An ant colony optimization algorithm based on the chaos search

Aifen Lu*†

†School of Information Engineering, Guangzhou Vocational and Technical University of Science and Technology, Guangzhou, Guangdong, 510550, China.
*Corresponding author’s e-mail: luaifen@gkd3.onexmail.com

Abstract. In this paper, through the analysis of colony bionic algorithm, the Ant Colony Algorithm is improved. Chaos search is added to the algorithm. Using the randomness of chaos search algorithm and the ergodicity of its local effective space, the efficiency and accuracy of Ant Colony Algorithm are improved and its convergence speed is accelerated. The simulation results show that the improved algorithm improves the stability, convergence speed and calculation accuracy, and effectively enhances the robustness of the Ant Colony Algorithm.

1. Introduction

There are all kinds of species in nature, and they all live in their way. Nature has evolved to be much better at choosing than humans. Human beings have also invented many tools and methods to help human beings by simulating the behavior and structure of organisms in nature. Since the computer came into being, the computing and simulation ability of human beings has been greatly improved. Naturally, there have also been algorithms that use the computer to simulate the survival and living ability of organisms to solve practical problems, such as the Fish Swarm Algorithm, Ant Colony Algorithm and Bee Colony Algorithm, Genetic Algorithm, etc. These algorithms solve complex optimization problems in daily life by simulating cooperative foraging of biological groups and finding the best food source at the fastest speed, which effectively improves the efficiency of mathematical optimization and solves the extreme value solving the problem of non-differentiable functions to some extent.

Chaos search algorithm[1] is a kind of random search optimization algorithm, which has attracted much attention in recent years. It performs random traversal in a solution space, and can traverse all states in the solution space according to its own operating rules. As an optimization algorithm with a strong ability of local optimization, it can be combined with other algorithms to find the optimal solution. As an optimization algorithm with strong local optimization ability, which can effectively avoid the optimization algorithm falling into the local optimal solution and unable to converge to the global optimal solution.

Ant Colony Algorithm[2] was proposed by Italian scholar Dorigo at the European Artificial Life Conference, and was used to solve the famous problem of travel agents successfully. This algorithm adopts the parallel computing mode, which is easy to combine with other algorithms, and has strong robustness[3]. However, it complicates the optimization problem, takes a long time to search, and has a slow convergence rate in the late stage, which is its biggest defects[4].
2. Chaos search algorithm
Chaos search is a kind of traversal search algorithm. For the disordered distribution, the algorithm can traverse all positions in a certain range one by one according to a certain search way, which is more suitable for solving the global optimal solution. However, because of its traversal nature, its running speed is slow, the convergence curve of the algorithm is gentle and the efficiency is low. Chaos search has three characteristics:
(1) Ergodicity; chaos search can experience the state of all positions one by one without repetition in a certain range.
(2) Randomness; the path of the chaotic algorithm follows the running rule and randomly accesses all nodes, so its initial value is very sensitive.
(3) Certainty; the motion trajectory of chaotic algorithm is generated by a definite iterative formula. Because of these characteristics, chaos algorithm is easy to implement and can effectively avoid falling into the local optimal solution, so it is often used as a local search algorithm. If as a global search algorithm, the convergence speed of chaos algorithm is slow and the global search ability is not strong, it is often used to combine with other algorithms for global search.

Given an n-dimensional optimization problem, \( x_i \) is the decision variable of the ith dimension, \( x_{\text{min},i} < x_i < x_{\text{max},i} \), then the specific steps of chaos search are as follows:
(a) Set \( k=0 \), and map the I dimension decision variable \( x_i^{(k)} \) to a chaotic variable \( c x_i^{(k)} \) using the following formula.
\[
 cx_i^{(k)} = \frac{x_i^{(k)} - x_{\text{min},i}}{x_{\text{max},i} - x_{\text{min},i}}
\]
(b) The next generation of chaotic variable \( c x_i^{(k)} \) is generated based on \( c x_i^{(k+1)} \) by chaotic mapping.
(c) Map the chaotic variable \( c x_i^{(k+1)} \) to the decision variable \( x_i^{(k+1)} \) by using the formula, and
\[
 x_i^{(k+1)} = x_{\text{min},i} + c x_i^{(k+1)} (x_{\text{max},i} - x_{\text{min},i})
\]
(d) Choose by evaluating the new decision variable \( x_i^{(k+1)} \)
(e) If a new decision variables \( x_i^{(k+1)} \) is superior to the \( x_i^{(k)} \), the output (\( x_1^{(k+1)}, x_2^{(k+1)}, \ldots, x_n^{(k+1)} \)) as the result of chaotic local search, otherwise, let \( k = k+1 \), return step (b).
Chaos mapping method is the more important formula in chaos algorithm. Logistic mapping is mostly used, and its iterative formula is as follows:
\[
 Z(k + 1) = \mu.Z(k).[1 - Z(k)]
\]
3. Ant colony algorithm
According to the research, although ants have no vision, they can find food sources as quickly as possible by working together in groups, and they can selectively find high-quality food sources. An individual ants release pheromone on the running path when they are searching for food. It judges whether they have ever walked this path through pheromones, which contain path related information, such as length, weight, etc. If it encounters a path without pheromones at the intersection, it will randomly choose a path to move forward and release pheromones. The longer the path is, the smaller the amount of information will be, which will pave the way for later foraging ants to choose the best. The more ants there are, the more pheromones there will be. And the path that has a lot of information, the more ants there are, the more pheromones there are, the more ants that choose that path, and ultimately that leads the colony to the optimal path. It can be seen that although individual ants have poor survival ability, the whole ant colony can be highly organized through pheromone release and selection, and the optimal path can be found through information exchange between each ant.
3.1. Initialization

Ant Colony Algorithm involves 6 parameters, the initial total number of ants m; Information heuristic factor $\alpha$; expected heuristic factor $\beta$; the pheromone volatile factor is $\rho$; Q is the total amount of pheromones; N is the number of iterations.

3.2. Ant Colony Algorithm solution space construction

The state transfer probability of the ant from the random starting point to food source J at time T is $\rho_{ij}^t$, then the calculation formula is:

$$
\rho_{ij}^t = \begin{cases} 
\frac{\tau_{ij}^\alpha(t)\eta_{ij}^\beta(t)}{\sum_{k \in allowed} \tau_{ij}^\alpha(t)\eta_{ij}^\beta(t)} & j \in allowed, \\
0 & \text{otherwise}
\end{cases}
$$

Where, $D(i,j)$ represents the distance between starting point i and food source point j; $\tau_{ij}$ indicated the pheromone between point I and food source point j at time T; $allowed_k = \{1,2,3...N\}$ stores the number of the food source, which indicates the next destination about the ant will reach; The $\eta_{ij}$ represents the degree of expectation from starting point i to food source point j.

3.3. Pheromone update

During the operation of Ant Colony Algorithm, the pheromone update is carried out according to the following formula:

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

Where, $\rho$ is the volatilization factor of pheromone, and $\rho \in (0,1)$, which indicates the degree of pheromone disappearance with time.

$\Delta\tau_{ij}(t) = \sum_{k=0}^{m} \Delta\tau_{ij}^k$; $\Delta\tau_{ij}^k$ refers to the amount of pheromones left by ant K from the starting point to the food source point during this foraging process. Its value is:

$$
\Delta\tau_{ij}^k = \begin{cases} 
\frac{Q}{L_k} & \text{If ant K passes the path from i to j}, \\
0 & \text{If ant K doesn't pass the path from i to j}
\end{cases}
$$

$L_k$ is the total length of ant K's path in this foraging process.

3.4. Termination conditions

When the number of iterations reaches the preset maximum number of iterations or the calculation precision meets the requirements, the operation is stopped and the optimal solution is output. Otherwise, the path record of the ant is updated and the execution is resumed.

4. Ant colony algorithm with chaotic search

Because Ant Colony Algorithm makes the problem complicated, the running time is too long and has poor robustness during operation[5], Therefore, this paper proposes an Ant Colony Algorithm based on chaotic search. Chaotic search strategy is introduced into the Ant Colony Algorithm. The specific steps are as follows:

1. The initialization of ant colony algorithm, including relevant information such as population number, information heuristic factor, expectation heuristic factor, pheromone volatilization factor, pheromone total amount and other related information.
2. Calculate the pheromone concentration on all paths.
3. The pheromone concentration on all paths was evaluated by comprehensive consideration (including expected heuristic factor, pheromone volatile factor, and information heuristic factor, etc.). If the resulting state after execution is better than the current state, the ant will move forward to the current result.
(4) Generate a random number Rand(). If the current pheromone concentration of Rand() is greater than the current pheromone concentration, the ant's search behavior will continue; otherwise, the individual will feedback and submit the current valid data.

(5) Chaotic search is added to the local search space

(6) Check whether the result meets the end condition of the algorithm. If the set number of iterations or accuracy requirements reach, the algorithm will be terminated and the results will be submitted. Otherwise, return to step 2 to continue.

The ant colony algorithm of chaotic search is added, and the ant colony algorithm is used to get the results in the initial local optimization and carry out local search. Because of the ergodicity of chaotic search, it can effectively avoid the algorithm falling into the local optimal. The improved algorithm adds a feedback mechanism, using a random number to decide whether to give feedback. If the current pheromone concentration is greater than the random number, the current valid data will be submitted, otherwise, the ant will continue to search.

By setting the feedback mechanism, the randomness of ant search can be increased and the precision of optimization results can be improved in the local search space. This feedback mechanism can not only effectively improve the global convergence of ant colony algorithm, but also improve the efficiency and accuracy of the algorithm. Its flow diagram is shown in Figure 1.

![Flow diagram](figure1.png)

**Figure 1.** The Flow diagram

5. Simulation experiment and analysis

In order to prove the effectiveness of the algorithm improvement, this paper tests the function, and compares it with the algorithm ZACO in reference[5] through test function, to test the convergence speed, the accuracy of the results and the operation efficiency of the algorithm. In this simulation experiment, five commonly used test functions are selected as shown in Table 1. All of the five functions are expressions with multiple local optimal solutions, which can effectively test the ability of the algorithm to jump out of the local peak and reach the global peak. Set the dimension of each test function as 100, run each function independently for 50 times, average the results, and finally calculate the average value and standard deviation of each function. The results are shown in Table 2:
Table 1. Test functions

| Function | Expression | Dimension | Search Space | Extremum |
|----------|------------|-----------|--------------|----------|
| Sphere   | $f_1 = \sum_{i=1}^{d} x_i^2$ | 100       | $(-100,100)^d$ | 0        |
| Rosenbrock | $f_2 = \sum_{i=1}^{d} \left[ 100(x_{i+1}) - x_i^2 \right] + (x_i - 1)^2$ | 100       | $(-100,100)^d$ | 0        |
| Griewank  | $f_3 = \frac{1}{4000} \sum_{i=1}^{d} x_i^2 - \prod_{i=1}^{d} \cos \left( \frac{x_i}{\sqrt{i/2}} \right) + 1$ | 100       | $(-100,100)^d$ | 0        |
| Rastrigin | $f_4 = \sum_{i=1}^{d} [x_i^2 - 10\cos(2\pi x_i) + 10]$ | 100       | $(-100,100)^d$ | 0        |
| Ackley   | $f_5 = -a \exp \left( -b \sqrt{\frac{1}{d} \sum_{i=1}^{d} x_i^2} \right) \prod_{i=1}^{d} \cos(c x_i) + a + \exp(1)$ | 100       | $(-100,100)^d$ | 0        |

Table 2. Test results

| Function | Ant Colony Algorithm | ZACO | Improved Algorithm |
|----------|----------------------|------|--------------------|
|          | Mean   | standard deviation | Mean       | standard deviation | Mean       | standard deviation |
| $f_1$    | 7.3269e-14 | 3.2571e-13      | 1.2213e-14 | 2.1725e-14       | 1.3216e-15 | 2.4973e-15         |
| $f_2$    | 2.3219  | 6.7425             | 15.3327    | 95.6518          | 97.3249    | 2.4217              |
| $f_3$    | 1.6326e-10 | 2.1574e-9      | 6.1246e-11  | 1.3521e-7        | 5.3417e-16 | 5.4423e-12         |
| $f_4$    | 0.4794  | 1.5191             | 3.3273e-4  | 0.0029           | 2.5374e-10 | 2.2396e-6          |
| $f_5$    | 9.6574e-9  | 1.2677e-8      | 1.3964e-12  | 9.1978e-13       | 5.6716e-14 | 4.4966e-14         |

From the above table, it can be seen that the three optimization algorithms can jump out of the local optimal solution and approach the global optimal result in five functions, the improved algorithm has the best performance and is closest to the extreme value of the function. However, compared with the improved algorithm proposed in this paper, the performance of Ant Colony Algorithm and ZACO algorithm in literature[5] is slightly worse. From the standard deviation point of view, the improved algorithm has the smallest standard deviation, which shows that the result is more accurate and the algorithm is more stable.

The convergence curves of the two functions are optimized by the three algorithms shown in Figure 2 and Figure 3. It can be seen from the figure that the improved algorithm has higher accuracy, faster convergence speed and higher efficiency compared with ant colony algorithm and ZACO algorithm.
6. Conclusion
Swarm artificial intelligence algorithm plays a very important role in solving the problem of finding the extreme value of non-differentiable function. It solves the problem that can not be solved by pure mathematics. But at present, most of the bionic swarm intelligence algorithms have some defects, such as Ant Colony Algorithm, which makes the optimization problem complicated, the optimization time is long, and the convergence speed is slow in the later stage. The ant colony algorithm with a chaos search algorithm can effectively improve the search speed and efficiency, as well as the accuracy of optimization and the robustness of the algorithm.

Reference
[1] Qi Rongbin, Feng Rupeng (2003) A new method for solving a Class of 0-1 Integer programming-Chaos Search Algorithm. Control and Decision, 06: 712-715.
[2] Colomi A, Dorigo M, ManiezzoV, et al. (1991) Distributed Optimization by Ant Colonies[C]// Proceedings of the 1st European Conference on Artificial Life. 91: 134-142.
[3] Information Technology-Industrial Informatics; Study Results from Xiamen University of Technology Provide New Insights into Industrial Informatics (Robust Scheduling of Hot Rolling Production By Local Search Enhanced Ant Colony Optimization Algorithm). Journal of Mathematics Week, 2020.
[4] Luo Z.H., Liu X.W. (2020) optimization of logistics distribution path based on Ant Colony Algorithm. Journal of Chongqing Technology and Business University (natural science edition), 37(04) 89-94. (In Chinese)

[5] Wenrui Zhang. (2017) Application of an Improved Ant Colony Algorithm in Coastal Tourism Route Optimization. Journal of Coastal Research, 208:98(SI).