

## Research on the Hybrid ant Colony Algorithm based on Genetic Algorithm

Xu Yan

*Xi'an Physical Education University, Shaanxi, China, 710068*  
*xuyanmini13@163.com*

### **Abstract**

*Since the ant colony algorithm is proposed, it has achieved the remarkable achievements in many fields. With the development of the times, the traditional ant colony algorithm exposes its limitations for solving the questions. In this paper, we improve the ant colony algorithm. And we combine the ant colony algorithm with the genetic algorithm. Then, we propose the GAPSPAC algorithm. The algorithm combines the advantages of the genetic algorithm and the ant colony algorithm. And it overcomes the disadvantages to improve the efficiency of solving the questions. In the last experiment, we can see the algorithm has the better problem solving ability and the stability.*

**Keywords:** *Ant colony algorithm; Genetic algorithm; Combinatorial optimization*

### **1. Introduction**

The ant colony algorithm is proposed by the Italian scholar M.Dorigo in 20 century 90 times. According to simulating the behavior that the ants found out the path in nature, he proposed the heuristic evolutionary algorithm which was based on the population [1-3]. According to studying the foraging behavior of the ants, they found that the ant colony collaborate by the chemical substance. The substance was called pheromones. According to the pheromones, they formed the positive feedback and made more and more ants gather to the shortest path. Since the ant colony algorithm was proposed, it has been paid attention by many scholars and has been widely used in many fields.

Dorigo did e double bridge experiment to illustrate the process that the ant colony found the shortest path. According to the experiment, Dorigo explained the process for the ant colony from the random path to the selected path according to the guideline of the pheromones. Santos and other people proposed an improved ant colony algorithm to solve the capacitated arc routing problem which had the capacity constraints. The algorithm improved from two aspects of path transfer rules and the local search. And it accelerated the convergence speed for the optimal solutions [4]. Zhang, Lin and Xue proposed a kind of the self-adaptive heterogeneous multi-ant colony algorithm to solve the TSP questions. According to using the evolutionary coefficients, the algorithm evaluated the solutions for different ant colonies. Then, according to the quality, the algorithm updated the pheromone in the path and accelerated the convergence speed for the optimal solution [5]. Gajpal and Abad proposed the improved ant colony system algorithm. The algorithm can solve the questions of the vehicle routing which loaded and unloaded the goods. Under the process of research, the algorithm used the multi-path search principle to search the customers in different vehicle path and expanded the search scope [6]. The algorithm which was proposed by Gajpal can solve the vehicle routing which can come back. The algorithm used a new path construction method. The local research adopted the multi-path search method and can research the customers in different vehicle path. The algorithm expanded the search scope [7]. Yang and other people proposed a multi-ant colony algorithm which was used to solve the data clustering questions. In this algorithm, each ant colony not only had its own pheromone, but also they had the different moving speed

and different transition probability. This method can expand the search space and increase the diversity of the search [8]. Urszula Boryczka and Jan Kozak merged the methods that had been developed for better construction of decision trees by ants. The Ant Colony Decision Tree (ACDT) approach was tested in the context of the bi-criteria evaluation function by focusing on two problems: the size of the decision trees and the accuracy of classification obtained during ACDT performance [9]. The empirical results clearly show that the ACDT algorithm creates good solutions which are located in the Pareto front [9]. M.Z.M. Kamali, N. Kumaresan, Kuru Ratnavelu implemented a nontraditional modified ant colony programming (ACP) method in the present work. The modified ACP algorithm was unique as it did not use the criteria of distance. It utilized the probability function which related to the quantity of the pheromone level in the ACP [10].

As a simulated evolutionary algorithm, the ant colony algorithm had widely used in combinatorial optimization [10-12], function optimization [13-16] and path planning [17-20] etc. and it achieved better results. However, the traditional ant colony algorithm cannot keep pace with the times. Aiming at the disadvantages of the initial pheromone and the lower efficiency, this paper combined the genetic algorithm with the ant colony algorithm. At the same time, we improve the ant colony algorithm and proposed the GAPSOAC algorithm. The structure of this paper is as follows. The first part is the introduction. In this part, we introduce the research background of the ant colony algorithm. The second part is the basic theory. We introduced the ant colony algorithm, genetic algorithm and the PSO algorithm. The third part is the GAPSOAC algorithm. In this part, we combine the genetic algorithm with the ant colony algorithm and propose the GAPSOAC algorithm. The fourth part is the experiment. And the last part is the conclusion.

## 2. Basic Theory

### 2.1. Mathematical Description of the Ant Colony Algorithm

In order to describe the ant colony algorithm, we applied the ant colony algorithm to the TSP question.  $m$  is the number of ants.  $n$  is the number of the cities.  $d_{ij}$  is the distance from the city  $i$  to the city  $j$ .  $\tau_{ij}(t)$  is the amount of the pheromone at  $t$  time.

The ants in ant colony algorithm had the following characteristics.

Firstly, when traversing the path, the ants will release the pheromone.

Secondly, when selecting the next path, the ants select the path according to the probability which contains the pheromone function.

Thirdly, in order to satisfy the constraint conditions, before the ants complete a cycle of search, it was not allowed to visit the city which has visited.

At the initial time, the amount of pheromone in each path is the same. We assume that

$$\tau_{ij}(0) = C \quad (1)$$

$C$  is a constant. When the ant  $K$  finds the optimal path, it determines the next transfer direction according to the density of the pheromone in each path. At  $t$  time, the transfer probability for the ant  $k$  between the city  $i$  and the city  $j$  can be expressed as follows.

$$P_{ij}(t) = \begin{cases} \frac{[\tau_{ij}]^\alpha \cdot [\eta_{ij}]^\beta}{\sum_{s \in allowed_k} [\tau_{is}]^\alpha \cdot [\eta_{is}]^\beta}, & \text{if } j \in allowed_k \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Among them,  $allowed_k$  represents the next city that the ant is allowed to enter.  $\alpha$  represents the relative important information heuristic factor.  $\beta$  represents the expected heuristic factor that the visibility is related important.  $\eta_{ij}$  represents the heuristic function. And the calculation formula is as follows.

$$\eta_{ij} = \frac{1}{d_{ij}} \quad (3)$$

When the ant completes a cycle of research, the update formula of pheromone is as follows.

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

$\rho$  represents the residual coefficient of the pheromone.  $(1-\rho)$  represents that the residual factor of the pheromone.  $\Delta\tau_{ij}$  is the left amount of information in the path. And it is the incremental sum of the pheromone from the path  $i$  to  $j$ .

$$\Delta\tau_{ij}(t) = \sum_{k=1}^m \Delta\tau_{ij}^k(t) \quad (5)$$

## 2.2. The Basic Steps of the Genetic Algorithm

The genetic algorithm includes the parameter design, the design of the fitness function, coding, the generation of the initial population, selection, crossover and mutation *etc.*

The first step is to determine the parameters of the algorithm. It contains the population size, selection probability, crossover probability, mutation probability and the evolution algebra *etc.*

The second step is to determine the fitness function. The genetic algorithm is based on the operation of the fitness. Therefore, the reasonable fitness function can reflect the advantages and the disadvantages of each individual. In addition, it can adapt the evolution process of the algorithm.

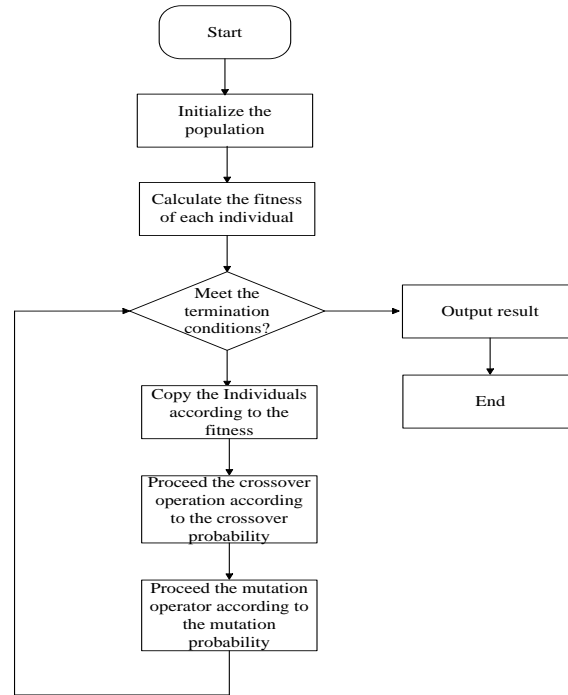
The third step is to determine the coding scheme of the problem. The genetic algorithm is usually not directly on the solution space of the problem. It uses the codes to express the evolution. Therefore, it has a great influence on the quality and the efficiency of the algorithm to select rational coding mechanism.

The fourth step is to generate the initial population. The genetic algorithm is the evolutionary operation for the population. Therefore, it must prepare the initial population which is composed by several initial solutions. Each individual of the initial population is selected by the stochastic method.

The fifth step is to design the genetic operator. It includes the selection, crossover and mutation *etc.*

The sixth step is the termination conditions of the algorithm. The termination criteria should make a reasonable balance or the focused in the operation quality and the efficiency according to the characteristics of the solution.

The specific work flow chart is as shown below.



**Figure 1. Flow Chart of Genetic Algorithm**

### 2.3. PSO Algorithm

The core idea of the particle swarm optimization algorithm is as follows. It finds the optimal path according to the collaboration and the information sharing of each individual. In the particle swarm algorithm, each bird is as an individual of the group. It is abstracted as a particle which has not the equality and volume. And it is extended to  $d$  dimensional space. If the number of group particles is  $n$ , the location of  $i(i=1,2,\dots,n)$  particle in  $d$  dimensional space can be expressed as a vector  $x_i = (x_{i1}, x_{i2}, \dots, x_{id})$ . The flight speed can be represented as a vector  $v_i = (v_{i1}, v_{i2}, \dots, v_{id})$ . It determines the fixed iteration displacement of the particle in the research space.

PSO algorithm is initialized to a group of random particle. Then, according to the iteration, it finds the optimal solution. During the process of the iteration, according to tracking the two extreme values, the particle updates itself. The two extreme values are  $pbest$  and  $gbest$ .

After finding the two optimal values, we update the speed and the position of the particle by the formula (6) and the formula (7).

$$v_i(t+1) = v_i(t) + c_1 r_1(t)(pbest_i(t) - x_i(t)) + c_2 r_2(t)(gbest_i(t) - x_i(t)) \quad (6)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (7)$$

$c_1$  and  $c_2$  are the learning factors. They are also called as the acceleration coefficient. They are used to adjust the functions of their own information and the global information during the motion. In general, the value is 2.

$r_1$  and  $r_2$  are the independent functions. The range is  $(0,1)$ .

The fitness value of each particle updates the global extreme value  $gbest$  and the individual extreme value  $pbest$  for the particle. Each particle uses the formula (8) to update the individual extreme.

$$pbest_i(t+1) = \begin{cases} x_i(t+1), & x_i(t+1) \geq pbest_i(t) \\ pbest_i(t) & x_i(t+1) < pbest_i(t) \end{cases} \quad (8)$$

In addition,

$$gbest_i(t) = \max(pbest_i(t+1)) \quad (9)$$

The maximum speed that the algorithm assumes is  $v_{max}$ . During the process of updating, the speed of the particle in each dimension should not exceed  $v_{max}$ .

That is,

If

$$v_i(t+1) > v_{max} \quad (10)$$

then

$$v_i(t+1) = v_{max} \quad (11)$$

If

$$v_i(t+1) < -v_{max} \quad (12)$$

then

$$v_i(t+1) = -v_{max} \quad (13)$$

### 3. GAPSOAC Algorithm

The ant colony algorithm is a kind of stochastic optimization algorithm which appears in recent years. It is a new bionic algorithm which sources from the nature. It is proposed by the Italian scholar. The ant colony algorithm achieves to the optimal path according to the information transform of the group. At first, it is called the ant colony optimization algorithm. Because it simulates the concept of the artificial ants, it is also called the ant system.

Genetic algorithm is a kind of heuristic algorithm. The algorithm has the ability of the quick and random global search. However, it cannot use the feedback information in the system. When the solution is a certain range, it will produce massive redundancy iteration. Then, it leads to the lower efficiency. According to the accumulation of the pheromone and the updating, the ant colony algorithm will converges to the optimal path. It has the distributed parallel global search ability. However, at first, the pheromone is little and the solution speed is slow. The ant colony algorithm is used in the solution of the pheromone intensity in most of the time.

In this paper, we combine the genetic algorithm with the ant colony algorithm. And we propose a hybrid algorithm. The basic thought of the hybrid algorithm which is based on the genetic algorithm and the ant algorithm is to learn the advantages and overcomes the disadvantages. Firstly, the algorithm adopts the genetic algorithm. It makes full use of the fast, random and global convergence of the genetic algorithm. The results are to produce the initial pheromone distribution of the related questions. Secondly, the algorithm adopts the ant colony algorithm. In the case that there are a certain of initial pheromone

distribution, the algorithm uses the parallelism, the positive feedback and the higher efficiency of the ant colony algorithm.

### 3.1. GAPSO Algorithm

GA algorithm and PSO algorithm have their own advantages. However, there are some disadvantages. When the GA algorithm iterates, it must go through the selection, crossover and mutation. The mechanism leads that the convergence time increases. However, there are two paths that the particles iterated in PSO algorithm. They are two extremes. *pbest* is the individual extreme. *gbest* is the global extreme. If the particle finds the optimal while it falls into the local optimum, PSO algorithm may fall into the local extreme value and it does not get the global optimal solution. Therefore, in this paper, we combine the GA algorithm and PSO algorithm firstly. The thought of GAPSO is as follows. Firstly, we use the PSO algorithm to search. When PSO algorithm falls into the local extreme, we introduce the GA algorithm to do the crossover and the mutation for the particles. According to the diversity of the particles, the algorithm escapes the local extreme. The steps are as follows.

The first step is to initialize the parameters of the particle group. It is the speed, location, individual extreme and the global extreme for each particle.

The second step is to input the particle and the sample. Then, we calculate the fitness of each particle.

The third step is to compare the fitness value of the current particle with the individual extreme and the global extreme. Then, we get the new individual extreme and the global extreme for the particle.

If

$$present < pbest_i, pbest = present, pbest_i = x_i. \quad (14)$$

Otherwise, *pbest<sub>i</sub>* is not changed.

If

$$present < gbest_i, gbest = present, gbest_i = x_i \quad (15)$$

Otherwise, *gbest<sub>i</sub>* is not changed.

*pbest<sub>i</sub>* is the individual extreme of the particle. *gbest<sub>i</sub>* is the global extreme of the particle. *present* is the fitness value of the current particle.

The fourth step is to update the speed and the location of the particle according to the formula (6) and the formula (7).

If

$$v_i(t+1) > v_{\max} \quad (16)$$

$$v_i(t+1) = v_{\max} \quad (17)$$

If

$$v_i(t+1) < -v_{\max} \quad (18)$$

$$v_i(t+1) = -v_{\max} \quad (19)$$

Otherwise, *v<sub>i</sub>(t+1)* is not changed.

If

$$x_i(t+1) > v_{\max} \quad (20)$$

$$x_i(t+1) = x_{\max} \quad (21)$$

If

$$x_i(t+1) < x_{\min} \quad (22)$$

$$x_i(t+1) = x_{\min} \quad (23)$$

Otherwise,  $x_i(t+1)$  is not changed.

The fifth step is to detect whether it falls into the local extreme. If it falls into the local extreme, we do the crossover and mutation operation. Then we jump to the step 2. Otherwise, we output the optimal solution.

### 3.2. The Path Selection Formula

For the ant  $k$  in the city  $i$ , the ant will select the city  $j$  according to the following formula.

$$\begin{cases} j = \arg \max_{j \in allow(i)} \{ \tau_{ij}(t) \eta_{ij}^\beta \}, q \leq q_o \\ p_{ij}^k = \frac{\tau_{ij}(t) \eta_{ij}^\beta}{\sum_{j \in allow(i)} \tau_{ij}(t) \eta_{ij}^\beta}, \quad else \end{cases} \quad (24)$$

Among them,  $\tau_{ij}(t)$  represents the pheromone in path  $(i, j)$  at  $t$  time. The heuristic information  $\eta_{ij} = 1/d_{ij}$  is used to guide the ants to move to the shorter city.  $allow(i)$  represents the set of the candidate cities when the salesman is in the city  $i$ .  $\beta$  is the weight coefficient. It is used to determine the degree of importance of the heuristic information.  $q$  is a random value that the range is  $[0,1)$ .  $q_o$  is the assumed threshold  $(0.8 \sim 0.9)$ . The formula can make the ants move to the city with the high probability. In the city, the multiply between the pheromone and the heuristic information is the biggest. It makes the ants move to the random cities with a little probability. This way not only ensures the direction of the ant, but also increases the diversity of the search.

### 3.3. The Updated Method of the Pheromone

At present, the updated methods of the pheromone that the ant colony algorithm uses include the updated method of the ant system pheromone and the updated method of the ant colony system *etc.* However, these methods generally exist the same questions. During the iteration, the current optimal path does not appear and the pheromone in the current path will increases in the updated strategy. It may lead to negative results. Firstly, the pheromone in current path may lead to the algorithm stagnate because of the excessive strengthening. Secondly, when the new path appears, the pheromone intensity in current path may be much lower than the original optimal path. This pheromone may not be alleviated after several iterations. In short, the current pheromone update method could not make the change of the current optimal path express in the pheromone distribution path. And this reduces the efficiency of the search algorithm.

According to analyzing the limitations of the pheromone update method, this paper proposed the pheromone update method which combines the local update with the global dynamic update.

### 1. The local update of pheromone

In the path construction process, when the ant goes an edge  $(i, j)$ , the algorithm will update the pheromone of this edge.

$$\tau_{ij}(t+1) = (1 - \varepsilon)\tau_{ij}(t) + \varepsilon\tau_0 \quad (25)$$

Among them,  $\varepsilon$  is the evaporation of the pheromone local updating.  $\tau_0$  is the initial value of the pheromone. When the ant goes through the edge  $(i, j)$ , the above formula will reduce the pheromone in this edge. It will reduce the probability that other ants select this edge. And it increases the opportunity to explore other paths.

### 2. The global dynamic update

After one iteration ends, the algorithm in this paper uses the following formula to update under the current optimal path.

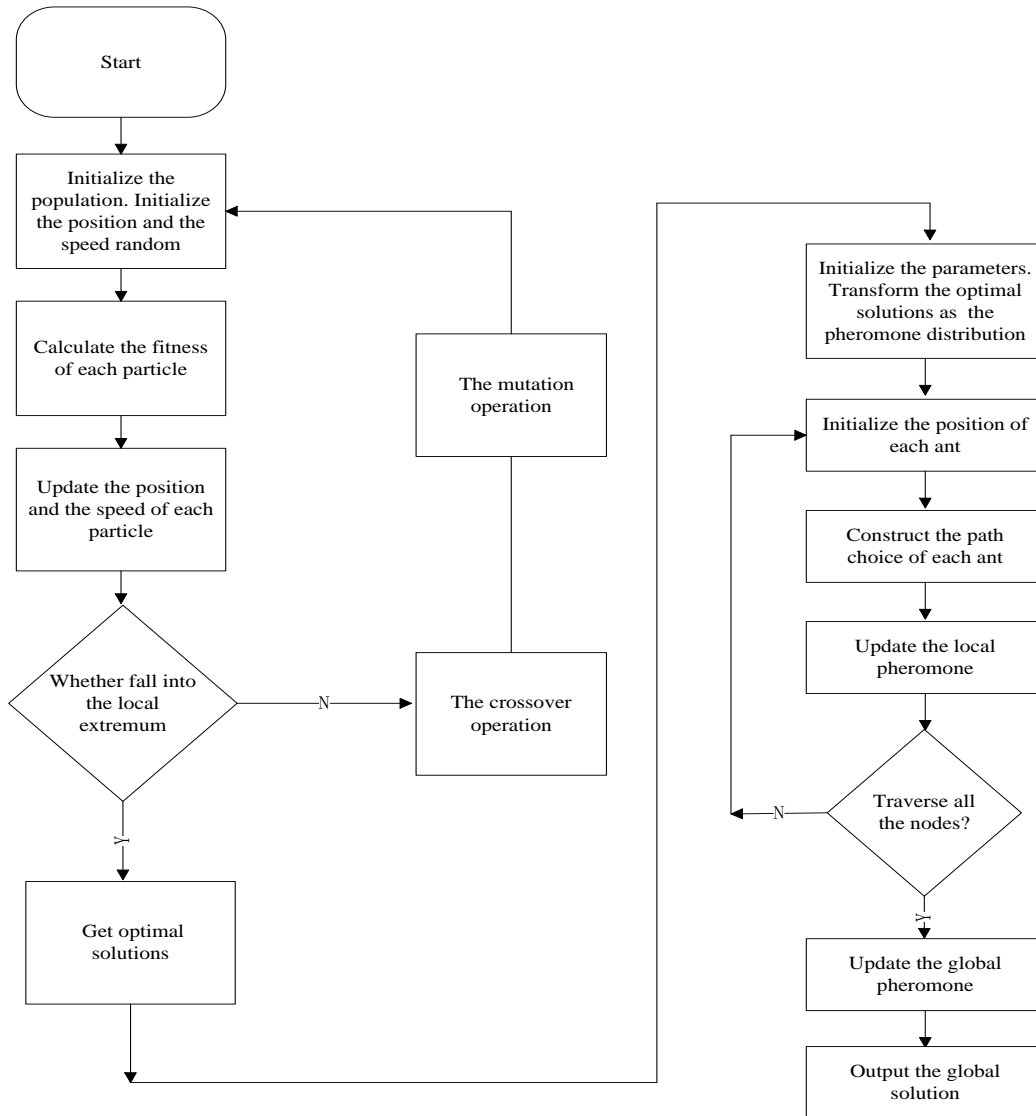
$$\tau_{ij}(t+1) = (1 - \rho)\tau_{ij}(t) + \rho\Delta\tau_{ij} \quad (26)$$

$$\Delta\tau_{ij} = \frac{L_1 - L_g}{L_g} \quad (27)$$

Among them,  $\rho$  is the evaporation of the global pheromone updating.  $\Delta\tau_{ij}$  is the increment of the global updating pheromone.  $L_1$  is the optimal path length of the current iteration.  $L_g$  is the optimal path length of the current path. Updating the current optimal path is to develop the optimal path. It makes the positive feedback of the pheromone in current path reserve to the next iteration until next optimal path appears.

The flow chart of the GAPSOAC algorithm is as follows.





**Figure 2. The Flow Chart of GAPSOAC Algorithm**

#### 4. Experiment

In the experiment, firstly, we study the performance of GAPSOAC algorithm. And we use GAPSOAC algorithm to solve the TSP question. Each TSP question operates 50 times. And the results are as follows.

**Table 1. The Algorithm Performance of GAPSOAC**

Question	Known optimal solution	Number of GAPSOAC optimal solution	Average search results of GAPSOAC
Berlin52	7542	40/50	7542.5
Ch130	6110	43/50	6110.4
D493	35002	45/50	35002.2
Eil76	538	46/50	538.2
KroA100	21282	48/50	21282.1

Then, we compare GAPSOAC algorithm with other algorithms. We select AS algorithm, GAAC algorithm and GAPSPAC algorithm to solve TSP question. The experimental results are as follows.

**Table 2. The Comparison of AS, GAAC and GAPSOAC**

Number	AS		GAAC		GAPSOAC	
1	21438		21137		20356	
2	21473		21154		20285	
3	21361		21174		20324	
4	21429		21203		20335	
5	21358		21152		20343	
6	21426		21169		20336	
7	21453		21176		20329	
8	21784		21213		20345	
9	21661		21147		20311	
10	21589		21138		20341	
Statistics results	Avg	Std. Dev	Avg	Std. Dev	Avg	Std. Dev
	21497	137.9225	21166	25.9831	20310	20.2004

From the above results, we can see that GAPSOAC algorithm has stronger ability of searching solution. The average results of 10 times are better than AS algorithm and GAAC algorithm. From the statistical standard deviation, GAPSPAC has better stability.

## 5. Conclusion

With the increasing complexity of the problem, the traditional ant colony algorithm shows its limitations of solving questions. Therefore, more and more scholars begin to study and improve the any colony algorithm. From the current point of the view, the improvement of the ant colony algorithm is from two aspects. On the one hand, it improves the ant colony algorithm itself. Firstly, it aims at the pheromone release to improve. Secondly, it aims at the probability selection method to improve. On the other hand, it combines with other intelligent optimization algorithms to improve.

Aiming at the short pheromone in initial and the lower efficiency, this paper combines the genetic algorithm and the ant colony algorithm. In addition, this paper improves the ant colony algorithm and proposes GAPSOAC algorithm. The major works in this paper are as follows. Firstly, this paper introduces the research status of the ant colony algorithm. Secondly, this paper introduces the basic steps of ant colony algorithm, genetic algorithm and PSO algorithm. Thirdly, aiming at the shortcomings, we combine the genetic algorithm and the ant colony algorithm and propose GAPSOAC algorithm. The last experiment verifies the algorithm has higher solution efficiency and the stability.

## References

- [1] M. Dorigo, V. Maniezzo and A. Colomi, "Ant system: Optimization by a colony of cooperating agents", *IEEE Transactions on Systems, Man, and Cybernetics-Part B*, vol. 26, no. 1, (1996), pp. 29-41.
- [2] A. Colomi, M. Dorigo, F. Maffioli, V. Maniezzo, G. Righini and M. Trubian, "Heuristics from nature for hard combinatorial optimization problems", *International Transactions in Operational Research*, vol. 3, no. 1, (1996), pp. 1-21.
- [3] M. Dorigo and L. M. Gambardella, "Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem", *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, (1997), pp. 53-66.
- [4] L. Santos and J. Coutinho-Rodrigues, "Current J R. An improved ant colony optimization based algorithm for the capacitated arc routing problem", *Transportation Research Part B: Methodological*, vol. 44, no. 2, (2010), pp. 246-266.

- [5] P. Zhang, J. Lin and L. Xue, "An Adaptive Heterogeneous Multiple Ant Colonies Algorithm", *Journal of Intelligent Systems*, vol. 194, (2010), pp. 301-314.
- [6] Y. Gajpal and Abad, "An ant colony system (ACS) for vehicle routing problem with simultaneous delivery and pickup", *Computers and Operations Research*, vol. 36, no. 12, (2009), pp. 3215-3223.
- [7] Y. Gajpal and P. L., "Abad Multi-ant colony system (MACS) for a vehicle routing problem with backhauls", *European Journal of Operational Research*, vol. 196, no. 1, (2009), pp. 102-117.
- [8] Y. Yang and M. S. Kamel, "An aggregated clustering approach using multi-ant colonies algorithms", *Pattern Recognition*, vol. 397, (2006), pp. 1278-1289.
- [9] U. Boryczk and J. Kozak, "Enhancing the effectiveness of Ant Colony Decision Tree algorithms by co-learning [J]", *Applied Soft Computing*, (2015), vol. 30, pp. 166-178.
- [10] M. Z. M. Kamali, N. Kumaresan and K. Ratnavelu, "Solving differential equations with ant colony programming", *Applied Mathematical Modelling*, vol. 39, Issues 10–11, (2015) June 1, pp. 3150-3163.
- [11] F.-C. Yang, Y.-C. Chou, "Superior/Inferior Segment-Discriminated Ant System for combinatorial optimization problems", *Computers & Industrial Engineering*, vol. 57, no. 2, (2009), pp. 475-495.
- [12] J. Yang and Y. Zhuang, "An improved ant colony optimization algorithm for solving a complex combinatorial optimization problem", *Applied Soft Computing*, vol. 10, no. 2, (2010), pp. 653-660.
- [13] A. Prakasam and N. Savarimuthu, "Metaheuristic Algorithms and Polynomial Turing Reductions: A Case Study Based on Ant Colony Optimization", *Procedia Computer Science*, vol. 46, (2015), pp. 388-395.
- [14] A. Kaveh and P. Sharafi, "Optimal priority functions for profile reduction using ant colony optimization [J]", *Finite Elements in Analysis and Design*, vol. 44, no. 3, (2008), pp. 131-138.
- [15] M. D. Toksari, "Minimizing the multimodal functions with Ant Colony Optimization approach", *Expert Systems with Applications*, vol. 36, no. 3, (2009), pp. 6030-6035.
- [16] Y.-y. LI, Q.-y. WEN and L.-x. LI, "Modified chaotic ant swarm to function optimization", *The Journal of China Universities of Posts and Telecommunications*, vol. 16, no. 1, (2009), pp. 58-63.
- [17] S. U. Seçkiner, Y. Eroğlu, M. Emrullah and T. Dereli, "Ant colony optimization for continuous functions by using novel pheromone updating", *Applied Mathematics and Computation*, vol. 219, no. 9, (2013), pp. 4163-4175.
- [18] C. García-Martínez and O. Cordón, F. Herrera, "A taxonomy and an empirical analysis of multiple objective ant colony optimization algorithms for the bi-criteria TSP", *European Journal of Operational Research*, vol. 180, no. 1, (2007), pp. 116-148.
- [19] A. Uğur and D. Aydin, "An interactive simulation and analysis software for solving TSP using Ant Colony Optimization algorithms", *Advances in Engineering Software*, vol. 40, no. 5, (2009), pp. 341-349.
- [20] J. Yang, X. Shi, M. Marchese and Y. Liang, "An ant colony optimization method for generalized TSP problem", *Progress in Natural Science*, vol. 18, no. 11, (2008), pp. 1417-1422.
- [21] A. Puris, R. Bello and F. Herrera, "Analysis of the efficacy of a Two-Stage methodology for ant colony optimization: Case of study with TSP and QAP", *Expert Systems with Applications*, vol. 37, (2010), pp. 5443-5453.

