}$ for all connectable point pairs (i, j) within the constraint.
3. The calculations $S(i, j) > 0$ are arranged in order from big to small.
4. Check one by one according to the above order of $S(i, j)$. If there are two such lines, one containing arcs or edges $(i, 0)$ and the other containing arcs or edges $(0, j)$, and the solution is feasible after merging, then introduce arcs or edges, merge the two lines, and delete $(i, 0)$ and $(0, j)$. Repeat this step until there are no combinable lines.
5. Return to step (4) until all insertable arcs (i, j) have been examined.

3.3 3-opt algorithm

In the TSP problem, there are also some locally improved algorithms, such as the 3-opt algorithm (3 element exchange), which is a type of K-opt algorithm. The idea of the algorithm is that after a feasible solution is obtained, the three sides on the path are randomly exchanged, and the new solution obtained is compared with the original solution. If the result of the new solution is good, it will be retained, otherwise it will be discarded, and then the loop iteration will continue until the replacement is completed. Delete 3 edges and reconnect them. there are 8 possibilities in total, as shown in figure 1.

Figure.1 3-opt exchange diagram

4 PROPOSED METHOD(DACA-3-OPT)

Although heuristic algorithm has achieved good results in solving TSP problem, there are still many shortcomings in solving efficiency and solving quality. To overcome these shortcomings, this paper proposes a dynamic ant colony optimization algorithm based on CW saving algorithm. The algorithm mainly improves the following three aspects:

- a) introducing CW saving algorithm to reasonably plan the initial pheromone matrix distribution in the early stage of ant colony algorithm;
- b) using the improved ant colony algorithm to find the excellent path solution in the path optimization stage;
- c) using the 3-opt algorithm to improve the quality of the solution.

In ant colony algorithm, the initial allocation of pheromone matrix is average. Whether ACS or MMAS,

the pheromone matrix is configured to have the same fixed value at the beginning of the algorithm. This makes the ant colony algorithm unable to find a better solution in the initial path optimization process, and it relies entirely on the update of late pheromone to improve the path optimization situation. Therefore, CW saving value algorithm is introduced in the early stage of the algorithm. The allocation of the pheromone matrix is improved by the path solution found by the CW saving value algorithm.

In the CW saving value algorithm, city points $i = 1, 2, 3, \dots, n$ are chosen as the origin to traverse, and i feasible solutions are obtained. The optimal solution of the n feasible solutions is selected as the basis for distributing the pheromone matrix.

In ACS, the initial value of the pheromone matrix is

$$\tau_0 = \frac{1}{n \cdot L_{nn}} \quad (5)$$

In the improved algorithm, the increment of pheromone on each CW line is

$$\Delta\tau = \frac{1}{\lambda \cdot L_{gst} \cdot n} \quad (6)$$

$\lambda \in (0, 1)$ is a fixed parameter. The distribution value of the pheromone matrix is

$$\tau_0 = \frac{1}{n \cdot L_{nn}} + \Delta\tau \quad (7)$$

The ant colony optimization algorithm is improved as follows:

4.1 Crowding degree

As the number of iterations increases, the pheromone matrix begins to show two levels of differentiation. There are more and more pheromones on the path of the optimal solution, while the pheromones on other paths become less and less with the number of iterations. Ants are insensitive to the changes of pheromones. After a certain number of iterations, due to the difference of polarization, the ant colony can no longer jump out of the current path, and can only continue to iterate under the current solution, resulting in the iterative process into a crowded stage.

Definition 1: Suppose the number of ants passing through the current optimal solution path S_k of k cycle is m_k :

$$d_i = \sum_{j=1, S_j=S_i}^k m_j, \quad i = 1, 2, \dots, k \quad (8)$$

d_i is the crowding of the current optimal path for each cycle. When d_i is greater than the threshold D , the optimal path is determined to be congested.

To maintain the ant colony algorithm's ability to search for new solutions, DACACW algorithm monitors the ant colony state in real time. Once the ant colony algorithm falls into a crowded state, the weight coefficient β in equation (1) is adjusted as follows:

$$\beta = \frac{L_{nn}}{L_{gst}} \quad (9)$$

As β changes, so does the probability that the colony will pick the next city.

4.2 Pheromone matrix

To avoid serious two-level differentiation of the pheromone matrix, inspired by the Max-Min Ant System (MMAS), the value of the pheromone matrix is limited to a certain interval $[\tau_{min}, \tau_{max}]$

$$\tau \in \left[\frac{1}{(1-\rho) \cdot L_{gst}}, \frac{k_0 \cdot (1 - \sqrt[n]{L_{gst}})}{(n-2) \sqrt[n]{L_{gst}}} \cdot \frac{1}{(1-\rho) \cdot L_{gst}} \right] \quad (10)$$

In formula (10), k_0 is a fixed parameter. When the congestion degree is detected, the pheromone matrix is reprogrammed once it gets into congestion, and the mean values of the maximum and minimum pheromone τ_{avg} are taken, that is

$$\tau_{avg} = \frac{1}{2 \cdot (1-\rho) \cdot L_{gst}} \cdot \left(1 + \frac{k_0 \cdot (1 - \sqrt[n]{L_{gst}})}{(N-2) \sqrt[n]{L_{gst}}} \right) \quad (11)$$

The part above the average pheromone changes its size.

$$\tau = \begin{cases} \tau & \tau < \tau_{avg} \\ \theta \cdot \tau & \tau > \tau_{avg} \end{cases}, \quad \theta \in (0,1) \quad (12)$$

4.3 Search field

Since the time complexity of the 3-opt algorithm is $O(n^3)$, it will take lots of time to traverse the entire path at a time. Therefore, when carrying out the local search algorithm, the scope of the search domain is prioritized, and 3-opt algorithm operations are carried out within this scope each time. In this paper, the search domain is defined as an area where the distance between two exchanged edges does not exceed a certain length. The 3-opt algorithm is operated

within the allowed length range, thus reducing the operation steps and increasing the operation speed.

Figure.2 Neighborhood search process
The flow chart of this algorithm is as follows:

Figure.3 Algorithm overall flow

5 EXPERIMENTAL RESULTS

In this section, the performance of DACACW algorithm is verified. The parameter settings used in the experiment are shown in table 1 below. Ants choose 67 % of the number of cities. The test problem is extracted from TSPLIB(a standard test case)and several algorithms in the previous paper are compared respectively. The test cases are eil51、eil76、St70、eil101、ch130、KroA200 respectively.

Table 1 parameter settings for this algorithm

α	β	ζ	ρ	q_0	λ	θ	μ	D	k_0
2	5	0.02	0.1	0.01	5	0.4	4	30	10

Table 2 experimental data of DACACW algorithm

Problem	BKS	Best	Worst	Average	SD	Error(%)	Time(s)
Eil51	426	428	431	429.29	0.736710069	0.77	6.737
Berlin52	7542	7544	7550	7545.75	2.268201235	0.05	7.116
Rat99	1211	1217	1221	1218.6	2.01049876	0.57	47.655
Eil76	538	544	547	544.85	1.136708082	1.27	21.689
St70	675	677	681	677.45	0.944513241	0.29	16.369
KroA100	21282	21285	21285	21285	0	0.01	49.054
Lin105	14379	14382	14400	14384.7	5.667079448	0.04	55.983
Kroa200	29368	29399	29420	29403.9	8.527972549	0.12	387.053
Ch150	6528	6530	6608	6545.15	26.13180156	0.26	162.864
Eil101	629	640	649	642.35	3.498495917	2.12	52.624
Att48	33523	33523	33523	33523	0	0	5.561
Ch130	6110	6110	6145	6115.75	12.80162407	0.09	106.998
KroA150	26524	26524	26673	26573.4	64.29995498	0.18	164.001
KroB100	22141	22139	22234	22150.6	28.83053936	0.04	49.176
KroB150	26130	26127	26214	26143.05	31.91679149	0.05	165.500
Pr107	44303	44301	44031	44301	0	-0.0045	60.688
TSP225	3916	3858	3889	3860.15	6.899847443	-1.42	605.767

Looking at Table 3, for the Rat99, St70, KroA200 and ch150 problems, the proposed method yields the closest result to the optimal solution with minimal standard deviation. The average results of Rat99, St70, KroA200 and ch150 were 1218.6, 677.45, 29403.9 and 6545.15, respectively. As can be seen from table 3, these results are

superior to the research results in the literature. It is also observed from table 2 that for the problems of Eil51, Berlin 52, ch150, Att48, ch130, KroA150, KroB100, KroB150, pr107 and TSP225, the results are close to the optimal solution, and in some cases the optimal solution is even better than the official data. Comparing the results

in the literature, we can see that the method proposed in this paper has produced ideal results.

Table 3 comparison of experimental data with other algorithms

Method	Problem BKS	Eil51 426	Berlin52 7542	Rao99 1244	Eil76 675	St70 675	KroA100 21282	Lin105 14379	KroA200 29368	Ch150 6528	Eil101 629
RABNET – TSP (2006)	avg	438.70	8073.97	—	556.10	—	21,868.47	14,702.17	30,257.53	6753.20	654.83
	SD	3.52	270.14	—	8.03	—	245.76	328.37	342.98	83.01	6.57
	Error(%)	2.98	7.05	—	3.36	—	2.76	2.25	3.03	3.45	4.11
Modified RABNET – TSP	avg	437.47	7932.50	—	556.33	—	21,522.73	14,400.7	30,190.27	6738.37	648.63
	SD	4.20	277.25	—	5.30	—	93.34	44.03	273.38	76.14	3.85
	Error(%)	2.69	5.18	—	3.41	—	1.13	0.15	2.80	3.22	3.12
SA ACO PSO (2011)	avg	427.27	7542.00	—	540.20	—	21,370.30	14,406.4	29,738.73	6563.70	635.23
	SD	0.45	0.00	—	2.94	—	123.36	37.28	356.07	22.45	3.59
	Error(%)	0.30	0.00	—	0.41	—	0.41	0.19	1.26	0.55	0.99
IVRS + 2opt	avg	431.10	7547.23	—	—	—	21,498.61	—	—	—	648.67
	SD	—	—	—	—	—	—	—	—	—	—
	Error(%)	1.20	0.07	—	—	—	1.02	—	—	—	3.13
ACO + 2opt	avg	439.25	7556.58	—	—	—	23,441.80	—	—	—	672.37
	SD	—	—	—	—	—	—	—	—	—	—
	Error(%)	3.11	0.19	—	—	—	10.15	—	—	—	6.90
HACO (2012)	avg	431.20	7560.54	1241.33	—	—	—	—	—	—	—
	SD	2.00	67.48	9.60	—	—	—	—	—	—	—
	Error(%)	1.22	0.23	1.42	—	—	—	—	—	—	—
CGAS	avg	—	7634.00	—	542.00	—	21,437.00	—	29,946.00	—	—
	SD	—	—	—	—	—	—	—	—	—	—
	Error(%)	—	1.22	—	0.74	—	0.73	—	1.97	—	—
WFA with 2-Opt	avg	426.65	7542.00	—	541.22	—	21,282.00	14,379.0	29,654.03	6572.13	639.87
	SD	0.66	0.00	—	0.66	—	0.00	0.00	151.42	13.84	2.88
	Error(%)	0.15	0.00	—	0.60	—	0.00	0.00	0.97	0.68	1.73
WFA with 3-Opt	avg	426.60	7542.00	—	539.44	—	21,282.80	14,459.4	29,646.50	6700.10	633.50
	SD	0.52	0.00	—	1.51	—	0.00	1.38	110.91	60.82	3.47
	Error(%)	0.14	0.00	—	0.27	—	0.00	0.56	0.95	2.64	0.72
ACO with ABC	avg	443.39	7544.37	—	557.98	700.58	22,435.31	—	—	6677.12	683.39
	SD	5.25	0.00	—	4.10	7.51	231.34	—	—	19.30	6.56
	Error(%)	4.08	0.03	—	3.71	3.79	5.42	—	—	2.28	8.65
PSO-ACO-3Opt	avg	426.45	7543.20	1227.40	538.30	678.20	21,445.10	14,379.2	29,646.05	6563.95	632.70
	SD	0.61	2.37	1.98	0.47	1.47	78.24	0.48	114.71	27.58	2.12
	Error(%)	0.11	0.02	0.28	0.06	0.47	0.77	0.00	0.95	0.55	0.59
Proposed Method	avg	429.29	7545.75	1218.6	544.85	677.45	21285	14384.7	29403.9	6545.15	643.35
	SD	0.73	2.26	2.01	1.13	0.94	0	5.67	8.52	12.8	3.49
	Error(%)	0.77	0.05	0.57	1.27	0.29	0.01	0.04	0.12	0.09	2.12

Figure. 4 The path formed by each TSPLIB data

6 CONCLUSIONS

As a famous NP hard problem, there is no exact method to find the exact solution of TSP problem. This paper presents an improved ant colony algorithm for traveling salesman problem (TSP). DACACW algorithm gets the initial solution by CW economizing algorithm and improved ACS algorithm, and 3-opt algorithm optimizes these solutions locally. DACACW algorithm has been in the calculation of the results and calculation efficiency has been a satisfactory result. Some of these data also get close to the optimal solution. However, there are still some problems, such as how to jump out of the local optimal solution more quickly, how to strengthen the relationship between ant colonies and how to enable ant colony to achieve the information sharing of each optimization path. How to combine the parallel computing characteristics of the computer to achieve multi-ant colony optimization. In this paper, based on CW economized value dynamic ant colony algorithm, numerical experiments are carried out on eil51, and the optimal path total distance is 428, while the optimal path total distance given in reference [17] [18] is obviously. The experimental results given in this paper are superior to those in the literature.

References

1. Xu Zongben. bionics in computational intelligence [M]. science press, 2003.
2. Stützle T. Ant Colony Optimization[J]. Computational Intelligence Magazine IEEE, 2007, 1(4):28-39.
3. Colorni A Dorigo M, Maniezzo V. Distributed optimization by ant colonies. In: Varela F, Bourguine P, eds. Proc. of the ECAL'91 European Conf of Artificial Life. Paris: Elsevier. 1991. 134-144.
4. Dorigo M, Maniezzo V, Colorni A. Ant system: optimization by a colony of cooperating agents[J]. IEEE Transactions on Systems Man & Cybernetics Part B Cybernetics A Publication of the IEEE Systems Man & Cybernetics Society, 1996, 26(1):29-41.
5. Zhao Yuxin. new meta-heuristic optimization method [M]. science press, 2013.
6. Gómezcabrero D, Ranasinghe D N. Fine-tuning the Ant Colony System algorithm through Particle Swarm Optimization[J]. 2018, 25(25):73-7.
7. Lissovoi A, Witt C. MMAS Versus Population-Based EA on a Family of Dynamic Fitness Functions[J]. Algorithmica, 2016, 75(3):554-576.
8. Yuan Wang Huang, you Xiao Ming, Liu sheng, et al. adaptive simulated annealing ant colony algorithm for tsp [J]. computer applications and software, 2018 (2): 261 - 266.
9. Liu G, He D. An improved Ant Colony Algorithm based on dynamic weight of pheromone updating[C]// Ninth International Conference on Natural Computation. IEEE, 2014:496-500.
10. Wei X M. Hybrid Behavior Ant Colony Algorithm[J]. Advanced Materials Research, 2012, 433-440:4496-4499.
11. Gao R, Wu J. Dynamic Load Balancing Strategy for Cloud Computing with Ant Colony Optimization[J]. Future Internet, 2015, 7(4):465-483.
12. Xu J M, Cao X B, Wang X F. Polymorphic Ant Colony Algorithm[J]. Journal of University of Science & Technology of China, 2005(1):59-65.
13. Jie L I, Zhen-Bo L I, Chen J P. A Hybrid Algorithm Based on Genetic Algorithm and Ant Colony Algorithm for Wireless Network Location[J]. Computer Engineering & Software, 2017.
14. Chen W, Jiang Y. Improving ant colony algorithm and particle swarm algorithm to solve TSP problem[J]. Information Technology, 2016.
15. Liu F, Zhong J, Liu C, et al. A novel strategy of initializing the population size for ant colony optimization algorithms in TSP[C]// International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery. 2017:249-253.
16. Jiang J, Gao J, Li G, et al. Hierarchical Solving Method for Large Scale TSP Problems[M]// Advances in Neural Networks – ISNN 2014. Springer International Publishing, 2014:252-261.
17. Şaban Gülcü, Mahi M, Ömer Kaan Baykan, et al. A parallel cooperative hybrid method based on ant colony optimization and 3-Opt algorithm for solving traveling salesman problem[J]. Soft Computing, 2016(2):1-17.
18. Marvin frost, Zhang Hongwei. tsp based on improved ACS - 3 - opt ant colony algorithm [J]. computer engineering, 2008,34 (19): 200 – 202.