Research on Logistics Distribution Path Planning Based on Fish Swarm Algorithm

Chuanyu Xiong*, Youhua Xu

Computer and Information Engineering College, Guangdong Songshan Polytechnic College, Shaoguan City, Guangdong, 512126, China
Email: 25505@gdsspt.edu.cn

Abstract: Logistics distribution related to the transportation cost and the competitiveness of logistics enterprises. How to effectively determine the shortest path of logistics distribution has become a key link for logistics enterprises to reduce costs, improve efficiency and enhance competitiveness. This paper proposes a method of using Fish Swarm Algorithm to optimize the logistics distribution path. Experiments show that the method is effective. The algorithm can quickly jump out of the local optimal solution and converge on the shortest path. The optimized path length is less than half of the initial length.

1. Introduction
In recent years, with the development of China’s economy, e-commerce has been vigorously developed. The development of e-commerce has brought great power to logistics transportation. Logistics transportation involves transportation vehicles, transportation costs, transportation path planning and so on. These problems are related to the competitiveness and profits of transportation enterprises. How to reduce the transportation cost and improve the transportation efficiency has become a hot issue for many scholars. The terminal of logistics distribution is the nearest gathering place where the end customer is located.

Reasonable arrangement of the transportation sequence (path) involves the length of the total path and ultimately the transportation cost. Therefore, how to plan the transportation route has become an important link in the distribution process of various logistics companies.

Traditional transportation route planning is done manually, it not only takes a lot of time, but also is difficult to find the optimal path planning. Scholars try to use intelligent algorithms to complete the path planning of logistics distribution, and many artificial intelligence calculations are applied to the problem. Genetic algorithm[1], particle swarm optimization algorithm[2], bee colony algorithm[3], ant colony algorithm[4] have all been well applied and achieved certain results.

2. Path planning problem model
The problem of single-vehicle path planning in logistics distribution is essentially a traveling salesman problem, and multi-vehicle path planning is a multi-traveling salesman problem. Assuming that there are 30 logistics sorting centres in a certain area, the logistics enterprises have to send these commodities from the warehouse of the distribution centre to the other 29 cities. The vehicles depart from the distribution centre and arrive at each sorting centre. After unloading the goods, proceed to the next city until the goods are delivered and returned to the starting point. Since 30 cities are randomly distributed, how to reduce the mileage of vehicles is the ultimate goal of path planning.
2.1 Modeling
Before establishing the path planning model, we need to assume the following problems:

1. All goods depart from the distribution centre warehouse, which is randomly selected in this implementation.
2. Cities are randomly generated, but once they are generated, they will not change. The geographic locations are relatively fixed, that is, the distance between cities is fixed.
3. During the driving process, the size of the path is considered only, and other issues such as fuel consumption, manpower, environment, and vehicle conditions are not considered.

2.2 Programming of objective function
Assuming there are 30 cities in the target, Table 1 shows the meaning of each character:

| Symbol | Express meaning |
|--------|-----------------|
| \( i,j \) | I or J City \((I, j = 1,2,\ldots,30)\) |
| \( W_{ij} \) | Represents the distance from city \( i \) to city \( j \) |
| \( x_{ij} \) | Indicates whether the vehicle is moving from \( i \) to \( j \), if so \( x_{ij} = 1 \) otherwise \( x_{ij} = 0 \) |

Objective function value \( C = \min \sum_{i,j=0}^{30} w_{ij}x_{ij} \), the constraints between each member are shown in the following formula, formula (1) indicates that the distance between any two cities can only be in three cases, and formula (2) (3) indicates that there is no overlap between any two cities, that is to say there is a certain distance between them. Equation (4) means that there is no loop in the calculation process. Equation (5) means that the distance between any two points is either 0 or 1. The distance is 0 only when \( i=j \). There is only one city.

\[
\left\{ \begin{array}{l}
\sum_{j=0}^{30} x_{ji} - \sum_{i=0}^{30} x_{ij} = \begin{cases} 
-1, & i = 0 \\
1, & i = 30 \\
0, & i \neq 0, 30 
\end{cases} \\
\sum_{j=0}^{30} x_{ij} \geq 1, i = 0,1,\ldots,29 \\
\sum_{i=0}^{30} x_{ij} \geq 1, i = 0,1,\ldots,30 \\
x_{ij} + x_{ji} \leq 1, \ i, j = 0,1,\ldots,30 \\
x_{ij} = 0 \text{ or } 1, \ i, j = 0,1,\ldots,30
\end{array} \right.
\]

3. Artificial Fish Swarm Algorithm
Artificial Fish Swarm Algorithm\(^5\) is a new algorithm proposed at 2002 by Professor Li Xiaolei of Shandong University. The algorithm adopts a top-down new optimization model with few method parameters and is relatively simple to implement. The set parameters are not sensitive and can converge faster in the optimization process.

A single fish is not very intelligent, it does not have complex comprehensive reasoning ability and logical judgment ability, but the fish living in groups can quickly find the place with the most abundant food and show certain high-intelligence behaviours. The artificial fish swarm algorithm divides the behaviour of fish into four types: foraging behaviour, swarming behaviour, tail-catch behaviour, and random behaviour. Gather near the optimal solution to complete the overall optimization process. The mathematical description of the four behaviours of fish as follows:
(1) Foraging Behavior
Suppose that fish A is in the current state \( x_a \), and then randomly select another state \( x_b \). At this time, the size of \( f(x_a) \) and \( f(x_b) \) is calculated by the fitness function. If \( f(x_a) < f(x_b) \), then this fish moves in the direction of \( x_b \), otherwise do not move, find another state to determine whether to move to a new state. If the maximum number of times is exceeded, and has not found a better state, the random behavior of fish will be executed, its mathematical expression for:
\[
X_a = (X_a + step \cdot rand \cdot \frac{x_a - x_b}{\| x_a - x_b \|}), \text{if } f(x_a) < f(x_b) \text{ and } a < limit \text{ else Random Behavior}
\]
Rand is a random function.

(2) Swarming Behavior
In order to get more food and enough sense of security, fish will choose to live in groups, but which group to the fish will move to depends on the function value of the center of the fish group. In addition, individual fish moving to the group must avoid crowding. Causes damage, so it must calculate the function value of the center of the fish group and the function value of its current position. Assuming that the position of the center of the fish group is \( x_c \), the crowding factor \( \sigma \) of the fish group, if \( f(x_a) < f(x_c) \) and the crowding degree is less than \( \sigma \), the fish will choose to move to the fish group, otherwise they will perform random behavior. \( X_{i+1} = X_i + step \cdot Rand \cdot \frac{x_c - x_i}{\| x_c - x_i \|} \), \( X_i \) represents the state of the fish at time t.

(3) Rear End Behavior
Also known as following behavior, it refers to the state of the fish searching for the best fish in its field of view in the current state. If \( f(x_a) < f(x_j) \) in the current state, and When the crowding degree around the fish in the optimal state is not high, move to j fish, otherwise perform foraging behavior.
\[
x_a = \begin{cases} 
  x_a + step \cdot rand \cdot \frac{x_a - x_j}{\| x_j - x_a \|}, & \text{if } f(x_a) < f(x_j) \text{ and } a < limit \\
  \text{else Foraging Behavior} 
\end{cases}
\]

(4) Random behavior
Random behavior is the default behavior of a single fish. When a fish does not join a group of fish, it does not have a certain direction for foraging, which is a random walk. When food is found in the process of random swimming, it will quickly approach the food and then perform other behaviors. \( X_{i+1} = X_i + Visual \cdot Rand \), Visual refers to the visual field of view of the fish.

4. Apply Fish Swarm Algorithm to path planning
When the fish school algorithm is applied to path planning, it needs to be adjusted according to the state of the fish. The objective function value is recorded on the bulletin board. Every time the state of the fish changes or after foraging, swarming, rear-end collision, etc., it needs to recalculate the objective function value and update the content of the bulletin board. The flow diagram of path planning based on the Fish Swarm Algorithm is shown in Figure 1:
5. Simulation results

Simulation design. In the fish school algorithm, each fish in the fish group represents a path in the path planning, that is, fish \( x_{ij} \) represents the path from city \( i \) to city \( j \), fish group represents all possible paths, fish fitness value represents the total length of the complete path, and the final optimization result refers to the optimal solution obtained after optimization by the fish school algorithm.

First, initialize the city node and generate it randomly. The distance between every two cities is mathematically expressed as the distance between two points, which is calculated by the distance formula. Set the iteration number \( \text{iter} = 1 \), and set the maximum iteration number \( \text{itermax} \), the size of the fish group is \( N \), and the field of view is visual. Then calculate the distance between the cities

\[
D_{ij} = \sqrt{(\text{city}_{i,1} - \text{city}_{j,1})^2 + (\text{city}_{i,2} - \text{city}_{j,2})^2}
\]

Initialize the fish swarm, randomly generate the initial population with the size of \( N \), calculate the fitness value of each fish, update the bulletin board information, and sort the fish from large to small according to the fitness.

In order to verify the effectiveness of the algorithm, 30 cities are used to verify the algorithm. The running environment of the system is Windows 7, and MATLAB 2016a is used for simulation experiment. The size of the fish swarm is set as \( n = 30 \), visual field is 8; the maximum iteration number \( \text{itermax} \) is set as 500, and the coordinates of each city are randomly generated by MATLAB. In order to verify the effectiveness of the experiment, a set of data generated is used as fixed points.

The initial graph is shown in Figure 2. The total length of the path is 1021.6818 units. After optimization, you can see the final path planning diagram as shown in Figure 3. The total path length is 435.4825 length units. The final result of optimization appeared in the 478rd generation. This final result is the shortest path visible to the naked eye. If the shortest path cannot be obtained in the end, it may be that the algorithm has fallen into a local optimal solution and cannot jump out. It may be solved by debugging the algorithm. The search process is shown in Figure 4:

![Flow Diagram](image-url)
It can be seen from the search process in Figure 4 that the algorithm has good convergence. The total length of the path decreases rapidly after the algorithm starts to run, and reaches the shortest path at about 480th generation. After that, it begins to oscillate, and the minimum value appears at 478th generation.

6. Conclusion
Vehicle routing problem is a common objective optimization problem. In daily life, intelligent algorithms are often used to calculate. From this example, we can see that the shortest path can be obtained quickly by optimizing the logistics distribution path with Fish Swarm Algorithm, and the algorithm converges quickly, which can be applied to the planning of logistics distribution path.

Reference:
[1] Whitley D.(1994) A genetic algorithm tutorial. Statistics and computing, 4(2): 65-85.
[2] Kennedy J, Eberhart R. (1995) Particle swarm optimization. In: Proceedings of ICNN’95 - international conference on neural networks. New York, pp. 4: 1942-1948.

[3] Karaboga D, Akay B. (2009) A comparative study of artificial bee colony algorithm. Applied mathematics and computation, 214(1): 108-132.

[4] Dorigo M, Birattari M, Stutzle T. (2006) Ant colony optimization. IEEE computational intelligence magazine, 1(4): 28-39.

[5] Li Xiaolei, Qian Jixin. (2002) Artificial Fish Swarm Algorithm. bottom up optimization model. System engineering theory and practice, 22 (3): 76-82.