Appl Intell (2016) 45:992–1007
DOI 10.1007/s10489-016-0798-7

AFSAOCP: A novel artificial fish swarm optimization algorithm aided by ocean current power

Hong-bo Wang[1,2] · Cheng-Cheng Fan[1,2] · Xu-yan Tu[1,2]

Published online: 23 June 2016

Abstract Artificial Fish Swarm is a kind of swarm intelligence algorithm, which focuses on the behavior of individual fishes and information interactions among them during foraging and preying on something in real environment. A novel Artificial Fish Swarm Optimization Algorithm Aided by Ocean Current Power (abbreviation for AFSAOCP) is proposed, which assumes the ocean current always causes certain influence on the fishes' activity speed. Firstly, the computing model of ocean current is developed and constructed. Then the influence level of the ocean current on fishes is analyzed. If fishes are swimming along ocean current direction, ocean current will drive fishes' speed increment, which is called positive influence; if fishes are swimming against ocean current, the current will hinder the fishes' speed, which is called negative influence. In addition, fishes' speed is not influenced by the ocean current, which is called merits offset faults. To sum this up, fishes in each group have different speed range, respectively. Grouping strategies can not only increase species diversity, but it can also make the algorithm escape from local optimal value in the iteration process. The proposed variant, AFSAOCP, is examined on several widely used benchmarked functions, and the experimental results show that the proposed AFSAOCP algorithm improves the existing performance of other algorithms when dealing with the different dimension and multimodal problems.

Keywords Artificial swarm algorithm · Swarm intelligence · Ocean current power · Optimization problem

1 Introduction

Swarm intelligence algorithms have been widely used in different areas in order to solve various problems. There are many algorithms in swarm intelligence that are developed by simulating the behaviors of the creatures in nature [1, 20, 21]. There are some relatively popular algorithms such as ant colony optimization algorithm (ACO) [2], particle swarm optimization algorithm (PSO) [3] and so on. Among them, a novel optimizing method, Artificial Fish Swarm Algorithm (AFSA), was referred to the behavior of fish swarm. It was proposed by the domestic scholars LI Xiao-lei, SHAO Zhi-jiang and QIAN Ji-xin in 2002 [4]. Similar to most social animals, the fish has its unique way of living. In the water, most fishes exist in the place that has rich food. According to this, AFSA is a kind of swarm intelligence algorithm that imitates the specific behavior of individuals and information interactions among them during preying process in real environment. As far as order is concerned, in AFSA each artificial fish (AF) individual explores food by four behaviors, which are random moving, preying, following, and swarming. Preying behavior laid the foundation of algorithm convergence, swarming behavior enhances the stability and global convergence of the algorithm, following behavior speeds up the algorithm convergence, evaluating of the four behaviors provides guarantee of algorithm convergence speed and stability. Each AF individual has a self-information that includes visual range,

✉ Hong-bo Wang
foreverwhb@ustb.edu.cn

1 Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing, Beijing 100083, People's Republic of China

2 Beijing Key Laboratory of Knowledge Engineering for Materials Science, No.30 Xueyuan Road, Haidian Zone, Beijing, 1000833, People's Republic of China

current position, food concentration, moving step length, crowing divisor etc., which decided the fish chooses next behavior in the range of motion, and these behaviors can influence each other. Ref. [5, 22] expounded five characteristics of AFSA: (1) parallelism, (2) simplification, (3) global searching ability, (4) fast convergence and (5) not sensitive to the requirements of the objective functions. Therefore, more scholars have studied on AFSA to pursuit better performance in many aspects.

Many practical (industrial/engineering) and challenging problems are multi-objective, where the related solution is a set of points that give trade-offs for the different targets. Introducing multi-objective optimization for AFSA is inspired by our team's research in evolutionary computing for similar problems [6]. In this paper, a new method named "A novel Artificial Fish Swarm Optimization Algorithm Aided by Ocean Current Power (AFSAOCP)" has been proposed.

Section 2 introduces the standard AFSA. In Section 3, a detailed description about some recent variants is put forward. Section 4 presents the proposed AFSAOCP. Section 5 makes comparative experiments on several widely used benchmark function and analyzes the related experimental results, and conclusions are made in Section 6.

2 Basic Artificial Fish Swarm Algorithm (AFSA)

Artificial Fish in the environment is the main part of the problem solution space, where an Artificial Fish next-step behavior depends on the previous moment and the current state of environment. Artificial Fishes exert influence on their neighbor or companions by their own activities and vice versa

As shown in Fig. 1, an Artificial Fish perceives external things with sense of sight. Current position of an Artificial Fish can be written as: $X = \{X_1, X_2, ..., X_n\}$, where $X_i (i = 1, 2, ..., n)$ is the control variable. The $Visual$ is used to indicate sight field of an Artificial Fish and X_v is a position in $visual$ where AF's view for a moment. The step length is expressed as $Step$ and the next position of an Artificial Fish is described as X_{next}. If X_v has better food consistence than current position of AF, it will consider to go one $Step$ toward X_v, which causes change in AF position from X to X_{next},but if the current position of AF is better than X_v, it continues searching in its $Visual$ area. The process can be expressed as follows in formulas (2-1) and (2-2):

$$X_v = X + Visual \bullet Rand() \tag{2-1}$$

$$X_{next} = X + \frac{X_v - X}{||X_v - X||} \bullet Step \bullet Rand() \tag{2-2}$$

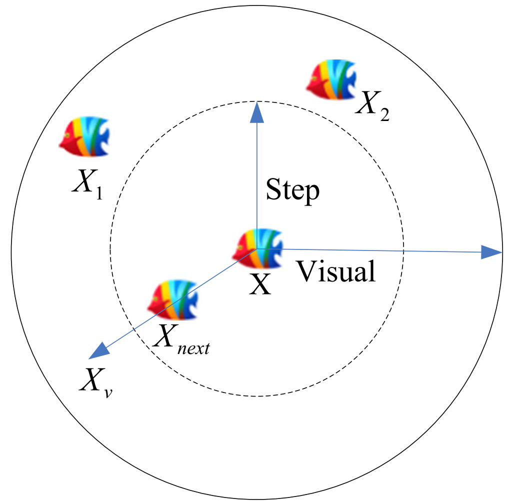


Fig. 1 The concept of AF's vision

In the two formulas, $Rand()$ denotes some random numbers which are between 0 to 1.

Food consistence in position X represents fitness value of this position and $Y = f(X)Y$ is an objective function. The distance between two AFs which are in t and X^t positions is shown by $d_{ij} = |X_i - X_j|$. The maximum tries of AF during preying process is represented by $BestFish^t = X^*$, which means the maximum number of attempts by the fish.

AF has four kinds of behaviors and are described as follows:

1) Preying behavior

This is a natural behavior of AF that is the tendency of food, the current position of AF i is X_i and randomly chooses a position in its sight field is x_j by formula (2-3).

$$X_j = X_i + Visual \bullet Rand() \tag{2-3}$$

If $Y_i < Y_j$, the AF makes a step towards the position of X_j by formula (2-4).

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{\parallel X_j - X_i^t \parallel} \bullet Step \bullet Rand() \tag{2-4}$$

But while $Y_i > Y_j$ is true, the position will be updated, namely, position X_j is selected again, and then AF needs to judge whether it satisfies the forwarding condition or not. If Try_number still does not find the fitness condition, AF will move a new position randomly by formula (2-5).

$$X_i^{t+1} = X_i^t + Visual \bullet Rand() \tag{2-5}$$

The Pseudo code of preying behavior is show as follows:

```
float prey()
{
    for(i = 0; i<Try_number; i++)
    {
        X_j = X_i + Rand() • Visual;
        if(Y_j<Y_i)
            X_{i/next} = X_i + Rand() • Step • (X_j − X_i)/||X_j − X_i||;
        else
            X_{i/next} = X_i + Rand() • Step;
    }
return foodconsistence(X_{i/next});
}
```

2) Swarming behavior

During searching for food, AFs will group together spontaneously so as to guarantee their own survival. This kind of behavior can not only be helpful to reduce the amount of AFs that are trapped into local optimal solutions, but are also useful to gather around the regions of the majority AFs that have global optimal solution. There are two action rules among AFs in the AFSA: (1) as far as possible move toward the center of the adjacent partners; (2) to avoid overcrowding. The current position of AF i is X_i, the number of partners in the current field $(d_{ij}<Visual)$ is n_f and center position is X_c. If $Y_c/n_f > \delta Y_i$, there is plenty of food in the partner's center and it's not too crowded here. Then the AF can move a step towards that central position according to formula (2-6). Otherwise, it executes preying behavior.

$$X_i^{t+1} = X_i^t + \frac{X_c - X_i^t}{\parallel X_c - X_i^t \parallel} \bullet Step \bullet Rand() \tag{2-6}$$

The Pseudo code of swarming behavior is show as follows:

```
float swarm()
{
    n_f = 0; X_c = 0;
    for(j = 0; j<FishNumber; j++)
    {
        if(d_{i,j}<Visual)
        {n_f++; X_c+ = X_j; }
        X_c = X_c/n_f;
        if(Y_c/n_f > δY_i)
            X_{i/next} = X_i + Rand() • Step • (X_j − X_i)/|| X_j − X_i ||;
        else
            prey();
    }
return foodconsistence(X_{i/next});
}
```

3) Following behavior

When AF finds a region that has enough food and is not too crowded, AF's nearby partners will follow it and quickly reach to this hot point. Following behavior is a kind of pursuit behavior where the neighbors have highest fitness. In optimum algorithm, it can be understood as a process of moving forward nearby optimal partner. The current position of AF is X_i, AF searches for partner X_j with the maximum food consistence in current field$(d_{ij}<Visual)$, written as Y_j. If $Y_j/n_f>\delta Y_i$, there is enough food in the partner's center and it's not too crowded here. Then, the AF can move a step towards that central position according to formula (2-7). Otherwise, it executes preying behavior.

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{\parallel X_j - X_i^t \parallel} \bullet Step \bullet Rand() \tag{2-7}$$

The Pseudo code of following behavior is show as follows:

```
float follow()
{
    Y_min = ∞;
    for(j = 0; j<FishNumber; j++)
    {
        if(d_{i,j}<Visual&&Y_j<Y_min)
        {Y_min = Y_j; X_min = X_j; }
        n_f=0;
        for(j = 0; j<FishNumber; j++)
            if(d_{min,j}<Visual)
            {n_f++; }
            if(Y_min/n_f > δY_i)
                X_{i/next} = X_i + Rand() • Step • (X_min − X_i)/|| X_min − X_i ||;
        else
            prey();
    }
return foodconsistence(X_{i/next});
}
```

4) Random behavior

To enlarge its search space, artificial fish will move a step randomly in its perception range and reach to a new position. This is a simple behavior that chooses a position randomly in AF's sight field, and then move forward to it. In a word, it is a default behavior of preying behavior.

AFSA should initialize relative parameters, evaluate the four behaviors and choose the optimal behavior. The procedure of the basic AFSA is show as follows:

Step 1: Initializing parameters.
Step 2: Randomly generating initial population

Step 3: Calculating every fish's food concentration (objective function), and putting the optimal value in the bulletin board

Step 4: For each of AF

1) Calculating the fitness value of following behavior and swarming behavior, and then selecting the optimal behavior as the moving direction of AF by selection strategy, the default behavior is preying behavior.
2) Calculating every AF's food concentration, and its optimal value compared with the values in the bulletin board, the bulletin board always maintains optimal values.

Step 5: Determining whether it satisfies the end's condition, if satisfy ends, otherwise go to step 4.

3 The related research

The basic AFSA requirement is not limited in the objective function, parameters and initial values, but it's one of the most effective algorithms in solving optimization problems. Many researchers have worked on improving its performance in various ways and developed many variants.

In Ref. [7], a method of population adaptation which has the feature of fast convergence, good global search capability, strong robustness. In this paper, the author applied the Artificial Fish Swarm (AFSA) to multi-objective optimization problems. An improved AFSA named IAFSA is proposed in Ref. [8], where the author suggests that adjust parameters $_{Visual}$ and $_{Step}$ adaptively with iterative operation is necessary, so $Visual$ and $Step$ are put forward to optimize the parameters of LS-SVM. Ref. [9] introduces an improved algorithm that two new adaptive methods based on AFSA execution in order to control the capability of global and local searching adaptively. In this paper, firstly, selected larger initial value for $Visual$ and $Step$. After that, by approaching the target, AF can accurately investigate the environment by smaller $Visual$ and $Step$. In Ref. [10], an improved AFSA by genetic algorithm is proposed, where the variation factor of genetic algorithm is introduced to AFSA. With a concept that is mentioned when $(recordFC - FC < eps)$ has appeared for several times continuously, it will execute variation factor on each AF's parameter with probability p. An improved AFSA named HAFSA is applied in Ref. [11]. This algorithm is based on PSO and AFSA: it makes full use of the fast local convergence performance of PSO and the global convergence performance of AFSA, and then is used for solving ill-conditioned linear systems of equations. Modified artificial fish swarm algorithm (MAFSA) is proposed to optimize the reactive power optimization is in Ref. [12].

Based on the above discussion, a novel Artificial Fish Swarm Optimization Algorithm Aided by Ocean Current Power (called AFSAOCP) fully considers the characteristics of fish life. The algorithm improves the performance of optimization algorithm through the implementation of a new mutation strategy and demonstrates a significant performance improvement over the AAFSA1 (the adaptive step length) [13], AAFSA2(introduces a new behavior) [14] and IAFSA [15].

4 The AFSAOCP

In order to keep balance between global search ability and local search ability of artificial fish swarm algorithm, step strategies can help AF timely to arrive at the extreme value point of convergence. But if the step length exceeds a certain range, it can possibly slow convergence speed, but also appear even oscillation phenomenon. In order to deal with these problems, and inspired by the phenomenon of symbiosis, AFSAOCP is put forward.

4.1 Ocean current

The ocean current is the surface water in the density of sea water, wind, and a variety of other factors which influence it along the direction of a certain large scale regular flow [16].

There are dozens of major ocean currents, often called the high temperature from tropical sea warm current, known as a cold snap from cold water temperature is relatively low. Ocean current is generally divided into wind current, compensation flow and density flow, and wind current is the main form [17, 18].

Figure 2 is the wind current schematic diagram. If the water on the surface of the ocean is in the atmospheric circulation and near the ground under the action of wind current, we call it wind current. In the northern hemisphere, the wind current is flowing by clockwise direction around the subtropical high. However, in the southern hemisphere, it is anticlockwise direction. The wind current is close to the ground under the action of wind, and to produce water friction *coriolis force balance* of current. There is a significant impact on ocean current in the global scope of sea: Pacific equatorial current, north equatorial current, south equatorial current, north Pacific warm current and warm Atlantic [19].

This above is a macroscopic description of ocean current. For microcosm, on the one hand, we know that there are many reasons that cause currents, one of which was caused by uneven terrain. According to the flow, it's not parallel degree and bending degree, but the ocean is divided into the gradient flow and rapidly varied flow. The gradient flow refers to streamline the parallel linear flow. Rapidly varied flow refers to the curvature of the streamline, the flow that

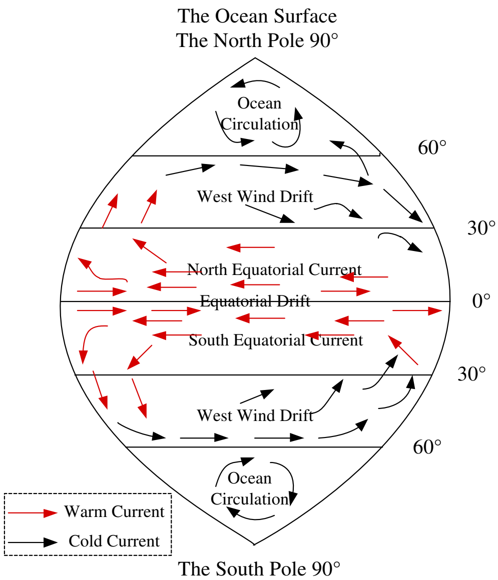


Fig. 2 The formation of the ocean current

has the bigger curvature or included angle. It is shown in Fig. 3.

On the other hand, we know that wind current is the main form of ocean current. Therefore, with the increasing depth of the sea, the influence of the wind on the water is more and more small. In other words, the flow velocity of ocean current reaches the maximum at the surface, slowing down with the increase of depth. In order to facilitate research, we divided the ocean into three layers (shown in Fig. 4). From the top to the bottom velocity decreases, which is for us to face the artificial fishes after grouping analysis is done. The flow rate from the ocean surface to the seabed decreasing, which is based on artificial fish grouping.

Next, let us analyze the situation of fishes in the ocean. In order to facilitate the research, we assume an artificial fish as a particle which is in a three-dimensional space. The particle under the action of external force is affected by the following Fig. 4.

In the Fig. 5, particle P is moving along the Y axis. At the same time, the velocity component v_1 and v_2, which

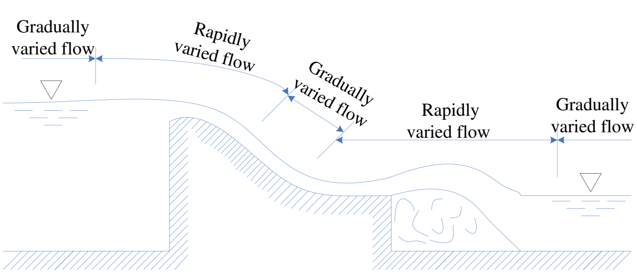


Fig. 3 Gradually varied flow and Rapidly varied flow

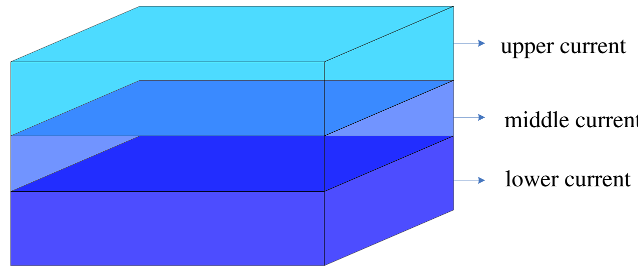


Fig. 4 The hierarchical diagram of ocean

are coming from the X axis and Z axis, affect P's speed. The trajectories of P will be decided according to the three directions of the three speeds eventually. The detailed trajectory is shown in Fig. 5.

In Fig. 6, the X axis and Y axis show the flow of ocean current, fishes from point A to point B. *R1* showed that fishes' swimming route without ocean current effect. *R2* said fishes under the influence by ocean current move to point B with curve of the parade route.

4.2 New searching strategy

Since the ocean current is one of the Marine natural phenomena, its velocity will inevitably affect the speed of artificial fishes. Firstly, AF is swimming along ocean current, saying there is a pair of invisible hand to promote AF forward. Details are followed in Fig. 7.

Secondly, if AF is swimming against the current, their ability to pursue food may be impeded. Details are followed in Fig. 8.

AF depends on a swimming motion to go from place to place in their search for food, and gradually they populated the ocean with different speed. With this in mind, we divide artificial fishes in different groups according to their speed range. By comparison, the optimal fish (local best) in each group can be chosen out, at the same time each iteration of

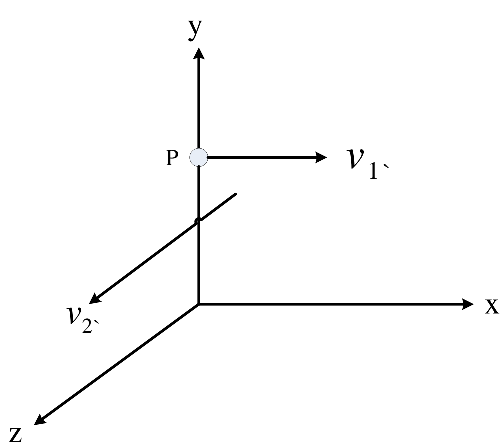


Fig. 5 Effects of the particle in three-dimensional space

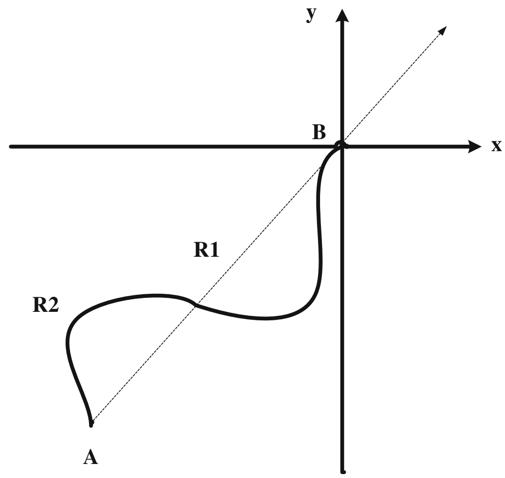


Fig. 6 The trajectories of fishes

the optimal fish (global best) can also be concluded. The graphical description of grouping and procedures of each subgroup will be shown in Fig. 9.

In Fig. 9, Artificial fishes are divided into three groups by the strength of influence (‘+’, ” and ‘’), each subgroup is moving with the different step length. The normal artificial fishes’ (‘’ subgroup) speed is v, the speed of ocean current is v_{ocean}, so the step length is $Step$ and mobile way according to formula (2-7). Therefore, in contrast to the current direction (that is adverse current) of artificial fishes’ speed isv_1(‘+’ subgroup), we know$v_1 > v$, so the step length is $Step + v_{ocean} \cdot \cos\alpha$, and mobile way according to formula (4-1).

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{\| X_j - X_i^t \|} \bullet (Step + v_{ocean} \cdot \cos\alpha) \bullet Rand() \tag{4-1}$$

The same as the current direction (that is fair current) of the artificial fishes’ speed is v_2(‘’ subgroup), we know $v_2 < v$,

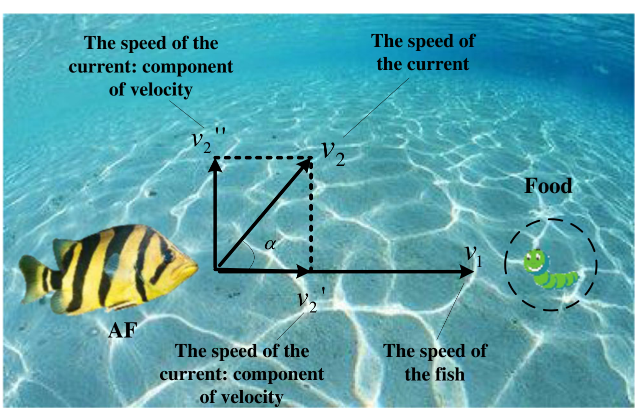


Fig. 7 Ocean currents make a positive difference in fish’s preying (‘+’ influence)

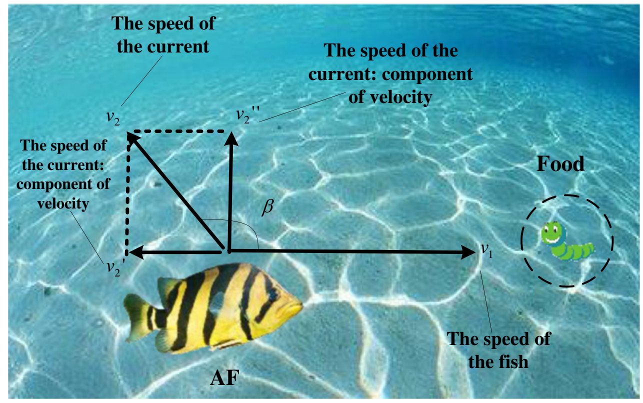


Fig. 8 Ocean currents make a negative difference in fish’s preying (‘−’ influence)

so the step length is $Step - v_{ocean} \cdot \cos(\pi - \beta)$. And mobile way is computing according to formula (4-2).

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{\| X_j - X_i^t \|} \bullet (Step - v_{ocean} \cdot \cos(\pi - \beta)) \bullet Rand() \tag{4-2}$$

after each iteration finishes, artificial fishes will be divided into three subgroups again.

4.3 Procedures of AFSAOCP

The procedures of the AFSAOCP is similar to the AFSA described in Section 2, including initialization, evaluation of behaviors, and selection of the optimal behavior. Repeat these steps until the conditions of the end of the algorithm are met.

But there is a difference between AFSAOCP and AFSA. The main difference is that considering the influence of the ocean current to living conditions of the fish in Section 4.1, we improve the evolution of fish: each individual in the initial population will generate three subgroups to improve the convergence speed through the iteration of the algorithm. So flow chart of AFSAOCP is shown in Fig. 10.

In Fig. 10 and Table 1:

- $FishNum$: The number of artificial fishes.
- b_value : The best value of every iteration
- $iterNum$: The maximum number of iterations
- $tryNum$: The maximum number of attempts
- Y_i : The fish *i* fitness values
- $value\_AFSAOCP$: The optimal value of the AFSAOCP.

According to the above chart of the AFSAOCP, the relative algorithm is defined as follows. Pseudo code of the AFSAOCP is shown in Table 1 and the grouping part of the algorithm is from line 04 to line 37.

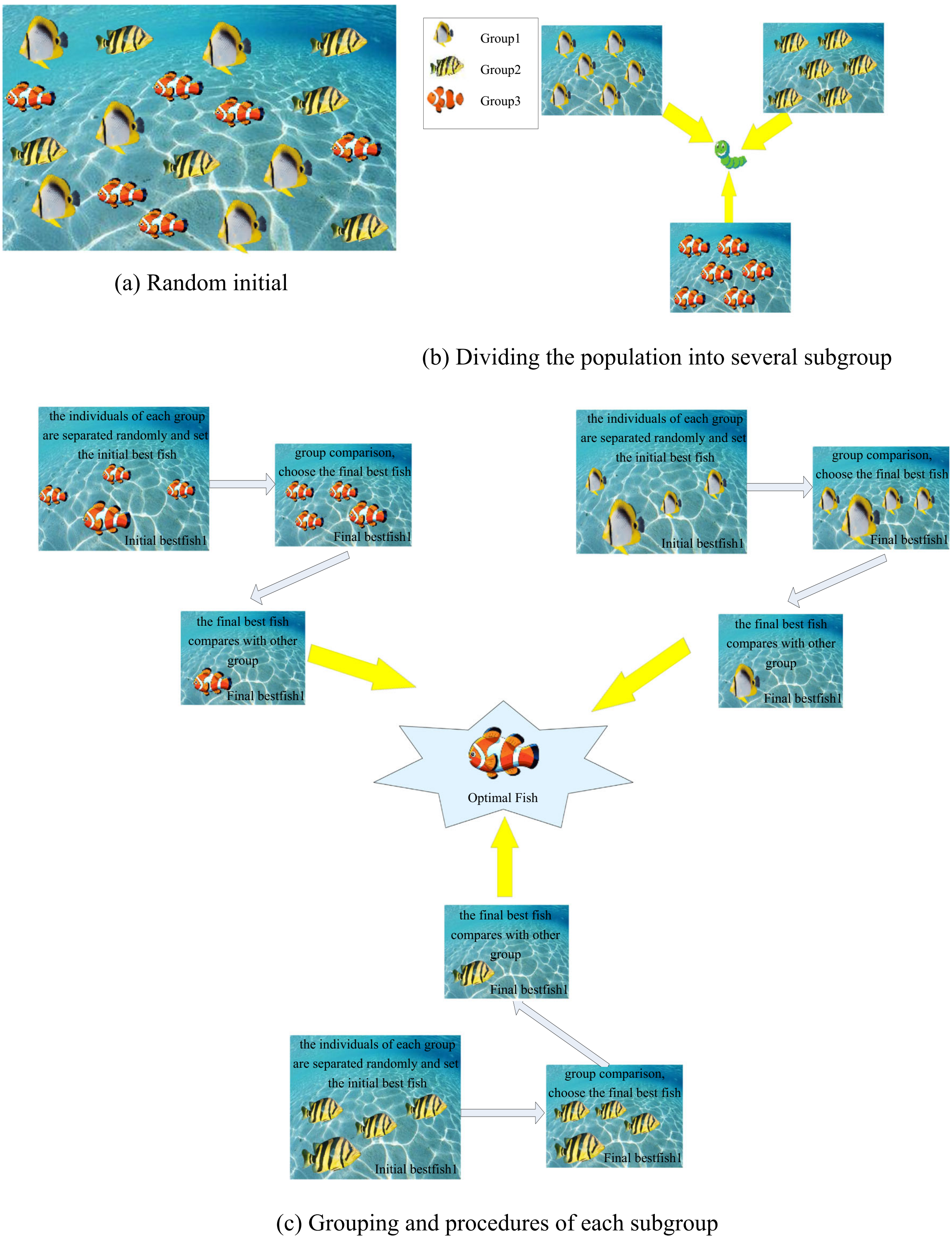


Fig. 9 Grouping and procedures of each subgroup

5 Experimentation

In this section, the proposed AFSAOCP is compared with the AFSA, AAFSA1, AAFSA2 and IAFSA .

5.1 The analysis of parameters

The main basic parameters of AFSAOCP is the number of artificial fishes $FishNum$, field of vision $Visual$, crowded

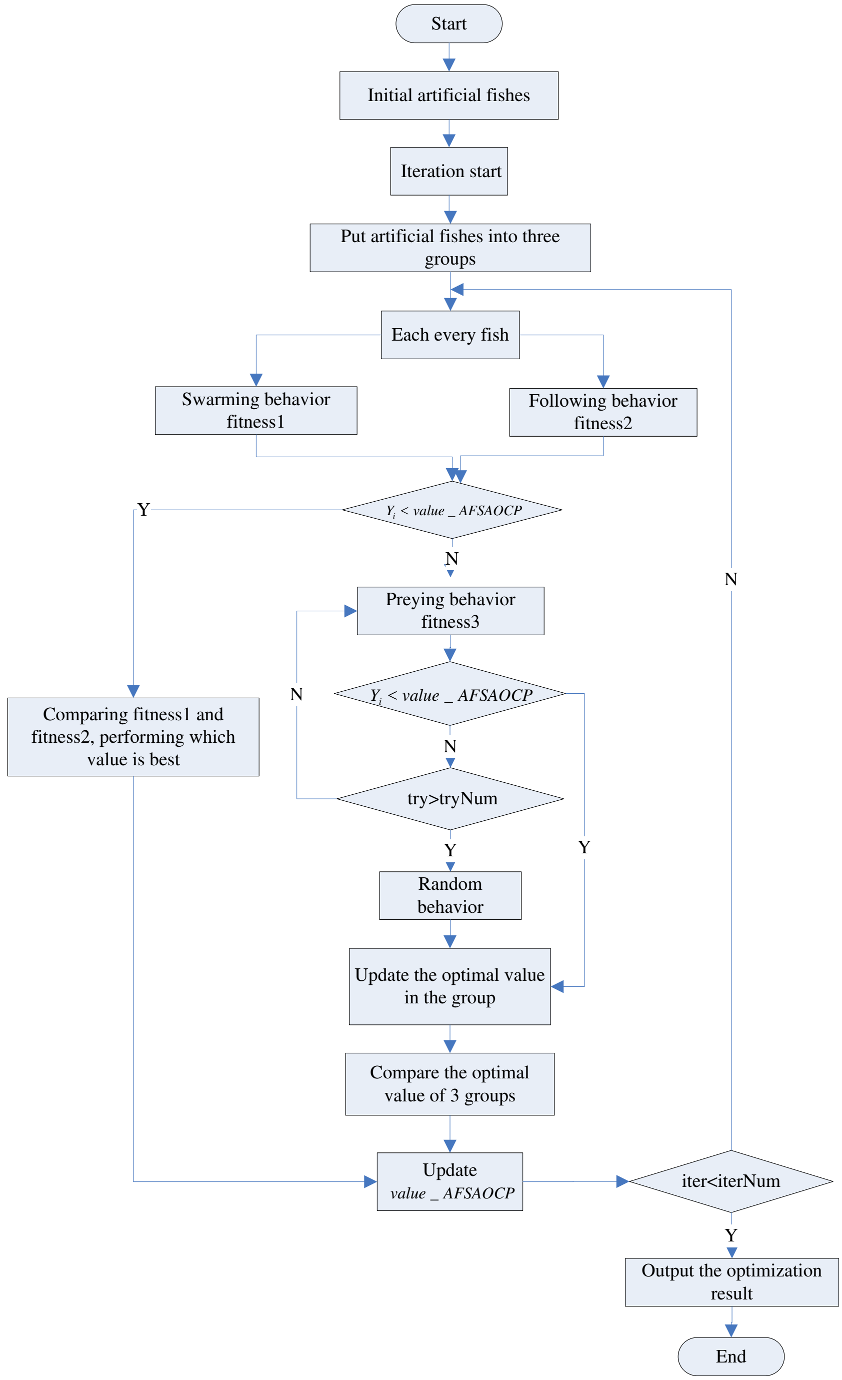


Fig. 10 Flow chart of AFSAOCP

degree factor δ and the angle of artificial fishes and ocean α, β. This section analyses the influence of various parameters on the algorithm precision in the algorithm, so that the subsequent simulation experiments can be more effectively carried out than before.

This section forms different parameter values using test function experiment and comparison, which used test function for *Rosenbrock*, specific expression such as (5-1).

$$f(x) = \sum_{i=1}^{N-1} (100(x_i^2 - x_{i+1}^2)^2 + (x_i - 1)^2) \tag{5-1}$$

(1) The number of artificial fishes $FishNum$

Table 1 Pseudo code of the AFSAOCP

```
For i = 1 to FishNum
  Fish(i,:) = xmin+(xmax-xmin).*rand(1,N)
EndFor
For j = 1 to iterNum
  iter = iter + 1
  If(iter > 1)
    b_value(iter) = b_value(iter-1)
  Else b_fish = zeros(1,N)
  End if
 best_af = xmin+(xmax-xmin).*rand(1,N)
 For i = 1to 1/3FishNum
   Fish(i,:) = evaluate(Fish(i,:))
   If (foodconsistence(Fish(i,:),smax) < foodconsistence(best_af1,smax))
     best_af1 = Fish(i,:)
   End if
   If(foodconsistence(Fish(i,:),smax) < b_value1(iter))
     b_value1(iter) = foodconsistence(Fish(i,:),smax)
   End if
 End for
 For j = 1/3FishNum +1 to 2/3FishNum
   Fish(j,:) = evaluate(Fish(j,:))
   If (foodconsistence(Fish(j,:),smax) < foodconsistence(best_af2,smax))
     best_af2 = Fish(j,:)
   End if
   If(foodconsistence(Fish(j,:),smax) < b_value2(iter))
     b_value2(iter) = foodconsistence(Fish(j,:),smax)
   End if
 End for
 For k = 2/3FishNum +1 to FishNum
   Fish(k,:) = evaluate(Fish(k,:))
   If (foodconsistence(Fish(k,:),smax) < foodconsistence(best_af3,smax))
     best_af3 = Fish(k,:)
   End if
   If(foodconsistence(Fish(k,:),smax) < b_value3(iter))
     b_value3(iter) = foodconsistence(Fish(k,:),smax)
   End if
 End for
 If (b_value1(iter)<b_value2(iter))
   min = b_value1(iter)
 Else min = b_value2(iter)
 End if
 If (min<b_value(iter))
   b_value(iter) = min
 End if
End for
```

The size of the artificial fish decided the convergence speed of the algorithm. This experiment respectively sets the number of artificial fish to 10, 30, 60, 100, 200 and the experimental results are shown in Table 2.

According to Table 1, we can see that when the scale is small, the algorithm requires a short period of time, but the optimal value is poorer; the scale is large, the optimal value of the algorithm is better, but at the same time the execution time is too long. Therefore, the size of the artificial fishes should be moderate that the algorithm not only has the better optimal value, but also can make the execution time as short as possible.

(2) The artificial fishes' view $Visual$

Behavior evaluation in view of artificial fishes has a bigger impact and it will affect the convergence of the algorithm. We know the fish's field of vision is limited, so this

Table 2 The influence of different artificial fish scale for solution

$FishNum$	10	30	60	100	200
avg	1.04E+02	4.13E+00	7.60E-01	6.06E-02	3.64E-01
avg time	5.712724	17.145010	32.626828	93.661618	130.434474

Table 3 The influence of different view for solution

$Visual$	0.1	1	10	30	50
avg	3.41E+04	2.25E+03	1.52E+03	1.95E+00	2.30E+00
avg time	5.083687	25.461808	27.344758	34.273285	34.495084

Table 4 The influence of different crowding degree factor for solution

δ	0.618	5	10	15	20
avg time	8.06E+00	1.66E+00	5.48E+00	7.05E-01	2.98E+00
time	35.012018	35.143308	32.183541	31.960114	31.908152

Table 6 The influence of mobile angle β for solution

β	90°	120°	145°	160°	180°
avg	8.14E+00	9.12E-01	6.48E-01	7.05E-01	2.98E+00
avg time	36.012352	34.99378	35.135123	35.35211	38.35829

experiment set of artificial fishes' view of 0.1, 1, 10, 30, 50 and experimental results are shown in Table 3.

According to Table 2, when the view of the artificial fish is small, the time required for the algorithm to run is shorter, but the optimal value obtained is the worst. With the increase of test field of vision, the average time of the algorithm is getting longer and longer, and the average optimal value of the algorithm is gradually tending to be ideal. The reason is that when the sight is small, the main behavior of artificial fishes are the preying behavior and random behavior, whereas if the visual field is larger, the main behavior of artificial fishes are the following behavior and swarming behavior. Overall, under the condition of the same time, it is easier to find the global optimal value and convergence of the artificial fish with larger view.

(3) Crowding degree factor δ

Crowding degree factor is one of the standards that is being used to determine whether artificial fishes perform the behavior. This experiment set crowding degree factor to 0.1, 0.618, 5, 10, 15, and experimental results are shown in Table 4.

According to Table 4, the crowding degree factor influence on the speed of the algorithm implementation is not big, so crowding degree factor should be to achieve the optimal value is closer to the ideal.

(4) The angle of artificial fishes and ocean α, β

If there is an angle between the direction of the current, it will affect the speed of artificial fishes. α, β expresses the angle of artificial fishes and ocean.

According to Table 5, the angle α influence on the speed of the algorithm implementation is not big, so crowding degree factor should be to achieve the optimal value is closer to the ideal.

According to Table 6 , the angle β influence on the speed of the algorithm implementation is not big, so crowding degree factor should be to achieve the optimal value is closer to the ideal.

Table 5 The influence of mobile angle α for solution

α	15°	30°	45°	60°	90°
avg	6.16E+00	1.78E+00	5.48E-01	7.05E-01	8.98E+00
avg time	36.274839	34.03712	34.182453	35.24216	37.28710

5.2 Description of benchmark functions

All the definitions of the benchmark functions are given according to the number in Table 7.

1) $f_1 = \sum_{i=1}^{N}(x_i^2 - 10\cos(2\pi x_i) + 10)$ *Rastrigin* is a non-convex function. It is highly multimodal and valuated on the hypercube$x_i \in [-10, 10]$, for all $i = 1, ..., N$.
2) $f_2(x) = -0.0001(|\sin(x_1)\sin(x_2)\exp(|100 - \frac{\sqrt{x_1^2+x_2^2}}{\pi}|)| + 1)^{0.1}$, *Cross-in-Tray* function has multiple global minima. It is usually evaluated on the square $x_i \in [-10, 10]$, for all $i = 1, 2$.
3) $f_3(x) = 0.5 + \frac{\sin^2(x_1^2-x_2^2)-0.5}{[1+0.001(x_1^2+x_2^2)]^2}$, *Schaffer* function N.2 is shown on a smaller input domain in the second plot to show detail. It is usually evaluated on the square $x_i \in [-100, 100]$, for all $i = 1, 2$.
4) $f_4 = -\frac{1+\cos(12\sqrt{x_1^2+x_2^2})}{0.5(x_1^2+x_2^2)+2}$, *Drop-Wave* function is multimodal and highly complex. It is usually evaluated on the square $x_i \in [-5.12, 5.12]$, for all $i = 1, 2$.
5) $f_5 = -|\sin(x_1)\cos(x_2)\exp(|1 - \frac{\sqrt{x_1^2+x_2^2}}{\pi}|)|$, *Holder* Table function has many local minima, with four global minima. It is usually evaluated on the square $x_i \in [-10, 10]$, for all $i = 1, 2$.
6) $f_6 = \sum_{i=1}^{N} x_i^2$ *Sphere* function is usually evaluated on the hypercube $x_i \in [-10, 10]$, for all $i = 1, ..., N$.
7) $f_7(x) = \sin^2(\pi\omega_1) + \sum_{i=1}^{N-1}(\omega_i - 1)[1 + 10\sin^2(\pi\omega_i - 1)] + (\omega_N - 1)^2[1 + \sin^2(2\pi\omega_N)]$, Levy function is usually evaluated on the hypercube $x_i \in [-10, 10]$, for all $i = 1, ..., N$.
8) $f_8(x) = \frac{1}{4000}\sum_{i=1}^{N} x^2 - \prod_{i=1}^{N} con(\frac{x_i}{\sqrt{i}}) + 1$, *Griewank* function is usually evaluated on the hypercube $x_i \in [-100, 100]$, for all $i = 1, ..., N$.

Table 7 Dimensions, search ranges, and brief description of test functions

Func. #	brief descriptions	Uni/Multi	Dimension	Range	Optimums
$f_1(x)$	Rastrigin	Multi	10	[-10,10]	0
$f_2(x)$	Cross-in-Tray	Multi	2	[-10,10]	0
$f_3(x)$	Schaffer	Multi	2	[-100,100]	0
$f_4(x)$	Drop-Wave	Multi	2	[-5.12,5.12]	-1
$f_5(x)$	Holder Table	Multi	2	[-10,10]	-19.2085
$f_6(x)$	Sphere	Uni	10	[-10,10]	0
$f_7(x)$	Levy	Uni	10	[-10,10]	0
$f_8(x)$	Griewank	Multi	10	[-10,10]	0
$f_9(x)$	Rosenbrock	Uni	10	[-100,100]	0
$f_{10}(x)$	Schwefel Problem1.2	Uni	10	[-100,100]	0

9) $f_9(x) = \sum_{i=1}^{N-1} (100(x_i^2 - x_{i+1}^2)^2 + (x_i - 1)^2)$, *Rosenbrock* function is usually evaluated on the hypercube $x_i \in [-100, 100]$, for all $i = 1, ..., N$.

10) $f_{10}(x) = \sum_{i=1}^{N} (\sum_{j}^{i} x_i)^2 + shifting$, and $shifting = 10$, *Schwefel* Problem1.2 function is usually evaluated on the hypercube $x_i \in [-100, 100]$, for all $i = 1, ..., N$.

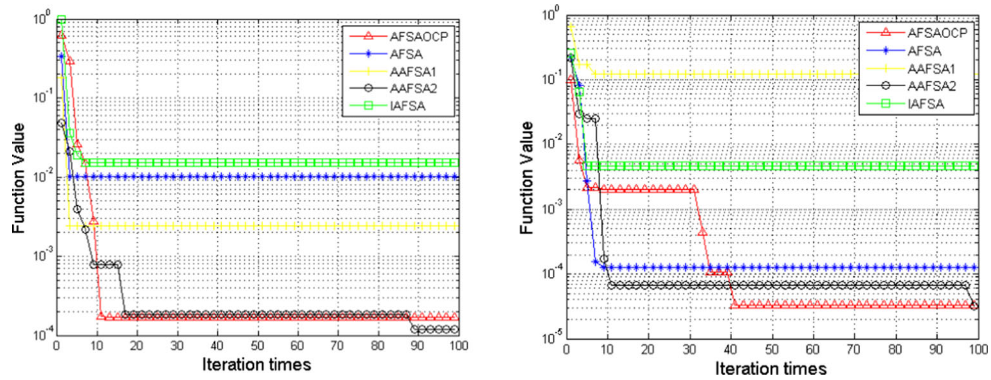
5.3 Experimental setting and parameterization

To be fair, the initial conditions of each algorithm are consistent. The population size is set to be 60. The size of the subgroup is set to 20. The 10 dimensional test function experiments are running in $f_7(x)$, $f_8(x)$, $f_9(x)$, $f_{10}(x)$, others are running in the 2 dimensional test function experiments, and in order to demonstrate their general performances that each function id running independently for 25 times.

Table 8 *Avg* and *std.* different of test functions about nine algorithms

Function	Indicator	AFSAOCP	AFSA	AAFSA1	AAFSA2	IAFSA
$f_1(x)$	avg	1.58E-03	1.94E-02	1.26E-02	1.37E-02	4.26E-03
	std	5.96E-01	2.31E+00	1.63E-01	2.80E+00	9.02E-01
$f_2(x)$	avg	3.14E-09	9.42E-07	6.49E-06	1.86E-07	7.58E-05
	std	8.66E-02	8.86E-03	5.53E-02	3.09E-02	2.54E-02
$f_3(x)$	avg 5.82E-05	1.15E-03	1.59E-03	3.66E-04	8.82E-04	
	std	4.79E-02	2.98E-02	1.20E-02	3.41E-02	2.19E-02
$f_4(x)$	avg	-9.36E-03	-9.53E-03	-9.97E-03	-1.17E-02	-9.36E-03
	std	4.45E-02	4.19E-02	1.34E-02	3.70E-02	3.05E-02
$f_5(x)$	avg	-1.92E-01	-1.82E-01	-1.92E-01	-1.86E-01	-1.92E-01
	std	2.63E+00	4.77E+00	2.19E+00	5.78E+00	1.42E+00
$f_6(x)$	avg	6.14E-04	6.27E-03	1.19E-02	1.11E-02	5.71E-03
	std	1.27E-01	8.48E-02	1.89E-02	1.66E+00	2.09E-01
$f_7(x)$	avg	2.71E-03	8.97E-03	1.48E-01	1.03E-02	4.26E-03
	std	1.30E+00	1.79E+00	1.63E-01	2.09E+00	9.47E-01
$f_8(x)$	avg	3.97E-03	9.08E-03	9.18E-03	1.13E-02	9.65E-03
	std	5.03E-02	3.86E-02	6.65E-03	3.05E-02	9.24E-03
$f_9(x)$	avg	1.02E-02	4.90E+00	8.35E+01	3.41E+00	4.01E+00
	std	1.53E+02	7.08E+02	4.27E+03	1.27E+03	3.84E+02
$f_{10}(x)$	avg	9.76E-04	1.75E-03	8.44E-02	1.63E-02	1.46E-02
	std	2.40E-01	3.30E-01	1.90E+00	1.72E+00	7.13E-02

Fig. 11 The convergence curve of the $f_1(x)$ to $f_2(x)$ test functions in different algorithms



(a) $f_1(x)$ (b) $f_2(x)$

Experimental environment configuration: Operation system is Windows 7; Minimum memory is 4G; Processor Type is Intel-Core-i3; Development tools & version is Matlab-R2012a.

5.4 Computational results and discussion

For each test function, Table 8 shows the comparison results of AFSA, AAFSA1, AAFSA1, IAFSA and AFSAOCP. In this table *avg* and *std,* respectively represent the optimal value and standard deviation after 25 independent experiments.

Figures 11, 12, 13, 14 and 15 illustrate the detailed convergence curves of AFSA, AAFSA1, AAFSA1, IAFSA and AFSAOCP for the 10 benchmark functions, which were drawn by using the average value of the 25 runs.

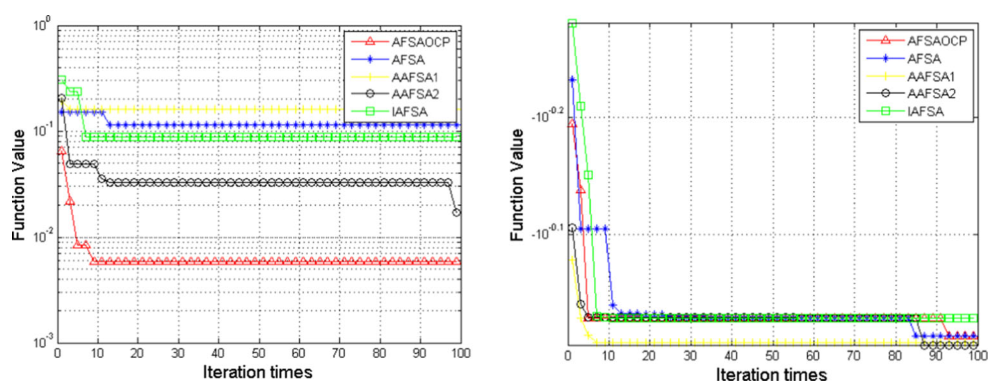
Figures 11 to 15 and Table 8 show that the convergence rate and accuracy of AFSAOCP are much better than other AF algorithms with the same dimension in solving great majority benchmark functions, such as $f_2(x)$, $f_3(x)$, $f_5(x)$, $f_6(x)$, $f_8(x)$, $f_9(x)$, $f_{10}(x)$. That is AFSAOCP has a more powerful global search capability and faster convergence speed than AFSA, AAFSA1, AAFSA2 and IAFSA but its performance is not as good as $f_1(x)$ and $f_4(x)$. In the two functions, we can see that AAFSA2 is best among other algorithms. The cause of this result may be that the algorithm introduces a new behavior: swallowing behavior. The fishes in the process of iteration will gradually appear the worst fish, which will affect the convergence rate of the whole fishes. Therefore, swallowing behavior can eliminate the worst fish, and effectively avoid the worst fish affect on the overall effect of the algorithm.

On the basis of precision guarantee, we count the time required for different algorithms to get results. Specific data is shown in Table 9 and we line out the time in bold which is the shortest.

According to the data on Table 9, AFSAOCP takes the shortest time in benchmark functions test, except in $f_3(x)$ and $f_4(x)$, where these functions are relatively simple and low-dimensional, and performs worst in $f_6(x)$.

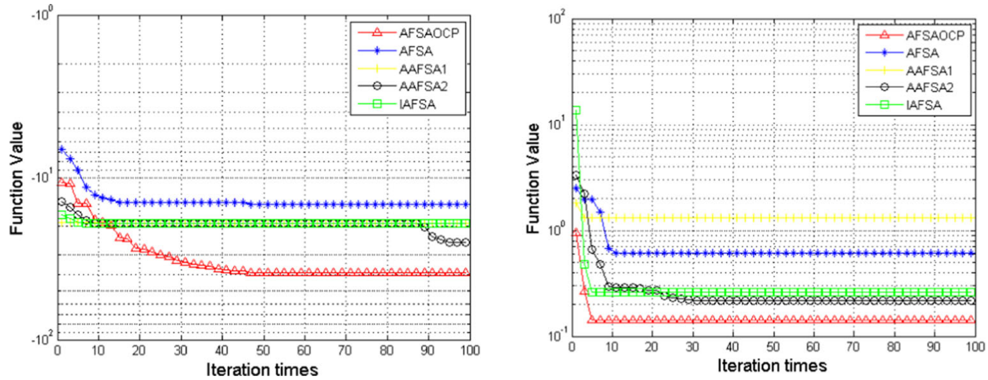
To compare AFSAOCP with different algorithms, we select some algorithms which have come from our team's

Fig. 12 The convergence curve of the $f_3(x)$ to $f_4(x)$ test functions in different algorithms



(a) $f_3(x)$ (b) $f_4(x)$

Fig. 13 The convergence curve of the $f_5(x)$ to $f_6(x)$ test functions in different algorithms



(a) $f_5(x)$ (b) $f_6(x)$

research results, this is Ref. [23]. Specific data is shown in Table 10.

6 Convergence analysis of the AFSAOCP

In this section, we analyze the convergence of AFSAOCP algorithm. Firstly, the values of the parameters used in the analysis are given for details.

- $X_S - -$ a set of All artificial fishes;
- $F(X)$–A certain kind of food concentration;
- X_S^i–non-empty subsets of X_S;
- $X^{i,j}$ $- -$position information of j-th Artificial Fish inX_S^i;
- $X^{i,j} \rightarrow X^{k,l}$– state transition probability$p_{ij,kl}$;
- $p_{ij,k}$– state transition probability of any Artificial Fish from$X^{i,j}$to X_S^k;
- $p_{i,k}$– state transition probability of any Artificial Fish fromX_S^itoX_S^k;
- Then $p_{ij,k} = \sum_{l=1}^{|X_S^k|} p_{ij,kl}$, $\sum_{k=1}^{F} p_{ij,k} = 1$, $p_{i,k} \geq p_{ij,k}$

Lemma *In the artificial fish algorithm,* $\forall X^{i,j} \in X_S^i, i = 1, 2, ..., F, j = 1, 2, ..., X_S^i$*, there are constraint of formula* (6-1) *and* (6-2)*:*

$$\forall k > i, \; p_{i,k} = 0 \tag{6-1}$$

$$\exists k < i, \; p_{i,k} > 0 \tag{6-2}$$

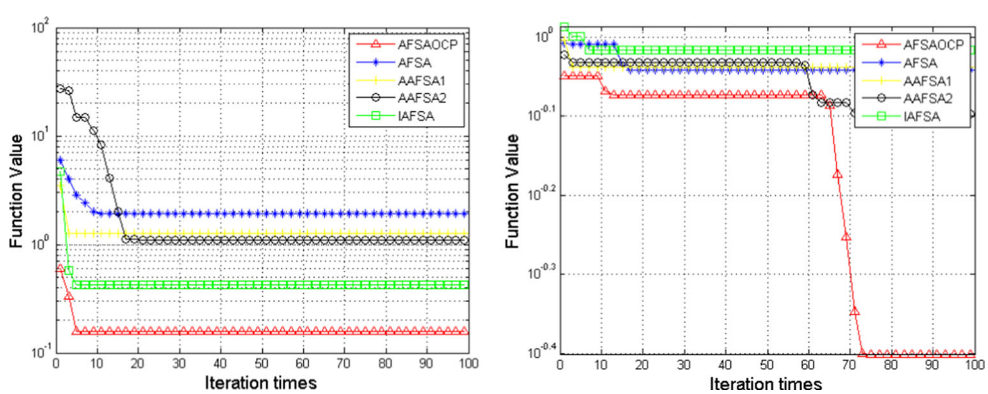
Proof X^t is after the t-th iteration of Artificial Fish which has gotten the best food concentration, it is called $BestFish^t = X^*$, and then $F(BestFish^t) = F_i$. From the update value after each iteration, we can see:

$$F(X^{t+1}) \leq F(X^t) \Rightarrow \forall k > i, \; p_{ij,kl} = 0 \Rightarrow \forall k > i,$$

$$p_{ij,k} = \sum_{l=1}^{|X_S^i|} p_{i,j,kl} = 0 \Rightarrow \forall k > i, \; p_{i,k} = 0$$

AF will choose one behavior that according to the results of the evaluation of foraging behavior, and the probability of three behaviors should be $p_{swarm}, p_{follow}, p_{prey} \geq 0$. The AF eventually will choose one correct behavior in

Fig. 14 The convergence curve of the $f_7(x)$ to $f_8(x)$ test functions in different algorithms



(a) $f_7(x)$ (b) $f_8(x)$

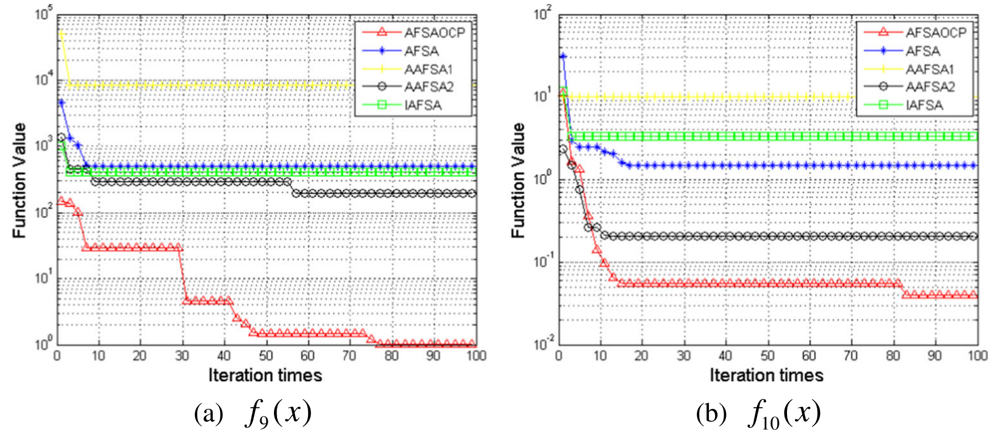


Fig. 15 The convergence curve of the $f_9(x)$ to $f_{10}(x)$ test functions in different algorithms

AFSAOCP, so $p_{swarm} + p_{follow} + p_{prey} = 1$, and then $\exists k < i$, $p_{i,k} > 0$ ⌈ proven ⌋ □

Theorem *AFSAOCP has global convergence.*

Proof the theorem will use the stability of the random matrix reasoning. For each $X_S^i, i = 1, 2, ..., F$, it can be seen as a state of finite markov chain. According to (51), transfer matrix of the markov chain is:

$$P = \begin{bmatrix} p_{1,1} & 0 & \dots & 0 \\ p_{2.1} & & \dots & 0 \\ \vdots & \vdots & & \vdots \\ p_{F,1} & p_{F,2} & \cdots & p_{F,F} \end{bmatrix} = \begin{bmatrix} C & 0 \\ R & T \end{bmatrix}.$$

According the (6-2), we can infer that:

$$p_{2,1} > 0, R = (p_{2,1}, p_{3,1}, ..., p_{F,1})^T,$$

$$C = (p_{1,1}) \neq 0, T = \begin{bmatrix} p_{2,2} & \cdots & 0 \\ \vdots & & \vdots \\ p_{F,2} & \dots & p_{F,F} \end{bmatrix} \neq 0,$$

Therefore, P is the random matrix that can be classified, and meets the conditions of the theorem, so there is

$$P^{\infty} = \lim_{k\to\infty} \begin{bmatrix} C^k & \dots & 0 \\ \sum_{i=1}^{k-1} T^i R C^{k-i} & \dots & T^k \end{bmatrix}$$
$$= \begin{bmatrix} C^{\infty} & \dots & 0 \\ R^{\infty} & \dots & T \end{bmatrix}, C^{\infty} = (1), R^{\infty} = (1, 1, ..., 1)^T,$$

And then,

$$P^{\infty} = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 1 & 0 & \dots & 0 \end{bmatrix}.$$

The matrix is stable random matrix, so $\lim_{t\to\infty} p\{F(X^t) = F_{Best}\} = 1$ F_{best} is the optimal objective function value, namely $F_{best} = f(X^t)$. Therefore, AFSAOCP has global convergence, prove to complete. ⌈ proven ⌋ □

Table 9 The mean time of different algorithms

#	AFSAOCP	AFSA	AAFSA1	AAFSA2	IAFSA
1	35.4635	38.5535	41.6838	35.7237	40.8809
2	29.7476	29.8377	33.0537	29.8280	31.5170
3	23.0279	22.6242	33.5681	31.5968	32.9383
4	26.9454	27.1657	30.0801	25.7277	28.5732
5	20.4825	21.9112	27.3724	21.4130	32.4824
6	45.7415	41.7542	43.0991	41.2156	42.0789
7	38.7957	39.0761	40.2842	41.6629	40.6588
8	31.0065	33.4853	36.6641	31.2848	33.5483
9	29.5249	31.3681	31.4547	29.9404	29.9539
10	30.9985	32.8960	35.2120	34.4783	32.4373

Table 10 The comparison results of function optimization

Function	Dim	Algorithm	Avg	Std
$f_8(x)=\frac{1}{4000}\sum_{i=1}^{N}x^2-\prod_{i=1}^{N}con(\frac{x_i}{\sqrt{i}})+1$	30	MSFLACM	7.39E-03	2.46E-03
		SBDE	9.10E-06	7.44E-07
		DE	8.90E-06	7.39E-07
		SFLSDE	1.57E-04	1.03E-03
		OBDE	4.78E-03	2.20E-03
		jDE	1.58E-03	3.29E-04
		AFSAOCP	1.11E-02	5.21E-02
$f_9(x)=\sum_{i=1}^{N-1}(100(x_i^2-x_{i+1}^2)^2+(x_i-1)^2)$	30	MSFLACM	4.93E-03	0.00E-00
		SBDE	4.16E+01	2.63E+01
		DE	3.05E+01	1.87E+01
		SFLSDE	2.34E+01	2.82E+01
		OBDE	2.97E+01	3.53E+01
		jDE	2.49E+01	3.65E+01
		AFSAOCP	6.91E+01	1.42E+03

7 Conclusion

Optimization problems exist in many areas, A novel Artificial Fish Swarm Optimization Algorithm Aided by Ocean Current Power (AFSAOCP) is put forward in this paper. The main idea has the following two points: introduce the current ideas and grouping evolution. The main idea to consider that the ocean current always influences fishes' speed, including positive influence and negative influence. The experimental studies in this paper show that the proposed AFSAOCP algorithm improves the existing performance of other algorithms compared to some same benchmarks.

The AFSAOCP also has a very big development space. In the future, it is very interesting to discuss that different fish have different speeds and the problem angle of fish swimming. In addition, it is very meaningful to change the mutation strategy of the fish's $Step$. We believe that the dynamic value of $Step$ or other fixed value would make the AFSAOCP better.

Acknowledgments Financial supports from the National Natural Science Foundation of China (No. 61572074) and the 2012 Ladder Plan Project of Beijing Key Laboratory of Knowledge Engineering for Materials Science (No.Z121101002812005) are highly appreciated.

References

1. Martens D, Baesens B, Fawcett. T (2011) Editorial survey: swarm intelligence for data mining. Machine learning, pp 1–42
2. Colomi A, Dorigo M, Maniezzo V (1991) Distributed optimization by ant colonies. In: Proceedings of the first European Conference on artificial life. Paris, France
3. Eberhrt R, Kennedy J (1995) A new optimizer using particle swarm theory. In: Proceeding of the 6th international symposium on micro machine and human science, pp 39–43
4. Li X, Shao Z, Qian J (2002) An optimizing method based on autonomous animates: fishswarm algorithm. Chinese Journal of systems engineering-theory & practice, pp 32–38
5. Li X (2003) A new intelligent optimization algorithm—artificial fish swarm algorithm. Zhejiang University
6. Wang H, Zhao X, Wang K, Xia K, Tu X (2013) Cooperative velocity updating model based particle swarm optimization. Applied intelligence, pp 322–342
7. Jiang M, Zhu K (2011) Multiobjective optimization by artificial fish swarm algorithm. In: Proceedings of the IEEE international conference on computer science and automation engineering(CSAE), pp 506–511
8. Liu J, Chen X, Liu Q, Sun J (2013) Prediction of satellite clock errors using ls-svm optimized by improved artificial fish swarm algorithm. Signal processing, communication and computing(ICSPCC), pp 1–5
9. Yazdani D, Toosi AN, Meybodi MR (2009) Fuzzy adaptive artificial fish swarm algorithm. International joint conference on computational sciences and optimization, pp 317–321
10. Yan W, Liguo Z (2011) Method of Bayesian network parameter learning bse on improve artificial fish swarm algorithm. Communications in computer and information science, pp 508–513
11. Zhou Y, Huang H, Zhang J (2011) Hybrid artificial fish swarm algorithm for solving 3-conditioned linear systems of equations. International conference on cloud computing and intelligence systems, pp 656–661
12. Liu S, Han Y, Ouyang Y, Li Q (2014) Multi-objective reactive power optimization by modified artificial fish swarm algorithm in ieee 57-bus power system. Power and energy engineering conference (APPEEC), pp 1–5
13. Liu Y (2009) Artificial fish swarm algorithm applicates in wireless aensor network (wsn) by optimization problems. Shandong University
14. Cheng YM, Jiang MY, Yuan DF (2009) Novel clusting algorithms based on improved artificial fish swarm algorithm. In: Proceedings of the 6th international conference on fuzzy systems and knowledge discovery, pp 141–145

15. Zhang C, Zhang F, Li F, Wu H (2014) Improved artificial fish swarm algorithm. Industrial electronics and applications(ICIEA), pp 748–753
16. Liang X (2013) Precise underwater localization based on ocean current information. Dissertation Submitted to Shanghai Jiao Tong University for the Degree of Master
17. Yang Z (2004) Marine geology. Shandong education publishing house
18. Huang X (2010) The knowledge of geography. The encyclopedia of China publishing house
19. Wu X (2009) The trace of water ShenYang publishing house, pp 92–95
20. Zhu Y-F, Tang X-M (2010) Overview of swarm intelligence. International conference on computer application and system modeling, pp 400–402
21. Yang X, He X (2015) Swarm intelligence and evolutionary computation: overview and analysis. Recent advances in swarm intelligence and evolutionary computation, pp 1–23
22. Neshat M, Sepidnam G, Sargolzaei M, Toosi AN (2014) Artificial fish swarm algorithm: a survey of the state-of-the-art, hybridization, combinational and indicative applications. Artificial intelligence review, pp 965–997
23. Wang H, Zhang K, Tu X (2015) A mnemonic shuffled frog leaping algorithm with cooperation and mutation. Applied intelligence

Hong-Bo Wang received the doctor degree in Computer Application Technology from the University of Science and Technology Beijing, China, in 2007. After graduation, he joined the School of Computer and Communication Engineering, USTB, where he is an Associate Professor and M.S. candidate supervisor; his main research interests include Natural Inspired Computing and Intelligence Optimization Scheduling. (Email: foreverwhb@ustb.edu.cn)

Cheng-Cheng Fan is a M.S. candidate at the Department of Computer Science and Technology, School of Computer and Communication Engineering, University of Science and Technology Beijing. Her research area is Artificial Fish Swarm Optimization Algorithm. (Email: fanchengcheng110@qq.com)

Xu-Yan Tu (ACM senior member) is a professor and doctor supervisor at the School of Computer and Communication Engineering, University of Science and Technology Beijing. His research areas are Artificial Intelligence and Artificial Life, Large Scale System Control and Intelligent Management. (Email: tuxuyan@126.com)