An Improved Artificial Fish Swarm Algorithm to Solve the Cutting Stock Problem

Chunying Cheng[1(✉)] and Lanying Bao[2]

[1] College of Computer Science and Technology, Inner Mongolia University for Nationalities, Tongliao 028043, China
chengchunying_80@163.com
[2] College of Mathematics, Inner Mongolia University for Nationalities, Tongliao 028043, China

Abstract. In order to improve the utilization rate of sheet,an improved artificial fish swarm algorithm is proposed in this paper, which improved the preying behavior and swarming behavior, meanwhile set upper and lower limit for the Congestion factor of swarming behavior. Furthermore, the proposed algorithm is used to solve the cutting stock problem. After comparing the results of the simulation experiment with the improved particle swarm algorithm in the literature and the basic artificial fish swarm algorithm, it shows that the optimal solution obtained by the improved artificial fish swarm algorithm is better than the algorithm in the literature, thus improves the utilization rate of sheet.

Keywords: Artificial fish swarm algorithm · Cutting stock problem
Improved preying behavior · Improved congestion factor · Utilization rate

1 Introduction

The cutting stock problem can be described as a process of cutting square, rectangular and other small regular pieces as much as possible from given flat raw materials with a certain amount of methods. It's about how to place small rectangles of pre-determined size on a large, limited rectangular sheet which can achieve the highest utilization rate of sheet. The goal of the cutting stock problem is to minimize the area that might be dumped. Due to the requirements of saving raw materials, the cutting stock problem has become a very important research topic in the industrial production process of glass, paper, wood and steel. So far, many valuable methods have been proposed as solutions to the cutting stock problem, such as simulated annealing [1], genetic algorithm [2] and particle swarm optimization [3]. Artificial Fish Swarm Algorithm (AFSA) is a new random searching algorithm proposed by Li [4] in 2002 by observing the habits of fish swarms. This algorithm is a new idea of seeking the global optimal solution based on animal behaviors, which is a typical application of behavioral artificial intelligence. The algorithm starts with constructing simple underlying behaviors of animals and finally makes the global optimal value emerge in the group through the local optimization behavior of individual animals.

Artificial fish swarm algorithm is featured for its good ability to overcome the local optimal and obtain the global optimal solution, and the realization of the algorithm does

T. Huang et al. (Eds.): ISNN 2018, LNCS 10878, pp. 165–172, 2018.
https://doi.org/10.1007/978-3-319-92537-0_20

not need gradient values of the objective function, so it has some adaptive ability to the search space. However, the main application of artificial fish swarm algorithm is still focused on solving continuous optimization problems. When solving the combinatorial optimization problems, it will appear "premature" or fall into the local optimal solution, or the convergence speed is too slow. Therefore, in this paper improved preying behavior, swarming behavior and improved congestion factor are introduced into the basic artificial fish school algorithm, which can prevent the artificial fish from falling into the local extremum effectively and accelerate the search speed. Meanwhile, the artificial fish can jump out of the local extremum and make the population converge quickly to the global maximum in the end and reach the solution with high precision.

2 Basic Artificial Fish Swarm Algorithm

In an area of waters, where fishes appear most is often rich in nutrients in this area. So, artificial fish swarm algorithm is based on this character to imitate the preying behavior of fishes, in order to achieve the optimization of the solution.

(1) Related definitions
Individual status of the artificial fish can be expressed as a vector, where $x_i(i = 1, 2, \cdots, n)$ are variables for optimization, the food concentration where the artificial fish currently locates is expressed as $Y = f(X)$, Y means the objective function value, the distance between individuals artificial fishes is expressed as $d_{ij} = \|X_i - X_j\|$, *Visual* means the perception distance of the artificial fish, *Step* means the moving step of the artificial fish, δ means the congestion factor.

(2) Description of the fish behavior

① Preying behavior
Let the current status of the artificial fish be X_i, we randomly select a status X_j within the scope of its perception. If the problem is for the search of the maximal value $Y_i < Y_j$ (or for the search of the minimal value, $Y_i > Y_j$, because the maximum problem and minimum problem can be converted to each other), then move a step in this direction. On the contrary, it will randomly select the status X_j again, and determine whether it can meet the conditions of moving forward. If it still cannot meet the conditions of moving forward after repeating several times, then it moves a step randomly.

② Swarming behavior
Let the current status of the artificial fish be X_i, it will explore the number n_f of partners in the current neighboring area (i.e. $d_{ij} < Visual\, d_{ij} < Visua$) and its center position X_c, and if $Y_c/n_f > \delta Y_i$, it indicates that the partner center has more food and is less congested, then the artificial fish will proceed one step closer to the partner's center position, otherwise it will execute its preying behavior.

③ Following behavior
Let the current status of the artificial fish be X_i, it will explore whether Y_j is the largest one X_j among partners who are in the current neighboring area. If $Y_c/n_f > \delta Y_i$, it indicates the partner X_j has a higher food concentration and it

is not crowded around, move one step further toward the partner X_j, otherwise it will execute its preying behavior.

④ Stochastic behavior

The implementation of stochastic behavior is simpler and it can be performed as follows: select a status randomly within the field of view and then move in that direction, which is actually a default behavior of preying behavior.

(3) Choice of behavior

According to the characteristics of the problem to be solved, assess the current situation of the artificial fish in order to choose a behavior. The easiest way to find the maximum is to use heuristics. The implementation is as follows: simulate the swarming, following and other behaviors, and then evaluate the value after the action, select the largest of them to execute the actual implementation, and the default behavior is the preying behavior.

(4) Bulletin board

The bulletin board records the current status of the best individual artificial fish. After every action, each artificial fish is compared with the individual status of the artificial fish on the bulletin board. If it is better than individual status of artificial fishes on the bulletin board, it will update individual status of the artificial fish on the bulletin board with its own status.

3 The Proposed Algorithm to Solve the Cutting Stock Problem

3.1 Encoding

In this paper, the sheet and the small pieces to be placed are encoded with free coordinates. The lower left corner of the large sheet is placed at the coordinate origin, each small pieces to be placed are represented by a 4-dimensional vector (x_k, y_k, l_k, w_k), where (x_k, y_k) is the position coordinates of the lower left corner of small pieces on the sheet, (l_k, w_k) is the length and the width of small pieces. That is, $0 \le x_k, l_k \le X_{\max}$, $0 \le y_k, w_k \le Y_{\max}$, where $X_{\max}$ and $Y_{\max}$ represent the length and the height of raw materials. Each individual in the artificial fish swarm represents a cutting stock pattern, and each cutting stock pattern consist a sequence of small pieces. The established coordinate system is as follows (Fig. 1):

The algorithm of cutting stock is about how to place the pieces to be cut in the right place on the sheet. Here the conversion is performed with a method similar to the Bottom Left Algorithm (BL) [5]. This method is called coordinate-based Bottom Left Algorithm (CBL). According to the rules of the Bottom Left algorithm, the pieces to be cut are placed in the upper right corner of the coordinate system. In order to reduce the amount of effort of moving back and forth when cutting the sheet, it is arranged in advance in the fixed order of x_k which is described by the encoding method of cutting sheet. The actual operation of the algorithm is as follows:

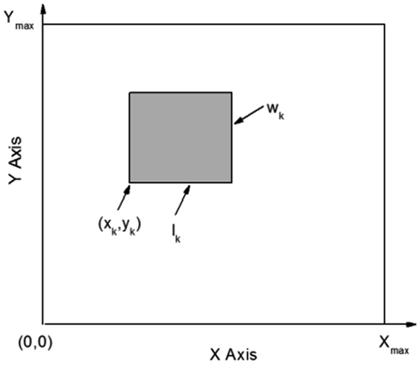

Fig. 1. Coordinate system of cutting stock

Step 1: Sort each small piece x_k ascendingly.
Step 2: The movement rule of small pieces to be cut is as follows: place small pieces according to the remarks of x_k and y_k, the left side and the top of the small pieces is placed inside the big sheet absolutely, and they will move on the direction of "first left then down" when they do not overlap other sheets. "First left then down" means it will move to the left as much as possible. And then move down when reaching the right side of the large sheet or the left side of other small sheet. Move based on the principle of "first left then down" again as much as possible when reaching the side of the large sheet or a small sheet. If there is a piece of sheet overlapping a placed sheet, then arrange it into the order of the sheet to be cut. After each small piece is placed, go to following step.
Step 3: The sheet where small pieces are not placed in Step 2 moves according to the principle of "first left then down". When the cut sheet cannot move further to the left and down and the surrounding edges are completely placed inside or the edge is aligned with the raw material, it is its final position, otherwise, this one should be abandoned.

3.2 Improvement of Artificial Fish Swarm Algorithm

The solution to the cutting stock problem is to find a place of higher concentration of food, which is described with $Y = f(X)$, where represents the artificial fish status, d_{ij} represents the length between the artificial fish X_i and X_j, the distance between the two fishes, $\max number$ represents the iteration number of the artificial fish, *Visual* represents the visual field of artificial fish, *Step* represents the distance each time artificial fish moves, N represents the number of individuals of artificial fish swarm.

(1) Design of fitness function
In the improved cutting stock algorithms of artificial fish swarm, the sheet utilization rate is used to represent the fitness function. Therefore, the fitness function $fitness(x)$ can be designed as:

$$fitness(x) = f(M) \tag{1}$$

Therefore, the utilization rate of the improved algorithm can be expressed as:

$$f(x) = \frac{r(M)}{S} \tag{2}$$

In formula (2), M represents each completed process in the actual design, a cutting stock pattern, $r(M)$ represents the total area of the small pieces cut by the algorithm, and S represents the area of the raw material.

(2) Improvement of preying behavior
Implement the optimal individual retention strategy on preying behavior to avoid the degradation of optimal artificial fish status. Set the current status of artificial fish as X_i, randomly select a status X_j within the perception range of the artificial fish. If the food concentration is $Y_i < Y_j$, then move forward in this direction. On the other hand, randomly select the status X_j again to determine whether the conditions for moving forward are satisfied. If the conditions for moving forward are still not satisfied after repeating several times and the current status X_i is not the optimal status in the current group, then move one step randomly. If it is the optimal status in the current group, do not move.

(3) Improvement of swarming behavior and congestion factor
Set the upper limit $\delta_{\max}$ and the lower limit $\delta_{\min}$ of the congestion factor δ, so as to restrict the conditions for swarming behavior to occur at $[\delta_{\min}, \delta_{\max}]$ and forbid the swarming behavior of $\delta < \delta_{\min}$ or $\delta > \delta_{\max}$ at the same time. When the congestion factor exceeds $\delta_{\max}$, let the other fish swarms in the range $1 - \delta_{\max}$ perform improved preying behavior. When the congestion factor is lower than $\delta_{\min}$, dismiss the congested fish swarm forcibly and let them perform the preying behaviors individually. This congestion behavior can prevent the occurring of "premature", and thus improve the ability of global optimization.

(4) Realization of the algorithm
$X_i = (q_{i1}, q_{i2}, \cdots, q_{in})$ represents a fish, where $i = 1, 2, \cdots, N$. N represents the number of fish swarms. q_{ik} represents a small piece that can be cut. The position of the coordinate system describes the raw materials and the small sheet to be cut. The lower left corner of a large sheet is placed at the origin of coordinates, each small piece to be placed is represented by a 4-dimensional vector $q_{ik} = (x_{ik}, y_{ik}, l_{ik}, w_{ik})$, where $k = 1, 2, \cdots, n$. n represents the number of sheet to be cut. (x_{ik}, y_{ik}) is the position of x axis and y axis of the sheet to be cut, (l_{ik}, w_{ik}) represents the length of the cut piece on the horizontal line and the height of the vertical line. The distance between the artificial fish X_i and the current X_j can be represented as:

$$d_{ij} = \|X_i - X_j\| = \frac{1}{n}\sum_{k=1}^{n}\|q_{ik} - q_{jk}\| = \frac{1}{n}\sum_{k=1}^{n}\sqrt{(x_{ik} - x_{jk})^2 + (y_{ik} - y_{jk})^2} \tag{3}$$

In the artificial fish swarm algorithm, the food concentration at the current position of the artificial fish is represented as the utilization rate of the sheet. That is, among $f(M) = \frac{r(M)}{S}$, M means the time when the specific setting of artificial fish swarm algorithm for the cutting stock problem is completed, $r(M)$ is the sum of areas of small pieces which could be cut from the raw materials, $S = Length * Width$ is the area of the raw material.
The steps for the improved artificial fish swarm algorithm for the cutting stock problem are described below:

Step 1: Initialize each artificial fish randomly, that is, each piece placed on the sheet is at the lower left corner.
Step 2: Calculate the food concentration value of the artificial fish and its partners. Compare the food concentration value with that on the bulletin board, and put the artificial fish with higher food concentration onto the bulletin board.
Step 3: Each artificial fish performs following behavior, improved swarming behavior and improved preying behavior respectively. After each execution, the food concentration at the new position is compared to the record on the bulletin board. Update the bulletin board when the current value is better than that on the bulletin board, and go to step 4. Otherwise, choose the next action based on the current behavior. The specific behaviors are as follows: If the food concentration in the new position is lower than the record on the bulletin board after the following behavior, perform the improved swarming behavior; if the food concentration in the new position is lower than the record on the bulletin board after the swarming behavior, perform the improved preying behavior.
Step 4: Stop the iterations if the maximum number of iterations or the threshold is reached. Otherwise, go to step 3 to continue.

(5) Selection of numerical examples and parameters
In this paper, 10 experiments were carried out, the number of sheets to be cut are ranged from 10 to 30 pieces. In this paper, Visual C++ language is used as a programming tool, the experimental environment is as follows: AMD Phenom (tm)II X6 1065T Processor, 2.9 GHz CPU, 4 GB memory, Windows 7 operating system. In order to verify the effectiveness of the improved artificial fish swarm algorithm to solve the cutting stock problem, in this paper the improved artificial fish swarm algorithm is used to solve the experimental cases in [3, 6]. The parameters were set as follows: the number of artificial fishes is 50, trying number *trynumber* in preying behavior is 20, congestion factor $\delta_{\max}$ is 0.6, $\delta_{\min}$ is 0.2, the maximum number of iterations is 2000, field view *Visual* of artificial fish in Example 1 to Example 5 are valued as 20, 30, 20, 30, 50 respectively.
Table 1 shows the properties of the test cases used in [3, 6]. Figure 2 shows the cutting results of the example 5 through the improved artificial fish swarm algorithm. Table 2 shows the loss rate of experimental cutting in the actual example.

Table 1. The size of the test cases.

No.	Number of pieces	Size of stock plate	Size of pieces
1	10	(30 × 40) × 2, (10 × 40), (20 × 40) × 3, (20 × 50), (20 × 30), (30 × 20) × 2	100 × 80
2	10	(20 × 4), (16 × 4), (10 × 6) × 2, (16 × 6), (4 × 10), (10 × 5) × 2, (10 × 10), (20 × 10)	40 × 20
3	15	(5 × 6) × 4, (10 × 3), (12 × 5) × 2, (10 × 7) × 2, (20 × 4) × 2, (10 × 5) × 3, (8 × 10)	40 × 20
4	20	(11 × 15) × 4, (17 × 5) × 4, (12 × 4) × 4, (3 × 9) × 4, (5 × 15) × 4	40 × 40
5	30	(17 × 6), (6 × 9) × 3, (6 × 12) × 5, (9 × 12) × 4, (9 × 6) × 2, (17 × 12), (14 × 9), (12 × 9) × 2, (6 × 15) × 2, (9 × 15), (11 × 12), (9 × 18), (9 × 9), (15 × 9), (15 × 6), (15 × 12), (6 × 6), (14 × 6)	65 × 45

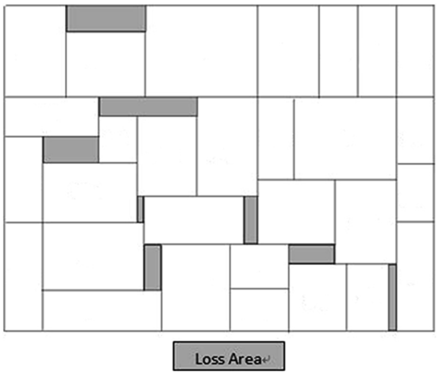

Fig. 2. Cutting results running hybrid artificial fish swarm algorithm

Table 2. Comparison of experimental results

No.	1	2	3	4	5
PSO loss rate (%)	0	0	7.5	6.3	7.1
Literature [3] loss rate (%)	0	0	0	6.3	6.8
Literature [6] loss rate (%)	0	0	0	3	6.6
IAFSA loss rate (%)	0	0	0	3	5.53

(6) Results analysis of numerical experiment
In this paper, 10 experiments are carried out. In Fig. 2, the experimental results of the improved artificial fish swarm algorithm are given. With the improved artificial fish swarm algorithm, 29 blocks can be cut out, and the shaded area

represents the area which cannot be used. Table 2 shows the contrast of the loss rates of the cutting stock problem with the same data through the improved particle swarm optimization algorithm, the basic artificial fish swarm algorithm and the improved artificial fish swarm algorithm in this paper. Looking upon the simulation results, it is obvious that the improved artificial fish swarm algorithm has a better performance in solving the cutting stock problem.

4 Conclusion

Through the simulation experiment, it is feasible to use the improved artificial fish swarm algorithm to solve the cutting stock problem. After comparing it with the improved particle swarm algorithm and with the basic artificial fish swarm algorithm in the literature, the experimental results show that the improved algorithm can obtain the most optimal solution better than the algorithm in the literature, and solve the problem about the utilization of sheets.

References

1. Changzheng, X.: Optimal board cutting based on simulated annealing genetic algorithm. J. LiaoNing Tech. Univ. (Nat. Sci. Ed.) **25**(3), 406–408 (2006)
2. Xingfang, Z., Yaodong, C., Ying, Y.: A genetic algorithm for the rectangular strip packing problem. J. Comput.-Aid. Des. Comput. Graph. **20**(4), 540 (2008)
3. Qijinbao, B., Jingqing, J., Chuyi, S.: Optimal stock cutting based on particle swarm optimization and simulated annealing. Comput. Eng. Appl. **44**(26), 246–248 (2008)
4. Li, X., Shao, Z.: An optimizing method based on autonomous animals: fish-swarm algorithm. Syst. Eng. Theory Pract. **22**(11), 2–38 (2002)
5. Leung, T.W., Yung, C.H., Troutt, M.D.: Applications of genetic search and simulated annealing to the two-dimensional non-guillotine cutting stock problem. Comput. Ind. Eng. **40**, 201–214 (2001)
6. Bao, L., Jiang, J., Song, C., Zhao, L., Gao, J.: Artificial fish swarm algorithm for two-dimensional non-guillotine cutting stock problem. In: Guo, C., Hou, Z.-G., Zeng, Z. (eds.) ISNN 2013. LNCS, vol. 7952, pp. 552–559. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-39068-5_66