Journal of Computers Vol. 32 No. 4, 2021, pp. 166-178
doi:10.53106/199115992021083204013

Research on Improved Ant Colony Optimization Based on Adaptive Chemical Reaction Optimization

Teng Fei[1], Xin-Xin Wu[2], Xu-Hua Pan[1*]

[1] Institute of Information Engineering, Tianjin University of Commerce, 409 Guangrong Road, Beichen District, Tianjin, China
fei_8825@163.com

[2] College of science, Tianjin University of Commerce, 409 Guangrong Road, Beichen District, Tianjin, China
wuxinxin_tjcu@163.com

Received 20 April 2021; Revised 12 May 2021; Accepted 16 May 2021

Abstract. The positive feedback mechanism of pheromone in the basic ant colony optimization greatly accelerates the optimization process of the algorithm, but also has disadvantages such as prone to stagnation and falling into local optimality. In order to overcome these shortcomings of ant colony optimization, an improved ant colony optimization based on adaptive chemical reaction optimization is proposed. In the optimization iteration process of the ant colony optimization, the decomposition reaction operation and the synthesis reaction operation of the adaptive chemical reaction optimization are introduced to enhance the algorithm's ability to jump out of the local optimum and find the global optimum. Then the paper uses the improved algorithm to simulate and solve the six classic data sets in the Traveling Salesman Problem (TSP). For the same data set, the adaptive chemical reaction ant colony optimization can find a better path than the basic ant colony optimization, its solution success rate is higher, and the numerical fluctuation range of the result obtained is also smaller. These verify that the adaptive chemical reaction ant colony optimization is superior to the basic ant colony optimization in terms of algorithm optimization ability, algorithm stability and algorithm reliability.

Keywords: adaptive, ant colony optimization, chemical reaction optimization, traveling salesman problem

1 Introduction

Ant Colony Optimization is a bionic intelligent optimization algorithm proposed by Italian scholar M. Dorigo et al. in 1991. [1] It is mainly inspired by the foraging behavior of real ants in nature. And it uses ants to work together to find food as a model, simulates the behavioral characteristics and movement rules of ants, and uses pheromone as the traction guide during the movement process to control the ants to find the optimal path.

With the development of science and the deepening of the research of ant colony optimization, the ant colony optimization has been widely used in various fields, such as vehicle routing problem [2], flow shop scheduling problem [3], robot path planning problem [4], image processing problems [5], digital signal processing problems [6], etc. At the same time, various algorithm improvements and algorithm fusion researches made to address the shortcomings of ant colony optimization are also constantly developing. Scholars have proposed many improvements and integration schemes. And these improvements have achieved good results and solved practical problems in different environments.

Since the chemical reaction optimization (CRO) is proposed, it has attracted much attention because of its strong robustness, flexibility in algorithm design and ease of software implementation. [7] The four elementary reactions in the algorithm can well balance the local search ability and global search ability of chemical reaction optimization, and effectively avoid the algorithm from falling into the local optimum.

* Corresponding Author

These make the chemical reaction optimization have conditions and advantages to integrate with other intelligent optimization algorithms.

The motivation of this paper is precisely because the basic ant colony optimization has achieved satisfactory results after being improved or fused with other algorithms. At the same time, the chemical reaction optimization also has conditions and advantages for fusion with other intelligent optimization algorithms. Therefore, the paper considers the fusion of ant colony optimization and chemical reaction optimization, combining the advantages of both to make up for the shortcomings of a single algorithm. And the main research in this paper is the integration of the local optimization capability of the basic ant colony optimization and the global optimization capability of the adaptive chemical reaction optimization. Based on the basic ant colony optimization, the decomposition reaction operation and the synthesis reaction operation of the adaptive chemical reaction optimization are introduced, and a new improved ant colony optimization based on the adaptive chemical reaction optimization—adaptive chemical reaction ant colony optimization (ACRACO) is proposed. Then the improved algorithm ACRACO is applied to solve the traveling salesman problem, and strive to find a smaller travel distance. The proposal of adaptive chemical reaction ant colony optimization not only improves the performance of basic ant colony optimization for solving path optimization problems, but also provides reference and inspiration for the improvement and integration of other algorithms.

The rest of this paper is organized as follows: section 2 is a review of relevant literature; section 3 introduces the principle of the basic ant colony optimization; section 4 introduces the principle and optimization mechanism of the improved adaptive chemical reaction ant colony optimization; section 5 applies the improved algorithm to solving the traveling salesman problem, testing and analyzing the performance of the algorithm; section 6 compares and analyzes the improved algorithm with the newly proposed Donkey and Smuggler Optimization Algorithm (DSO) [8] to verify the effectiveness of the improved algorithm; section 7 summarizes the improved algorithm and proposes future work directions.

2 Literature Review

This section mainly introduces related improvements of ant colony optimization and chemical reaction optimization.

2.1 Improved Ant Colony Optimization

The improvement methods for the ant colony optimization can be roughly summarized into three categories: the improvement of the pheromone adjustment strategy, the improvement of the algorithm search strategy and the fusion with other algorithms. Ebadinezhadet al. proposed an adaptive ant colony optimization with dynamic evaporation strategy to improve the algorithm's uncertain convergence time and random decision-making. [9] Stvtzle et al. proposed the concept of a maximum-minimum ant system, which only updates the pheromone on the optimal path during each iteration to speed up the convergence speed of the algorithm. [10] Yang et al. used the chemical reaction optimization to generate a better solution, and used a pheromone conversion strategy to convert the better solution into the initial pheromone of the ant colony optimization to improve the efficiency of the algorithm. [11] Luan et al. proposed a new hybrid algorithm, which took advantage of the high initial acceleration convergence of genetic algorithm and combined with the parallelism and positive feedback mechanism of ant colony optimization. [12] The hybrid algorithm used the solution generated by the genetic algorithm to assign the initial pheromone for the ant colony optimization. And the hybrid algorithm was applied to solve the multicriteria supplier selection model. Engin et al. proposed a hybrid ant colony optimization based on crossover and mutation mechanism, which was applied to solve the problem of no-waiting flow shop scheduling. [13] Li et al. proposed a pheromone update method that combines global asynchronous and elite strategies. It used particle swarm optimization to optimize the parameters of ant colony optimization, which could reasonably reduce the time cost of the algorithm while ensuring the quality of the solution. [14] You et al. proposed an improved ant colony optimization (DSACS) that mixed simulated annealing dynamic search inducer and 3-opt operator. [15] The simulated annealing search induced operator was used to control the search direction and convergence speed of the algorithm, and then the local optimization ability of the 3-opt operator was used to further improve the quality of the optimized solution. Wu et al. used the ant colony optimization to train the weights of the BP neural network, and

proposed an ant colony neural network algorithm that has both the nonlinear mapping ability of the neural network and the fast global convergence ability of the ant colony optimization [16].

2.2 Improved Chemical Reaction Optimization

Nayak et al. proposed a new high-order neural network algorithm based on chemical reaction optimization (CRO-HONN), which applied the chemical reaction optimization to the weight set training of a single hidden layer high-order neural network (PSNN). [17] The improved algorithm combined the fast learning of chemical reaction optimization and the strong nonlinear classification capabilities of high-order neural network. And through the large number of experimental simulations, it was proved that the performance of CRO-HONN is better than other optimization techniques, with higher classification accuracy and lower error rate. Shaheen et al. proposed a hybrid algorithm (GCRO) based on chemical reaction optimization and greedy algorithm to solve the traveling salesman problem. [18] In the initialization phase of chemical reaction optimization, a greedy algorithm is introduced to generate a better initial molecular population to improve the quality of the initial solution and speed up the convergence of the optimal solution. Combining the local search ability of chemical reaction optimization with the global search ability of particle swarm optimization, Nguyen et al. proposed a hybrid optimization method HP-CRO. [19] The update operator of particle swarm optimization is combined with the local search operator of chemical reaction optimization to improve the efficiency of the hybrid algorithm. In the chemical reaction optimization design framework, it retains its own advantages and incorporates the advantages of other algorithms to obtain a hybrid algorithm with better solution performance, which improves the scope of application of the algorithm.

3 Basic Ant Colony Optimization

The basic principle of ant colony optimization comes from the shortest path principle of ants foraging in nature. When ants are looking for food sources, they transmit information between individuals by releasing pheromone on the path traveled, so that they can efficiently find the shortest path for searching for food. In recent decades, in order to solve different problems in many engineering fields, various linear and non-linear algorithms have been designed and implemented. And the traveling salesman problem (TSP) as a benchmark has been widely used in various optimization techniques and heuristic searches. This paper still uses this classic TSP as an example to introduce the ant colony optimization model. [20]

For TSP, in order to maintain the generality of the algorithm, we suppose the number of cities is n; the number of ants is m; the distance between the city i and the city j is $d_{ij}\left(i, j=1,2,...,n\right)$; the pheromone concentration on the path between the city i and the city j at the moment t is $\tau_{ij}\left(t\right)$. At the initial moment, the pheromone concentration on all paths is equal. Each ant k is randomly placed in a different city, and then the next city to be visited is selected according to a certain selection probability $p_{ij}^{k}\left(t\right)$. Here $p_{ij}^{k}\left(t\right)$ is also called the transition probability of ant k from city i to city j at time t, which is the path selection process of ants. The specific formula is shown in formula (1):

$$p_{ij}^{k}\left(t\right)=\begin{cases}\dfrac{\left[\tau_{ij}\left(t\right)\right]^{\alpha}\bullet\left[\eta_{ij}\left(t\right)\right]^{\beta}}{\sum\limits_{s\in allow_k}\left[\tau_{is}\left(t\right)\right]^{\alpha}\bullet\left[\eta_{is}\left(t\right)\right]^{\beta}}, & j\in allow_k\\ 0, & j\notin allow_k\end{cases}. \tag{1}$$

Where $\eta_{ij}\left(t\right)$ is called heuristic function, which represents the expectation of ants from city i to city j at time t, $\eta_{ij}\left(t\right)=1/d_{ij}$; $allow_k$ is the collection of cities to be visited by each ant k; α is the pheromone factor, which reflects the degree of influence of the ant by the pheromone; β is the heuristic function factor, which reflects the degree of influence of the ant by the heuristic function. Too large or too small values of α and β will affect the randomness and search ability of the algorithm, and hinder the algorithm from finding the optimal solution. When each ant has walked through all cities, that is, after completing a cycle, the pheromone concentration on the path between cities will increase when the ant

secretes the pheromone, and it will also decrease when the pheromone is volatilized. The specific update mechanism is as follows:

$$\tau_{ij}\left(t+1\right)=\left(1-\rho\right)\bullet\tau_{ij}\left(t\right)+\Delta\tau_{ij},\ \ 0<\rho<1. \tag{2}$$

$$\Delta\tau_{ij}=\sum_{k=1}^{m}\Delta\tau_{ij}^{k}. \tag{3}$$

Generally, the value of $\Delta\tau_{ij}^{k}$ can be calculated according to the Ant Cycle System model: [21]

$$\Delta\tau_{ij}^{k}=\begin{cases}\dfrac{Q}{L_k}, & ant\ \text{k goes from city i to city j}\\ 0, & \text{otherwise}\end{cases}. \tag{4}$$

Where ρ is called pheromone volatilization factor, which reflects the volatilization degree of pheromone; $\Delta\tau_{ij}^{k}$ represents the pheromone concentration increased by all ants secreting pheromone on the path between the city i and the city j; $\Delta\tau_{ij}^{k}$ represents the pheromone concentration increased by the secretion of pheromone by the ant k on the path between the i and the j; Q is called the pheromone constant, which represents the total amount of pheromone secreted by each ant after walking through all the cities; L_k represents the length of the path that the ant k passes. Formula (2) is to simulate the renewal process of pheromone, including the volatilization of pheromone and the superposition of pheromone on the path that the ant passes. Formula (3) is to superimpose new pheromone on the path that the ant walks, and the value of $\Delta\tau_{ij}^{k}$ is calculated according to formula (4).

4 Adaptive Chemical Reaction Ant Colony Optimization

4.1 Basic Principles of Adaptive Chemical Reaction Ant Colony Optimization

The ant colony algorithm is robust and has a strong ability to find the optimal solution. Its pheromone positive feedback mechanism greatly accelerates the evolution process, while the ant colony algorithm also has the following shortcomings. [22] First, the ant colony optimization is looking for a better local optimal solution instead of forcing the global optimal solution. Second, the positive feedback mechanism of the ant colony optimization makes the algorithm converge faster at the initial stage, but in the later optimization process, after a certain number of iterations, it is prone to stagnation.

The adaptive chemical reaction optimization ACRO is an algorithm developed by Yu et al. to improve the chemical reaction optimization CRO. [23] CRO is inspired by the nature of chemical reactions and is a meta-heuristic algorithm. In chemical reactions, the reactants often release superfluous energy to the environment and approach to a lower energy state. CRO uses this trend and combines the idea of chemical reaction to construct an optimization algorithm. On the premise of CRO, ACRO reduces the number of optimized parameters in CRO to reduce the workload of parameter adjustment. At the same time, an adaptive scheme is developed to evolve them. While the ACRO algorithm retains the advantages of the CRO algorithm, it can better adapt to different optimization problems, and its performance is better than the CRO algorithm.

The chemical reaction optimization is a design framework, there is such a characteristic that other attributes can be incorporated into the molecular structure. By combining the advantages of the two algorithms, an improved ant colony optimization based on adaptive chemical reaction optimization is proposed. ACRACO combines the pheromone feedback mechanism of ant colony optimization, the local optimization ability of ant colony optimization and the global optimization ability of adaptive chemical reaction optimization. And the decomposition reaction operations and synthesis reaction operations are introduced into the ant colony optimization to make the ant colony jump out of the local optimum and avoid the stagnation of the algorithm.

When a certain number of molecules collide in a closed container to cause a chemical reaction,

different molecular structure ω represents different solution to the problem. The molecule is the main body to perform operations of the algorithm. Each molecule contains kinetic energy KE and potential energy PE, and follows the law of conservation of energy. Since only the decomposition reaction and synthesis reaction are introduced in the ACRACO, the kinetic energy KE is ignored and only the molecular potential energy PE is considered. The mathematical attribute of PE here is the objective function, that is, the shortest path in the TSP.

(1) Decomposition Reaction

There is only one molecule involved in the decomposition reaction. It collides violently with the container wall, and then produces two new molecules. The molecular structures of the new molecules are quite different from the original molecule, and the energy change is also great. Assuming that the original molecular structure is ω, the new molecular structures generated are ω_1' and ω_2' respectively. Then the conditional formula for the decomposition reaction is shown in formula (5):

$$PE_{\omega} \geq \min\left(PE_{\omega_1'}, PE_{\omega_2'}\right). \tag{5}$$

Where $PE_{\omega_1'}$ and $PE_{\omega_2'}$ represent the potential energy of the two newly generated molecules respectively.

(2) Synthesis Reaction

There are two molecules involved in the synthesis reaction. The collision between molecules is violent, and then a new molecule is produced. There is a huge gap between the new molecular structure and the original molecular structure, and the energy changes is also great. Assuming that the original molecular structures are ω_1 and ω_2 respectively, the new molecular structure generated is ω'. Then the conditional formula for the synthesis reaction is shown in formula (6):

$$\max\left(PE_{\omega_1}, PE_{\omega_2}\right) \geq PE_{\omega'}. \tag{6}$$

Where $PE_{\omega'}$ represents the potential energy of the new created molecule.

In order to control the frequency of decomposition reaction, control the frequency of synthesis reaction and maintain molecular diversity, we assume that during the search process, the number of molecules is similar with the initial number of molecules when the number of molecules is stable. Then the paper introduces the initial reaction change rate ChangeRate to describe the probability of decomposition reaction and synthesis reaction, and introduces parameters f_{pop} to control the occurrence of decomposition reaction and synthesis reaction. These make the decomposition reaction more likely to occur when the number of molecules is small, and the synthesis reaction more likely to occur when the number of molecules is large. And the specific formula of f_{pop} is shown in formula (7):

$$f_{pop} = \frac{1}{2} \times \left(1 - \frac{curPopSize - iniPopSize}{iniPopSize}\right). \tag{7}$$

Where *iniPopSize* is the value of the initial number of molecules; *curPopSize* is the value of the current number of molecules. When $\text{ChangeRate} \leq f_{pop}$ or the current number of molecules is 1, the decomposition reaction is executed; otherwise, the synthesis reaction is executed.

According to the idea in the literature [24], the specific operations of the decomposition reaction and the synthesis reaction are as follows. Assuming that the number of cities in TSP is 5, in the decomposition reaction operator, a path A is randomly selected to represent a molecular structure ω. Then the reaction operator divides molecule A into two parts by drawing a random serial number. The value range of this random serial number is an integer in [1, 5]. For the two new molecules generated by the decomposition reaction, the left part of the A_1 molecule retains the left part of the random serial number of the original molecule A, and the right part is randomly assigned by the remaining nodes; the right part of the A_2 molecule retains the right part of the random serial number of the original molecule A, and the left part is randomly assigned by the remaining nodes. As shown in Fig. 1, the randomly selected random serial number is 3, A_1 and A_2 are two new molecular structures obtained after the decomposition reaction is completed.

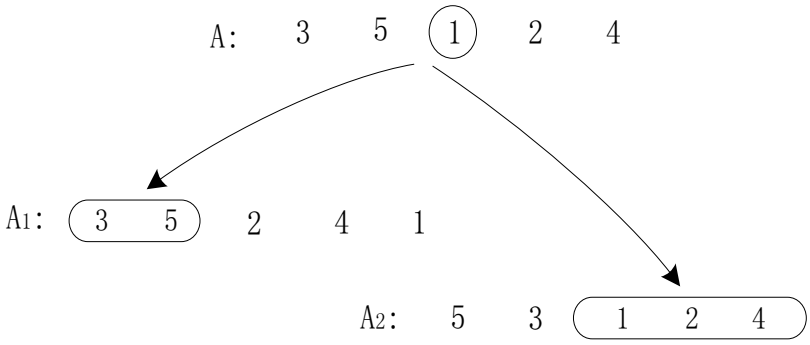

Fig. 1. Decomposition reaction operator

In the synthesis reaction operator, as shown in Fig. 2, two paths A and B are randomly selected as the original molecules. Then a molecular node is randomly selected (note that this is not a random serial number). Assuming that the selected node is "1", the reaction operator will respectively rotate the node loop of molecule A and molecule B to the right until "1" becomes the first node of molecule A and molecule B. Then the operator respectively calculates the distance between the first node "1" and the next node in numerator A and numerator B, and judges its size. When the distance between the first node in A and the next node is larger, the reaction operator keeps the first node of A and the whole of B unchanged, and the remaining node loops of A continues to rotate to the right until the second node of A is the same as B. Before the second rotation starts, the value "1" of the first node remains unchanged and is copied to the new molecule AB as the first node of AB. Then the reaction operator removes the first node "1" in molecule A and molecule B, so that the second node of molecule A and molecule B becomes the first node. The operator rotates in this way to the end to get a brand-new molecule AB, which is the new molecule obtained after the synthesis reaction is completed.

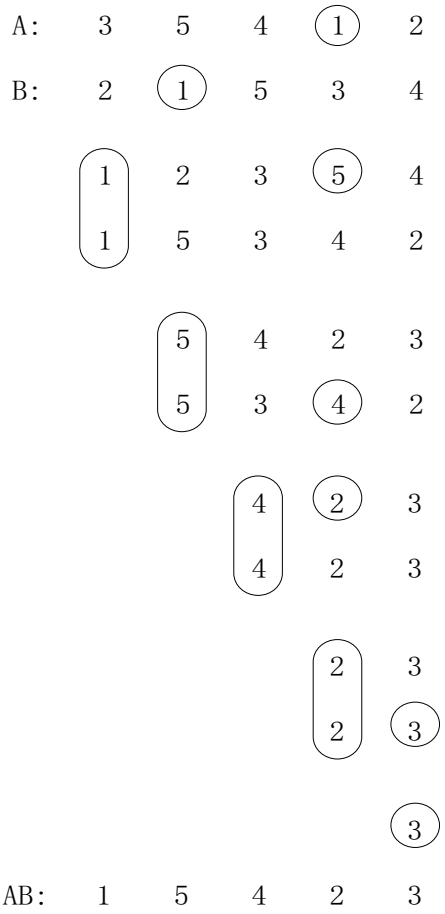

Fig. 2. Synthesis reaction operator

4.2 Algorithm Flow

Step 1: Initialize the relevant parameters of the ant colony optimization and calculate the distance matrix between nodes.
Step 2: Construct a solution space, randomly place ants at different starting points; calculate the transition probability of each ant according to formula (1), select the next access node, and record it in the path record table; repeat this process until all ants have visited all nodes.
Step 3: Calculate the length of the path that each ant passes, and record the optimal solution in the current iterations; if the number of current iterations is larger than 1, and the optimal solution of the current iteration is equal to the optimal solution of the previous iteration, an adaptive chemical reaction optimization algorithm is introduced and go to the next step; otherwise, skip to step 7.
Step 4: Initialize the relevant parameters of the adaptive chemical reaction optimization, including the initial reaction molecule record table (that is, the ant path record table), the potential energy record table, the initial reaction change rate, the initial reaction times, etc.
Step 5: Introduce decomposition reaction and synthesis reaction operation. If the path length is improved after the reaction is completed, update the initial reaction molecule record table, potential energy record table and initial reaction times.
Step 6: If the molecular potential energy obtained after the initial reaction is less than the optimal solution of the current iteration in the ant colony optimization, the adaptive chemical reaction optimization is ended. And replace the shortest path in the ant colony optimization with the molecule with the smallest potential energy, update the optimal solution that records the current iteration times. Otherwise, when the total number of initial reactions reaches the maximum number of reactions $NR\_\max$, the adaptive chemical reaction optimization is ended, and the longest path in the ant colony optimization is replaced with the molecule with the smallest potential energy.
Step 7: Update the pheromone on the path according to formulas (2)-(4); clear the path record table, and jump to step 2 to continue iterating.
Step 8: When the algorithm reaches the set maximum number of iterations $NC\_\max$, the algorithm iteration is terminated, and the optimal path and length are output.

5 Algorithm Testing and Analysis

In order to test the performance and effectiveness of the adaptive chemical reaction ant colony optimization (ACRACO), the paper uses MATLAB R2016b to do simulation test for six classic data sets (burma14、bays29、dantzig42、berlin52、st70、eil101) in the TSPLIB standard library. [25]And the paper compares its optimization results with those of the basic ant colony optimization (ACO). During simulation, the maximum number of reactions is set to $NR\_\max=500$. According to the literature [26], the other adjustable coefficients are respectively set as: number of ants is $m=50$, pheromone factor is $\alpha=1$, heuristic function factor is $\beta=5$, pheromone volatilization factor is $\rho=0.1$, pheromone constant is $Q=500$, and the maximum number of iterations is $NC\_\max=200$.

(1) Algorithm Optimization Ability

The paper evaluates the optimization ability of the algorithm through the optimal value and average value obtained by simulation. Table 1 shows the results of 30 times' simulations for data sets burma14, bays29, dantzig42, berlin52, st70 and eil101 by ACO and ACRACO respectively.

Table 1. The Results of 30 Times' Simulations for TSP

TSP Data	TSP Optimal Solution	Simulation Optimal Solution		Simulation Worst Solution		Average Value		Deviation Rate /%	
		ACO	ACRACO	ACO	ACRACO	ACO	ACRACO	ACO	ACRACO
burma14	NA	30.8785	30.8785	31.2269	30.8785	31.2123	30.8785	—	—
bays29	9291.3525	9148.3435	9074.148	9318.6158	9077.9177	9273.0440	9075.2186	-1.5392	-2.3377
dantzig42	NA	697.8502	679.2019	748.1140	686.7984	727.0581	679.9505	—	—
berlin52	7544.3659	7663.5851	7544.3659	7818.0552	7548.9927	7699.1784	7544.6862	1.5802	0
st70	NA	708.7990	677.1096	724.9847	691.7047	713.4747	686.0063	—	—
eil101	642.3095	688.8735	652.1096	706.1799	663.6686	697.2200	656.8819	7.2495	1.5258

Where the TSP optimal solution is the length value calculated according to the optimal path provided in the TSPLIB standard library; NA means that the TSPLIB standard library does not provide the optimal path for the data; the deviation rate indicates the degree of deviation between the optimal value obtained by the algorithm simulation and the optimal solution of TSP. The specific calculation formula of the deviation rate is shown in formula (8):

$$\text{deviation rate} = \frac{\text{simulation optimal solution-TSP optimal solution}}{\text{TSP optimal solution}} \times 100\% . \quad \textbf{(8)}$$

It can be seen from Table 1 that when the common parameter settings are the same, the optimal solution, worst solution and average value obtained by ACRACO simulation are basically better than the ACO simulation results. For the data set burma14, the optimal path length obtained by ACRACO simulation is 30.8785, which is the same as ACO, but the worst solution and average value obtained by simulation are 0.3484 and 0.3338 less than ACO respectively. For the data set bays29, the optimal solution obtained by ACRACO and ACO simulation is smaller than the TSP optimal solution, but ACRACO is still 74.1955 less than ACO, and the average value is still 197.8254 less than ACO. For the data set berlin52, the optimal solution obtained by ACRACO simulation is the same as the optimal solution of TSP, which is 119.2192 less than ACO, and the average value is also 154.4922 less than ACO. For the data set eil101, although neither ACRACO nor ACO find the same or smaller path as the TSP optimal solution, ACRACO still shows better performance compares with ACO. The optimal solution and average value obtained by ACRACO simulation are 36.7639 and 40.3381 smaller than ACO respectively, and the deviation rate is also 5.7237% smaller. For the data sets dantzig42 and st70, although the corresponding TSP optimal solution is not found, compares with ACO, ACRACO can still find a better solution. For the data set dantzig42, the optimal solution and average value obtained by ACRACO simulation are 18.6483 and 47.1076 less than ACO respectively. And for the data set st70, the optimal solution and average value obtained by ACRACO simulation are 31.6894 and 40.3381 less than ACO respectively. In summary, ACRACO has better global optimization capabilities than ACO. In order to show this advantage more intuitively, the optimized paths and optimization curves obtained by the two algorithms on the data sets bays29, berlin52 and eil101 are respectively given, as shown from Fig. 3 to Fig. 11.

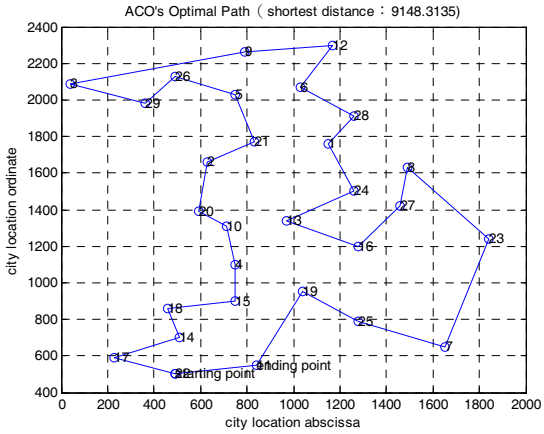

Fig. 3. The optimal path of ACO simulation for bays29

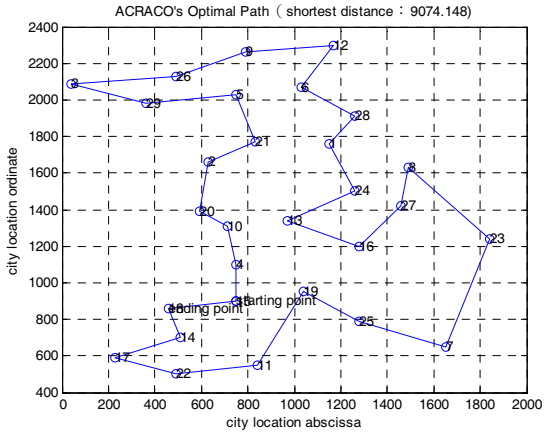

Fig. 4. The optimal path of ACRACO simulation for bays29

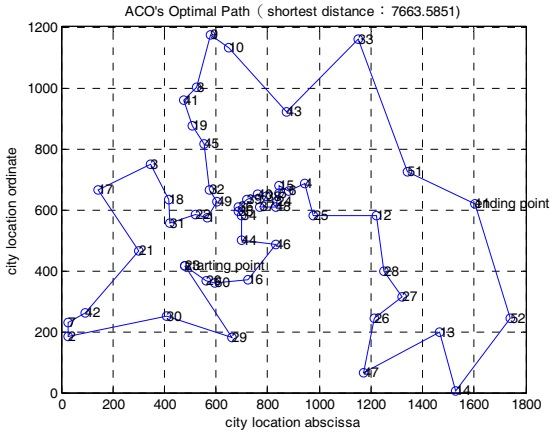

Fig. 5. The optimal path of ACO simulation for berlin52

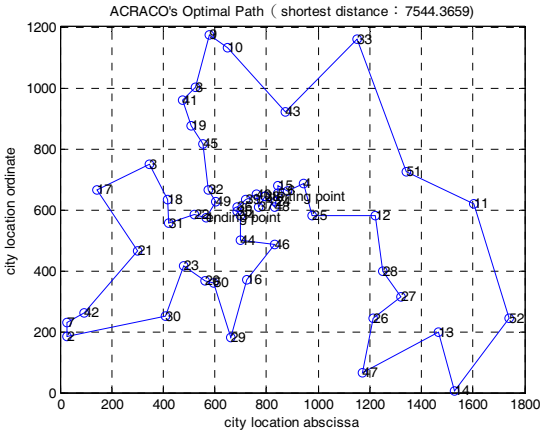

Fig. 6. The optimal path of ACRACO simulation for berlin52

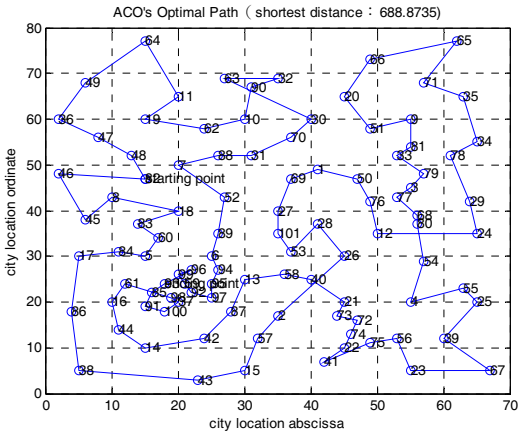

Fig. 7. The optimal path of ACO simulation for eil101

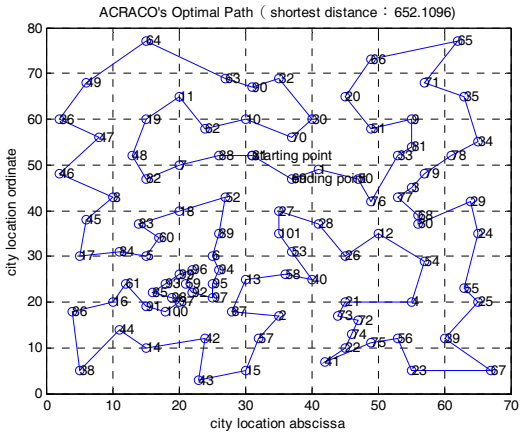

Fig. 8. The optimal path of ACRACO simulation for eil101

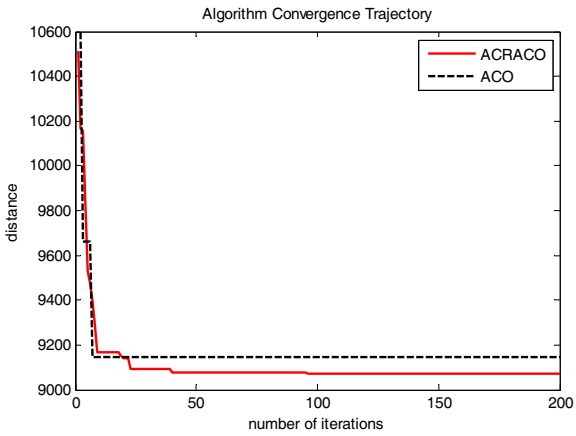

Fig. 9. The optimization curve of the data set bays29

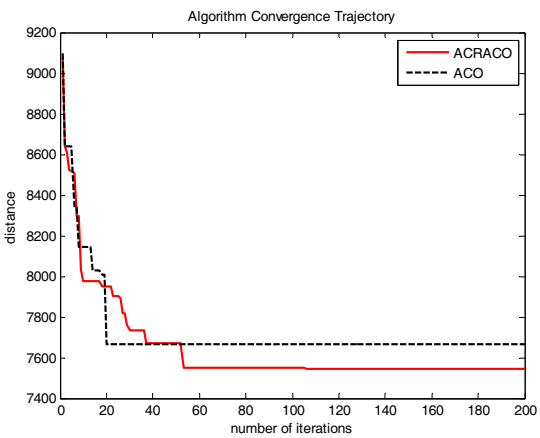

Fig. 10. The optimization curve of the data setberlin52

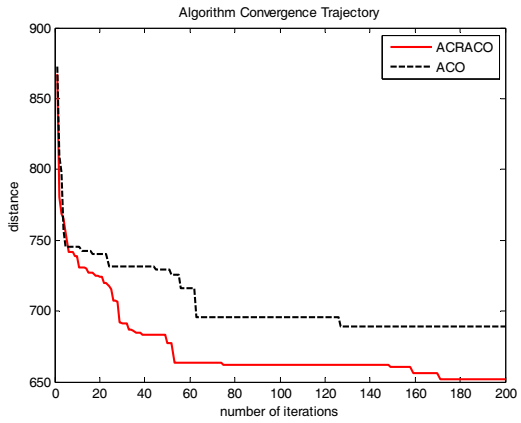

Fig. 11. The optimization curve of the data seteil101

According to the No Free Lunch Theorem, we can know that no learning algorithm can always produce the most accurate learner in any field. [27] While the algorithm improves the speed of solving a certain type of problem, it will inevitably reduce the speed of solving another type of problem. The adaptive chemical reaction ant colony optimization also shows this no free lunch effect. It improves the algorithm's ability to find the shortest path while making the algorithm's program running time longer. There is no specific analysis for it due to objective reasons such as experimental conditions. But in the future research work, a potential research direction is whether we can improve the algorithm optimization ability while shortening the running time of the program as much as possible, so that they can reach a balanced state.

(2) Algorithm Stability and Reliability

The stability and reliability of the algorithm are evaluated by the standard deviation and success rate obtained in each simulation. In the basis of a preset threshold, the success rate refers to the ratio of the number of times that the result of the algorithm reaches the threshold and the total number of runs. Table 2 shows the stability and reliability comparison results obtained by ACO and ACRACO for 30 times' simulations of data sets burma14, bays29, dantzig42, berlin52, st70 and eil101.

Table 2. Stability and Reliability Comparison Results

TSP Data	Standard Deviation		Success Rate /%		Threshold
	ACO	ACRACO	ACO	ACRACO	
burma14	0.0623	0	100	100	32
bays29	43.6211	1.4165	70	100	9291
dantzig42	13.2763	1.4777	10	100	710
berlin52	39.6660	1.1527	73.33	100	7700
st70	3.6672	3.1984	20	100	710
eil101	4.5919	3.2488	60	100	699

According to Table 2, the standard deviations obtained by ACRACO's simulation for the six data sets are all less than ACO, and the solution success rate of ACRACO is 100%. It shows that the stability and reliability of the adaptive chemical reaction ant colony optimization are better than the basic ant colony optimization.

6 Comparative Analysis with DSO

Donkey and smuggler optimization algorithm (DSO) is a new optimization algorithm recently proposed. It has been successfully applied to solve the traveling salesman problem, and it has obtained better optimization results than the basic ant colony algorithm. [8] Therefore, taking the data sets dantzig42 and berlin52 in TSP as examples, this paper compares the optimization results of ACRACO and DSO, and the specific solution results are shown in Table 3:

Table 3. Simulation Results of ACRACO and DSO

TSP Data	Simulation Optimal Solution		Simulation Worst Solution		Average Value	
	DSO	ACRACO	DSO	ACRACO	DSO	ACRACO
dantzig42	822.0950	679.2019	822.0950	686.7984	822.0950	679.9505
berlin52	8182.1916	7544.3659	8182.1916	7548.9927	8182.1916	7544.6862

According to Table 3, although the donkey and smuggler optimization algorithm has excellent solution stability, the adaptive chemical reaction ant colony optimization can find a shorter distance traveling salesman path. It shows the superiority of adaptive chemical reaction ant colony optimization in solving medium-scale traveling salesman problem.

7 Conclusions

In order to improve the shortcomings of ant colony algorithm, which are prone to stagnation and fall into local optimality, this paper combines the advantages of ant colony optimization and adaptive chemical reaction optimization. And the paper introduces the decomposition and synthesis operations of the adaptive chemical reaction optimization into the ant colony optimization, which enhances the algorithm's global search capabilities. And then an improved ant colony optimization based on adaptive chemical reaction optimization ---- adaptive chemical reaction ant colony optimization (ACRACO) is proposed. At the same time, six classic data sets in the Traveling Salesman Problem (TSP) are used for performance testing. The results show that the adaptive chemical reaction ant colony algorithm is superior to the basic ant colony optimization in terms of algorithm optimization ability, algorithm stability and algorithm reliability.

The focus of research and improvement in future work is: improve the problem of long program running time in the adaptive chemical reaction ant colony optimization, so that the algorithm can achieve a better balance between the optimization ability and the running speed; provide mathematical theoretical support for parameter values; improve the ability of adaptive chemical reaction ant colony optimization to solve large-scale urban traveling salesman problems, etc. In addition, it has been a long time since the ant colony optimization was proposed, and its performance has been slightly behind. Therefore, another direction of future work is to study the recently proposed new algorithms, or to propose new algorithms with better performance.

Acknowledgements

This work is support by Tianjin Natural Science Foundation (No. 20JCYBJC00320）and College Students' innovation and Entrepreneurship Project (No. 202010069062)

References

[1] A. Colorni, M. Dorigo, V. Maniezzo, Distributed optimization by ant colonies. European Conference on Artificial Life, 1991.

[2] Y. Wang, L. Wang, G.C. Chen, Z.Q. Cai, Y.Q. Zhou, L.N. Xing, An Improved Ant Colony Optimization algorithm to the Periodic Vehicle Routing Problem with Time Window and Service Choice, Swarm and Evolutionary Computation 55(2020) 100675.

[3] W. Qin, Z.L. Zhuang, Y. Liu, O. Tang, A two-stage ant colony algorithm for hybrid flow shop scheduling with lot sizing and calendar constraints in printed circuit board assembly, Computers & Industrial Engineering 138(2019) 106115.

[4] C.W. Miao, G.Z. Chen, C.L. Yan, Y.Y. Wu, Path planning optimization of indoor mobile robot based on adaptive ant colony algorithm, Computers & Industrial Engineering 156(2021) 107230.

[5] V. Sharma, R.N. Mir, An enhanced time efficient technique for image watermarking using ant colony optimization and light gradient boosting algorithm, Journal of King Saud University - Computer and Information Sciences 2019.

[6] H.E. Tseng, C.C. Chang, S.C. Lee, Y.M. Huang, Hybrid bidirectional ant colony optimization (hybrid BACO): an algorithm for disassembly sequence planning, Engineering Applications of Artificial Intelligence 83(2019) 45-56.

[7] A.Y.S. Lam, V.O.K. Li, Chemical-Reaction-Inspired Metaheuristic for Optimization, IEEE Transactions on Evolutionary Computation 14(2010) 381-399.

[8] A.S. Shamsaldin, T.A. Rashid, R.A.A. Agha, N.K. Al-salihi, M. Mohammadi, Donkey and smuggler optimization algorithm: a collaborative working approach to path finding, Journal of Computational Design and Engineering 6(4)(2019) 562–583.

[9] S. Ebadinezhad, DEACO: Adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem, Engineering Applications of Artificial Intelligence 92(2020).

[10] T. Stvtzle, H.H. Hoos, MAX-MIN Ant System, Future Generation Computer Systems 8(2016) 889-914.

[11] Z. Yang, Theoretical research and improvement of chemical reaction optimization, Nantong University, 2014.

[12] J. Luan, Z. Yao, F. Zhao, X. Song, A novel method to solve supplier selection problem: Hybrid algorithm of genetic algorithm and ant colony optimization, Mathematics and Computers in Simulation 156(2019) 294-309.

[13] O. Engin, A. Gvclv, A new hybrid ant colony optimization algorithm for solving the no-wait flow shop scheduling problems, Applied Soft Computing 72(2018) 166-176.

[14] Q. Li, C. Zhang, P. Chen, Y. X. Ying, An improved ant colony algorithm based on particle swarm parameter optimization, Control and Decision 28(06)(2013) 873-878+883.

[15] X.M. You, S. Liu, J.Q. Lv, A dynamic search strategy ant colony algorithm and its application in robot path planning, Control and Decision 32(03)(2017) 552-556.

[16] D.M. Wu, J.P. Shao, Y.N. Bing, Research on fault diagnosis technology of CNC machine tool based on ant colony algorithm and neural network, Machinery Design & Manufacture 1(2013) 165-167.

[17] J. Nayak, B. Naik, H.S. Behera, A novel chemical reaction optimization based higher order neural network (CRO-HONN) for nonlinear classification, Ain Shams Engineering Journal 6(2015) 1069-1091.

[18] A. Shaheen, A. Sleit, S. Al-sharaeh, An improved chemical reaction optimization algorithm for solving traveling salesman problem, in: Proc. 2018 9th International Conference on Information and Communication Systems, 2018.

[19] T.T. Nguyen, Z. Li, S. Zhang, T.K. Truong, A hybrid algorithm based on particle swarm and chemical reaction optimization, Expert Systems with Applications 41(2014) 2134-2143.

[20] H.B. Duan, The principle and application of ant colony algorithm, Beijing: Science Press, (2005) 24-43.

[21] L. Ma, G. Zhu, A.B. Ning, Ant Colony Optimization Algorithm, Beijing: Science Press, 2007.

[22] J.W. Zhuo, B.W. Li, Y.S. Wei, J. Qin, Application of MATLAB in mathematical modeling, Beijing: Beihang University Press, 2014.

[23] J.J.Q. Yu, A.Y.S. Lam, V.O.K. Li, Adaptive chemical reaction optimization for global numerical optimization, IEEE Congress on Evolutionary Computation (2015) 3192-3199.

[24] Y.C.H. Ou, Research on chemical reaction optimization algorithm for solving TSP Problem, Hunan University, 2014.

[25] TSPLIB[EB/OL]. <http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/tsp/>, 2021 (accessed 21.03.17)

[26] Y. Wang, L.Y. Zhang, T. Fei, X.F. Zhou, Simulated annealing ant colony algorithm with chaotic disturbance for solving TSP, Computer Engineering and Design 37(04)(2016) 1067-1070+1112.

[27] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, IEEE Transactions on Evolutionary Computation 1(1)(1997) 67-82.