

# A Comparative Study of Cuckoo Algorithm and Ant Colony Algorithm in Optimal Path Problems

Guanyu Wang<sup>1, a</sup>

<sup>1</sup>Internet of Things Engineering, Beijing University of Posts and Telecommunications, 100876, Beijing, China

**Abstract.** Finding the optimal path can be realized through a wide range of algorithms, which is demanded in many fields. Among countless algorithms that are used for solving the optimal path problem, the ant colony optimization (ACO) is one of the algorithms used to solve the approximate optimal path solution, while the cuckoo search (CS) algorithm is a swarm intelligence algorithm featuring Levy flight, whose core idea is derived from the cuckoo nesting property. In order to provide more ideas and directions for future research on optimal path problems, this paper discusses in detail the advantages and disadvantages of the two algorithms for solving the optimal path problem and their scopes of application by comparing principles and flows of the two algorithms.

## 1 Introduction

### 1.1. Principles and Difficulties of the Optimal Path Problem

The path is the path in the undirected graph satisfying the condition that all the vertices (except for the starting point and the ending point) and all the edges are different. The optimal path is the one that best meets certain requirements in the path. For example, a path will be called the shortest path if the sum of the weights of its constituent edges is minimized. The optimal path algorithm refers to the calculation method for finding the optimal path of a graph. Many scholars both at home and abroad have done numerous research on how to solve the optimal path problem, such as the classic Dijkstra algorithm, A\* algorithm, ant colony optimization, and simulated annealing algorithm. These classical algorithms are studied from different angles, and are continuously supplemented and improved to obtain an efficient optimal path algorithm. This paper focuses on the comparison between the cuckoo search and the ant colony optimization in the optimal path problem.

### 1.2. Common Algorithms for Solving the Optimal Path Problem

#### 1.2.1. Dijkstra's algorithm

Dijkstra's algorithm is an algorithm for finding the shortest paths between nodes in a graph, whose core idea can be described in this way: Starting from the path, by constructing the node queue, the path node spreads out from each possible path until the end. As for single-source shortest paths which are directed and

weighted, the algorithm uses breadth-first search to obtain the nodes of the shortest path.

#### 1.2.2. A\* Algorithm

Because of its flexibility, the A\* algorithm has become one of the popular algorithms for searching the shortest path, and can be used in many different situations. Like other graph search algorithms, A\* algorithm searches a large area of the graph. A\* can not only be used to search for the shortest path like Dijkstra, but also guide itself with heuristic functions like BFS.

#### 1.2.3. Ant Colony Optimization Algorithm

The ant colony optimization algorithm (ACO) is inspired by the group behavior of ants searching for food, during which the ant colony gradually tends to the shortest path. Marco Dorigo discovered this feature of ant colonies in nature and applied this inspiration to the calculation of the shortest path. In 1992, he proposed the ant colony optimization algorithm in his doctoral thesis. ACO belongs to a probabilistic algorithm, which is essentially a heuristic global optimization algorithm featuring distributed computing, positive feedback of information and heuristic search.

#### 1.2.4. Particle Swarm Optimization

Particle swarm optimization (PSO) is very popular for solving practical problems because of its easy implementation, highly precise solution and fast convergence. PSO belongs to an evolutionary algorithm, whose principle lies in finding the optimal solution by iteration, and evaluating the solution quality by the fitness of the optimal solution.

## 2 Cuckoo Search (CS)

### 2.1. Principles of Cuckoo Search

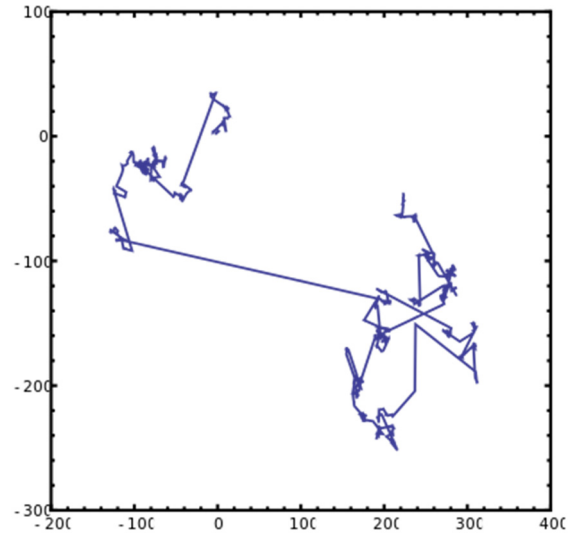
The cuckoo search algorithm is an emerging metaheuristic algorithm developed by Xin-she Yang and Suash Deb of Cambridge University in 2009, which is based on simulating the nest parasitism of cuckoos in nature and also combined with the characteristics of Levy flight to solve the optimization problem.

#### 2.1.1. Brood parasitism

According to long-term observations by biologists, cuckoos breed young birds in a parasitic manner. Brood parasitism refers to a special reproductive behavior in which some birds have eggs in other birds' nests and rely on others to hatch and raise their offspring, which can be summarized in three characteristics. First, cuckoos usually lay their own eggs in host nests in advance of the hosts' hatching period, each cuckoo laying one egg at a time. Second, cuckoos push the existing eggs out of host nests or remove them when laying eggs there. Third, as soon as they hatch, cuckoo chicks will also have the habit of pushing other host chicks of the same nest out so as to enjoy exclusive fostering of host birds due to their nature.

#### 2.1.2. Levy Flight

Apart from cuckoos' unique approach to hatching, Levy flight also plays an important part in cuckoo search algorithm. Numerous studies have shown that many animals and insects demonstrate regular characteristics of Levy flight in their flight paths. The discovery of Levy flight was made in 1996 by Viswanathan and his colleagues using satellite positioning systems to study the albatrosses' foraging behavior. They found that the albatrosses' flight intervals obeyed the power-law distribution and they proved this finding through consistent spatial distribution of the food quantity on ocean surface. In addition, a recent study by Reynolds and Frye showed that fruit flies would fly at right angles from time to time, which also demonstrated the intermittent scale-free searching feature of Levy flight.



**Figure 1** Levy flight path [1]

By abstracting the cuckoo nesting method into a theory, a new algorithm is derived. Cuckoo search is based on three idealized rules:

- 1) Each cuckoo lays one egg at a time, and dumps its egg in a randomly chosen nest;
- 2) The best nests with high quality of eggs will carry over to the next generation;
- 3) The number of available host nests is fixed, and the egg laid by a cuckoo is discovered by the host bird with a probability  $P_a$ . On discovering parasitic eggs, host birds either push them out or give up existing nests and build a new one. To simplify the model, let this idealized rule approximate to the probability  $P_a$  of the host birds building new nests.

The above rules can be summarized as follows:

- 1) Each egg in a host nest represents a solution;
- 2) Each cuckoo represents a new solution, whose purpose is to iteratively replace inferior solutions in the original scheme with a better one;
- 3) There is only one egg in each nest.

Based on the above three idealized rules, the formula for updating the path and position of cuckoo nesting is as follows:

$$x_i(t+1) = X_i(T) + \alpha \oplus L(\lambda) \quad (i = 1, 2, \dots, n)$$

The steps are as follows:

Step 1: The objective function  $f(x)$   $x = (x_1 \dots x_d)^T$  is established to initialize the group. Then the initial positions  $X_i (i = 1, 2, \dots, n)$  of  $n$  nests are randomly generated, and the algorithm parameters are set;

Step 2: The objective function value of each nest is calculated and the current optimal solution is recorded;

Step 3: The last optimal solution is kept and the positions of other nest are updated according to the formula for position updating;

Step 4: The position of the current nest is compared with that of the previous nest. If the former solution is better, it will be recorded as the current optimal solution;

Step 5: A random number  $R$  as the probability of the host bird finding parasitic eggs is selected compared with

Pa. If  $R > Pa$ , the position of the nest will be randomly changed to obtain a new set of nest positions;

Step 6: If the condition for ending is not met, the operation will return to Step 2;

Step 7: The globally optimal position is output.

## 2.2. Problems and Drawbacks of Cuckoo Search

Cuckoo search is a new meta-heuristic algorithm for swarm intelligence optimization, which is established by combining the brood parasitism of cuckoos and the model simulation of Levy flight. By continuously iterating the optimal solution until the globally optimal solution is obtained, this algorithm provides an idea for finding the optimal path in some manner, namely, continuously iterating the optimal path until the globally optimal path is obtained. Therefore, compared with other optimal path algorithms, cuckoo search is more advantageous in some cases. However, the algorithm is still undergoing rapid development and improvement, and the processing of some links still needs continuous optimization.

### 2.2.1. Convergence

Cuckoo search has been proved to be effective in practical applications, whose convergence is mainly based on the establishment of the Markov chain model through analyzing the homogeneity of the chain to prove that it satisfies the two conditions of the global convergence in a random search algorithm. However, the internal implementation of the cuckoo search is not comprehensive enough and still needs detailed study and analysis, which calls for further understanding of the convergence speed and quality of the algorithm. It is believed that the convergence of cuckoo search will be more thoroughly understood, which also plays a profound role in better applying this algorithm.

### 2.2.2. Limitations

The original cuckoo search can only be used to solve continuous problems. As for discrete problems and multi-objective problems, however, it fails to perform well. Therefore, the algorithm has certain limitations when processing some other problems, which is exactly what we need to further study and improve. As for the continuous problems that the algorithm can deal with, there has been much advancement in terms of step size, parameter, intercoupling with other algorithms and other factors used to improve the performance of related indicators. Meanwhile, this algorithm also has problems in adaptability and obtaining ideal search results, and its ability to solve complex problems is limited. Future research should be directed towards studying how to explore new methods and strategies to improve the performance of high coupling functions between variables.

### 2.2.3. Balance Between global Search and Local Search

The two indicators of local search ability and global convergence ability are common standards for measuring intelligent optimization algorithms. The so-called local search refers to the ability of the algorithm to gradually approach the optimal solution, while the global convergence ability is the ability to decide the approximate range of the global optimal solution. [2] These two metrics play a significant role in evaluating optimization algorithms, which determine the merits of an algorithm from two different aspects.

The significance of local search lies in that it offers an idea to solve complex problems, that is, to exchange time for precision. At present, NP-complete problems still accounts for a large part in optimal solutions. The time required for the optimal solution to this problem exponentially increases over time, which is very troublesome. Therefore, local search can be used to find the most approximate (second-best) solution within a reasonable time span, achieving the balance between time and solution.

How to balance the local search and global convergence in cuckoo search is also one of the directions to be further studied, so it is necessary to understand the three existing improved algorithms of the local search algorithm so as to further understand the internal principles and improving directions of local search. The improvement of local search from three aspects is described as follows:

#### a) Local Optimum

The local optimum refers to an algorithm falling into the local extremum point in the continuous iterative search process and then ending the operation. This is similar to climbing the summit of mountains which has many small peaks. Due to the lack of knowledge about the mountains, a small peak reached by the algorithm might be wrongly judged to be the highest. However, this local peak is likely to much lower than the summit of the mountains. Therefore, regarding such a local extremum point as the optimal solution is unwelcome.

#### b) Step Size

The step size also affects whether the final solution of the problem can be the optimal one. For some problems, the mapping function  $N$  value that can obtain the optimal solution must be some fixed values. Therefore, if the selection range of the step size does not cover those specific values, chances are that the optimal solution cannot be obtained. The root cause of this situation lies in that the step size is fixed and can only search a few fixed points, which can be solved by changing the step size during the search.

#### c) Initial Point

The selection of the initial point is, at heart, similar to the step size problem, because different step sizes can also be understood as different initial points. When the optimal solution can't be obtained due to the initial point, it is necessary to randomly select the initial point, and finally obtain the optimal solution by comparison, thus avoiding falling into the local optimal solution caused by the fixed initial point.

The current research shows that although cuckoo search is a new type of optimization algorithm, it can be optimized by following the three improving directions mentioned above. The convergence and applying limitations of cuckoo search are still problems to be solved. By studying ant colony optimization (ACO) and making a horizontal comparison between CS and ACO in the following, this paper provides further understanding of advantages and disadvantages and scopes of application of these two algorithms, and helps learn how to analyze different algorithms from various dimensions.

### 3 Ant Colony Optimization (Aco)

#### 3.1. Principles of Ant Colony Optimization (ACO)

The ant colony algorithm was first proposed in 1991 when Marco Dorigo and his colleagues researched into new algorithms. They found that ant colonies in nature secreted pheromones to exchange information when foraging so as to quickly find the target, inspired by which, they put forward the ant colony optimization algorithm based on positive feedback of information.

The basic idea of the ACO is derived from the shortest foraging path gradually formed by the ant colonies in nature, which is benefited from the biological characteristics of the ants. Although ants are visually undeveloped, they can keep secreting a biological hormone called pheromone when foraging. The pheromones secreted by ants are left in paths they pass. Such paths are random, but because the ants that first obtain food will leave first, so constant accumulation will increasingly raise the concentration of pheromones on the shortest path, which attracts the following ants to choose this path, resulting in more and more pheromone on this path. Higher concentration of pheromones in turn attracts more ants, thus an optimal path will be chosen by ant colonies in the long run. This selection process is called the autocatalytic behavior of ants. While individual ants just randomly select their foraging path, an ant colony will gradually form the optimal path due to the secretion of pheromones, objectively reaching the effect of finding the optimal solution. This phenomenon is called swarm intelligence.

Problems that are properly solved through swarm intelligence algorithms are numerous. The TSP problem is classical case, in which requires the travelling salesman to find what the shortest possible route that visits each city and returns to the origin city is under the condition that each city can only be visited once. Due to its similarity to the ant colony foraging, ACO is suitable for modeling and solving this kind of problem. The solving steps of the ACO are as follows:

Step 1: Parameters are initialized. The ant colony optimization algorithm involves many parameters are more, so it's necessary to initialize the problem scale, pheromone factor, pheromone volatilization factor, pheromone constants, heuristic function factor and the maximum number of iterations and then preprocess the data information.

Step 2: The initial point of the ant is randomly positioned and the set point the each ant passes is calculated until all the set points are passed.

Step 3: The path length of each ant is calculated, the current optimal solution is recorded, and the pheromone concentration on the path is updated. [3]

Step 4: Steps from Step 2 to this step are repeated until the maximum number of hits is reached.

Step 5: The result is output.

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

Calculation formula

There are many parameters in the formula of ACO, so difference in the selection of parameters will lead to difference in the overall calculation process. Selecting different initial values will also randomly affect the optimization process. As mentioned above, improper selection of the initial value will also cause a local optimum. Because the final aim is to convert the local optimum into a globally optimal solution, the setting of parameters is particularly important, the key of which lies in balancing the relation between local and global optimum. This is also the key issue gradually studied and summarized at present. Nevertheless, the optimization formula and steps described above are followed to solve problems relating to ACO. The solution, namely the local or global optimum, can be finally obtained by iterating according to the formula.

#### 3.2. Problems and Drawbacks of ACO

The ant colony optimization algorithm is fairly popular among path optimization problems because of its strong robustness, parallelism and positive feedback. However, no matter how excellent the algorithm is, it still has limitations. In the following, some limitations of ACO are analyzed.

1) Ant colony algorithm is not efficient for dealing with large-scale combinatorial problems. Because the time complexity of ACO is  $O(n \cdot (n-1) \cdot m \cdot T/2)$ , it takes longer time to process large-scale problems.

2) Because there are many parameters involved in ACO, improper selection of initial values may lead to local optimum, stagnation around the range of the local optimum, immature converge prematurely, eventually resulting in the convergence of the solution of the entire ant colony and the failure to obtain the optimal solution.

3) The ant colony algorithm cannot properly handle continuation problems.

4) Like the situation described in the limitation of time complexity, when the problem size is too large, individual ants will move randomly, which takes longer time to operate the whole algorithm, lowering the efficiency of finding the optimum.

5) The selection of pheromone-related parameters in the formula is largely empirical, so if the pheromone

parameters are selected without experience and theoretical arguments, chances are that the results obtained are greatly dispersed due to the inefficiency and poor convergence of the algorithm.

## 4 Comparative Analysis of Cuckoo Algorithm and Ant Colony Algorithm in Optimal Path Problems

### 4.1. Metrics that Measure Algorithm Performance

It is widely acknowledged that the most important indicators for measuring algorithms are time complexity and space complexity. The so-called time complexity refers to the amount of time required by the algorithm, and the space complexity refers to the size of memory space needed by the algorithm. Calculation of these two complexities is similar, usually represented by the asymptotic behavior of complexity. In addition to time complexity and space complexity, there are other metrics, including correctness, readability, and robustness. The so-called correctness regulates that the algorithm must be able to reproduce the same problem stably. At the same time, the readability of the code also demonstrates quality of an algorithm because if the logic of an algorithm is difficult to understand, it is unfriendly for others to use and improve in the future. Finally, the robustness of an algorithm is also an important measuring index. An excellent algorithm should be equipped with corresponding logic for any abnormal situation, thus avoiding easy termination or abnormal situations.

Therefore, it can be seen from the metrics mentioned above that the ACO has disadvantages of slow convergence and easily falling into local optimum. Because the ACO involves too many parameters, it easily leads to the lack of initial pheromone, and high complexity. Moreover, the algorithm is prone to stagnate and falls into local optimum of the algorithm, which is not conducive to finding the global optimal solution. Although it can be calculated for multiple times through random initial values, it is time-consuming to do so. Therefore, how to find the balance point is also a problem that needs to be dealt with.

In contrast, cuckoo search is a new meta-heuristic search algorithm enjoying extensive research prospects. [4] By measuring it with the above performance metrics, it can be conclude that the cuckoo search has the advantages in terms of great robustness on the whole, readable logics, portability and platform independence. The algorithm also demonstrates advantages in strong global search ability, few selected parameters, excellent search path, and strong ability of solving multi-objective problems.

### 4.2. Comparative Analysis and Conclusions of the Two Algorithms

As a path optimization algorithm for finding approximate

solutions, ACO is not quite suitable for dealing with problems requiring precision but usually applicable to the problems whose accurate solutions can't be obtained within time demanded. In other words, ACO can be adopted to obtain the second-best solutions of NP problems, such as the classic TSP problem. The TSP problem can also be said to be the shortest path problem, but the problem is to find the minimum Hamiltonian path of a complete graph. Therefore, this kind of problem belongs to the NPC problems. In other words, when the scale of the problem has a great impact on the time of processing the problem, the time cost of using the existing algorithms for finding the exact solution for the large-scale TSP problem is enormous. Therefore, similar algorithms like the ACO are usually adopted to find the second-best solution of such problems, trying to achieve a balance between time consumed and the precision of the solution.

While as a new type of optimization algorithm, cuckoo search solve the continuance problems which cannot be handled by ACO. Meanwhile, since the cuckoo search algorithm requires fewer parameters in the formula of, it is less affected by other factors and more robust. In the following, a comparative analysis of specific performances of these two algorithms is made.

#### 4.2.1. Analysis of the Ant Colony Optimization Algorithm

In a sense, the ACO can be considered as a self-organizing type in system theory. The so-called self-organization means that organization instructions come from the system. [5] Abstractly, self-organization is the process of increasing the entropy of the system without external influences (that is, the process of the system changing from being disorderly to orderly), which is exactly the case for ACO. In the initial stage of the algorithm, individual ants search for the solution disorderly, and the algorithm goes through a series of optimization calculation. Then those individual ants spontaneously tend to find the second-best solutions individual through pheromones, shifting from being disorderly to orderly. Therefore, the ACO is self-organizing.

Meanwhile, the ant colony optimization is also a parallel algorithm. The foraging process of each ant is independent of each other, and only communicating with others through the pheromones left by them. Therefore, practically, a distributed system can be established to ensure independent process of finding the optimal solution, which not only enhances the robustness of the algorithm, but also improve its global search ability.

Another obvious feature of ACO is its positive feedback. The positive feedback lies in that individual ants mainly rely on the pheromones generated by other ants to communicate with each other in the process of foraging. A large number of individual ants gradually accumulate pheromones, and finally find the optimal path. In this process, it can be seen that the essence lies in the process of positive feedback. The pheromone concentration in different paths is different because each

ant leaves pheromones in its own search path, and the positive feedback reflects the phenomenon that more pheromones are left in the path of better solutions, which directs many other ants to find this path. The local or global optimum which has the most pheromones can be found by repeating this. The process of positive feedback is also an important feature of ACO, which is the basis of its operation.

#### 4.2.2. Analysis of Cuckoo Search

As a relatively new type of path finding algorithm, the cuckoo search demonstrates obvious advantages, including its excellent global search ability. Due to the characteristics of Levi flight, cuckoo search can better avoid falling into the local optimum. Meanwhile, compared with ACO, cuckoo search uses fewer parameters, which contributes to its excellent robustness. Also, cuckoo search shows good applicability when dealing with multi-objective problems. What's more, the Levi flight demonstrating in cuckoo search is an efficient global random search algorithm and has been proven to perform better than other algorithms in optimal problems. The rise of the cuckoo algorithm over the past few years has confirmed that based on the advantages mentioned above, it can be said to be a new star among swarm intelligence algorithms.

#### 4.2.3. Comparison of Main Advantages and Disadvantages between Cuckoo Search and Ant Colony Optimization

|                      | Cuckoo Search  | Ant Colony Optimization  |
|----------------------|--|--|
| Advantages           | The algorithm depends on fewer parameters and is robust;<br>It has the trait of Levi flight, which enhances its global search ability;<br>It's easy to couple with other algorithms, demonstrating strong versatility. | It has a positive feedback mechanism and is effective in dealing with distributed problems;<br>It's easy to combine with other algorithms, featuring great robustness and reliability. |
| Disadvantages        | Convergence rate is affected by Levi flight and may be slightly slower.  | Initial information of the algorithm is scarce and it is easy to fall into local optimum;<br>It usually takes longer time to search, not suitable for large-scale problems.            |
| Scope of application | Continuous optimization  | Discrete optimization;<br>Small-scale problems   |

## 5 Conclusion

Cuckoo search algorithm is a new type of meta-heuristic swarm intelligent algorithm for path searching, which has a good research potential for continuous path finding problems. At the same time, ant colony optimization is also a very good algorithm for dealing with discrete NP problems, demonstrating great research value. Both of the two algorithms show good robustness when solving problems in different fields. In addition, the powerful features such as portability and platform independence of the cuckoo algorithm are also of profound significance.

The cuckoo search displays the feature of Levi flight and has strong global search ability. It is suitable for solving continuous problems and multi-objective problems. However, the cuckoo search algorithm also has its limitations, such as slower convergence, and future research should be focused on improving its shortcomings. By analyzing the principles, advantages and disadvantages of cuckoo search and ant colony optimization, this comparative study inspires future studies.

Ant colony optimization is a classical algorithm for finding the approximately optimal solutions. Although it has disadvantages of slower convergence and easily falling into the second-best solution because of large number of parameters, it has a positive feedback mechanism, strong global search ability, strong robustness, high reliability, and can be easily combined with other algorithms. Therefore it is still a good meta-algorithm researched and improved by scholars. Studying how to improve the above two algorithms, or maybe combine advantages of other algorithms into the two algorithms is also the direction and goal of the author's future research plan. Hopefully, the discussion of cuckoo search and ant colony optimization algorithms for solving optimal path problems can help future research. Finally, the author welcomes any suggestion for improving or correcting this paper.

## References

1. Lan Shaofeng & Liu Sheng, Overview of Research on Cuckoo Search Algorithm [J]. Computer Engineering and Design, 2015(04).
2. Fan Shuaijun, Application Research and Improvement of Cuckoo Search Algorithm [J]. Southwest Jiaotong University, 2016.
3. Cheng Manzhong & Wang Jiangqing, Research on Ant Algorithm to Vehicle Routing Problem [J]. Computer & Digital Engineering, 2007(05) P146-P148.
4. Yao Yuanyuan & Ye Chunming, Solving Job-shop Scheduling Problem by Cuckoo Search Algorithm [J]. Computer Engineering and Applications, 2015(05) P255-P260, 265.
5. Liu Jianfang, Research and Application of Ant Colony Algorithm [J]. Chongqing University, 2015.