# An Improvement of AFSA in Global Search with Scout Swarms

**Tian Guo, Heji Zhao**
Computer Science and Technology of Shandong University, Jinan, Shandong Province, China
SpiderTian@163.com, hejizh@sdu.edu.cn

**Abstract** - Since the emergence of AFSA (Artificial Fish Swarm Algorithm), because of a lot of good features in solving complex optimization problems, AFSA has been studied by scholars from domestic and overseas, however it also has some shortcomings such as high time complexity, lack of balance between global search and local search. In recent years lots of researchers have attempted to improve this algorithm. In this paper, a new improvement will be introduced to solve its ability of balance between global search and local search, and its name is Scout-AFSA (Scout-Artificial Fish Swarm Algorithm). And the final experiment results show that Scout-AFSA does work better than AFSA in global search**.**

Index Terms - AFSI (Artificial Fish Swarm Intelligence), Scout-AFSA, Global Search.

## I. Introduction

At first, we should have an entire look at how nature fish swarm works, and it will help us have a better understanding of AFSA.

In the nature world, almost all the things (living and non-living things) are going with the different performances but with the same law, maybe the only difference between them is the different visual opinion used by ourselves in our mind.

On the earth, there are many kinds of living beings and they usually swarm and form a population to protect themselves from their enemies. In this way, they can have more chances to survive. However there are some differences between these populations, some animals have one leader in the swarm, like monkeys, lions, wolves and elephants, but some animals don’t have their leaders in their populations, like fish, ant, pigeon and so on(although the ants have a queen, the queen can do no effect to the swarm but breeding), mostly the members of this kind of swarm are almost the same, like their performance features, looks and so on, and they are always a little of mechanical. But those simple individuals performing mechanically can form a swarm with high intelligence that can give us human beings a big surprise.

In order to solve some complex problems, we should turn to nature for help, because fish can survive even though the food is hard to find and the natural enemies are always ready to hunt them. AFSA simulates the elements inspired by the social behaviors of natural swarms like fish swarm. Complex optimization is the main domain which AFSA has been widely applied in and it is currently a major research topic.

AFSA is a kind of evolutionary computation methods, which was first proposed in 2002. It has many good features such as independence from gradient information of the objective function, what’s more, it has the ability to solve nonlinear complex high dimensional problems. It has high convergence speed and needs few parameters to be adjusted. A set of randomly generated potential solutions initializes the system firstly and then iteratively performs the algorithm to find the optimum. In fact an artificial fish is an atom entity of a thing with its own data and a series of behaviors. It can be seen as a cell of a living being, with its simple and single behaviors the living being can display high intelligence. In fact, I think swarm intelligence is the final solution to AI, just like the existing computer operation system which in nature is a kind of swarm intelligence, so do a country, the world and the earth. The artificial fish accepts the information of the environment through sense organs and do responding behaviors.

This paper is structured as follows. Part 2 reviews the introduction of the general AFSA in detail. Part 3 gives a new improvement of AFSA, called Scout-AFSA. Part 4 concludes.

## II. The Introduction of AFSA

If we want to make a simulation of the natural fish swarm, we should know how the real fish swarm works and its mechanism.

In nature, fish always swarm and form a group to protect themselves from enemies, at the same time they can find more food more easily in this way and other advantages for the survival of this species. They can find more nutritious area by individual search or following after other fish, because the area with much more fish generally has more food. Through imitating the behaviors of real fish such as preying, swarming and following, AFSA can find the optimum. We just make the solution space as the environment of the Artificial Fish, and the AF’s next movement only depends on their current state and the local environmental state. At the same time, the AF will influent each other’s behavior and the environment’s state. In a water area, the fish usually have those behaviors like preying, swarming, following and stochastic searching. As usual, fish will swim to the direction of the area which has more nutrition. At the same time, the fish will swarm together to protect themselves from danger and they abide by established rules to prevent the companies’ environment from being too crowd, to keep the same direction with most companies and to swim to the direction of the center of its companies. If some individuals find an area with more nutrition, the others around them will arrive this area quickly just by following them. At most time those Artificial Fish are swimming stochastically in order to find the area with more food.

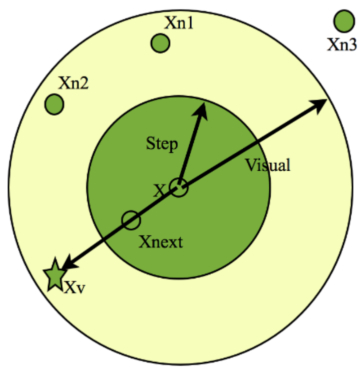


Fig. 1 Describe the overview vision of a fish.

In the figure 1 above [1], it describes the overview vision of a fish. *X* represents the current position of the fish. *Step* represents the length of one step the fish can make. *Visual* is the visual distance of the fish. *Xnext* represents its next position after its next movement. *Xv* is the visual position at some moment. *Xn1* and *Xn2* is other artificial fish in its visual distance, but *Xn3* not.

And $X=(x_1, x_2,...,x_n)$, $Xv= (x_1^v, x_2^v, . . . , x_n^v)$

$$X_i^v=x_i+Visual*rand() \qquad i=1, 2, . . . n. \tag{1}$$

$$Xnext=X+(Xv-X)/||Xv-X||*Step*rand() \tag{2}$$

In the above, *rand()* can produce random numbers between 0 and 1. And *try_number* represents the maximum try times and δ is the crowd factor (0<δ<1).

Then as following, the fish behaviors will be introduced in detail.

***Swarm:*** The fish have the habit of assembling together in the moving process. This nature will promise no individual fish which is alone without company. The group can protect an individual from enemies and other dangers. The behavior can be described as following: Let $X_i$ be the current position of the fish, $X_c$ be the center position of the swarm, $n_f$ be the number of its companies, *n* be the number of the whole fish in the swarm and *Y* be the food concentration.

If $Y_c>Y_i$ and $(n_f/n)<\delta$ then it will go forward a step to the center position:

$$X_i(t+1)=X_i(t)+(X_c-X_i(t))/||X_c-X_i(t)||*Step*rand() \tag{3}$$

Otherwise it executes the ***Prey*** behavior.

***Prey***: Let $X_j$ be a position in one AF's visual distance, and its current position is $X_i$. So the relationship between $X_i$ and $X_j$ is $X_j=X_i+Visual*rand()$. Then if $Y_j>Y_i$ in the maximum problem, the AF will go forward a step in this direction. It can be described as following:

$$X_i(t+1)=X_i(t)+(X_j-X_i(t))/||X_j-X_i(t)||*Step*rand() \tag{4}$$

Otherwise, select another position randomly and judge whether $Y_j>Y_i$, if it can't be satisfied after *try_number* times, then the fish will swim toward a step randomly. It can be described as following:

$$X_i(t+1)=X_i(t)+Visual*rand() \tag{5}$$

***Follow***: If $Y_j$ is the biggest food concentration of all the neighborhood partners of the fish, whose current position is $X_i$, and $Yj>Yi$ and $n_f/n<\delta$ then it will go forward a step to the direction of $X_j$:

$$X_i(t+1)=X_i(t)+(X_j-X_i(t))/||X_j-X_i(t)||*Step*rand() \tag{6}$$

Otherwise it will behave Prey behavior.

***Move***: Just select a state and go forward a step to the direction in its vision:

$$X_i(t+1)=X_i(t)+Visual*rand() \tag{7}$$

Now, the general AFSA has been introduced in detail. But it has some disadvantages, and some improvements have been made. In the next part, a new improvement will be introduced, and it is named as Scout-AFSA.

## III. The Improvement of AFSA (Scout-AFSA)

In the nature, the fish swarms march in the water area in order to find more food, in most cases, the swarms are working respectively. Being confined by some physical elements, they can't exchange information between each other. However, in the computer world information is the basic element and it can be transported easily, so why not import information sharing to AFSA.

So how to solve it, in fact we have solve it thousands of years ago. When an army is marching, especially when they are searching for something in an unfamiliar environment, there are a lot of chances that the main army moves for a long time and finds it wasting time in the end. So in case of this' happening, the captain of the army usually commands a number of soldiers to nose for the environment in different directions before the main army decides to move on. At most time, there is no need to take a lot of computing resources to do something uncertain, but at the same time there is a need to do those things uncertain, so we should take appropriate resource to do those things uncertain.

To realize the improvement, there are different implementation methods. Here, two implementations will be introduced in detail, and in order to introduce them clearly by separating them. Depending on different situations, they can have different efficiency and they all have their features in solving different problems, what's more, they can even be combined together.

With regard to the artificial fish swarms used in the improvement, they are no different with the artificial fish swarm created by AFSA. The only difference between AFSA and Scout-AFSA is that the former only depends on one swarm and when the only one swarm stops, the algorithm will also stop, even though it should not yet, because maybe it has fallen into local optimum. So it should be able to judge whether it should stop by itself and know what to do when it finds that it's not the right time when it stops right now.

In the two implementations, there are two different kinds of artificial fish swarm, they are main artificial fish swarm (abbreviated as Main-AFS) and vice artificial fish swarm (abbreviated as Vice-AFS), the former is used to find the optimum and the latter is used to judge whether the Main-AFS should stop and which direction the Main-AFS should move towards in the next movement.

The following figure can reflect the outline of the new improvement, Scout-AFSA.

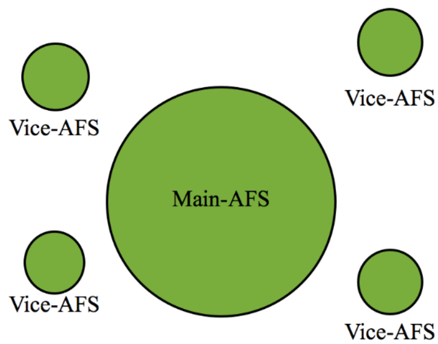


Fig. 2 Reflect the outline of Scout-AFSA

Main-AFS and Vice-AFS are all the same artificial fish swarms with the one used by AFSA and they all have the same mechanism, but they have different missions and Main-AFS' size is bigger than Vice-AFS'.

The first implementation is the same with the general AFSA at the beginning and the Main-AFS will begin to work and the running swarm is no different with the one used in AFSA. But when the Main-AFS finds the local optimum and stops, the whole algorithm will not stop and it will generate n Vice-AFS and they begin to work. When they all stop, the Main-AFS will find which Vice-AFS found the biggest nutrition area and then it will move a distance, decided by how big the biggest nutrition area is, towards the Vice-AFS who find the biggest nutrition area. Then the algorithm will go back to the beginning, but if the Main-AFS can't find a better area with more nutrition than the area found before the Main-AFS has changed its location M times, the algorithm will really stop and return the biggest that has been found ever as the global optimum.

The main steps of the first implementation are as following:

1. *Initialize one Main-AFS and n Vice-AFS artificial fish swarms (Vice-AFS's location can be in a certain range of the Main-AFS and generated randomly);*
2. *Main-AFS starts to work;*
3. *When the main swarm stops, let the vice swarms begin to work;*
4. *Through biggest optimum found by the n vice swarms, if the biggest optimum is not bigger than the current optimum found by the main swarm, then the main swarm will move to the vice swarm who found the biggest optimum and the distance how long the Main-AFS will move towards it depends on the biggest optimum, the bigger it is, the longer the distance will be.*
5. *After having try those things M times, if the main swarm can't find better optimum, then the course will stop. And the biggest optimum found by the main optimum ever will be the final global optimum.*
6. *The end.*

To have a better understanding about this algorithm, some mathematic elements are introduced into the description of this algorithm:

$X_{mc}$ represents the fish which is in the center position of Main-AFS swarm;

$X_{vc}$ represents the fish which is in the center position of Vice-AFS swarm;

$M$ represents the number how many times the Main-AFS will try before it really stops;

$B_m$ represents the best local optimum found by Main-AFS ever;

$B_v$ represents the best local optimum found by the all Vice-AFS;

$X$ represents each fish in the Main-AFS;

$C$ represents a coefficient used in the movement of the fish in Main-AFS;

At first, the Main-AFS works and then stops for a moment when it finds the local optimum (we don't know whether it's the global optimum or not), at the same time the vice swarms will begin to work;

then X=X+C(Xvc-Xmc) $B_v/B_m$

If the Main-AFS with new position can find a better area than before, the algorithm will go back to the beginning.

Otherwise, the Main-AFS with new position will start to work again, and the course will be repeated M times at most, and if it can't help the Main-AFS find a new area with more nutrition, the algorithm will stop really.

The flowchart:

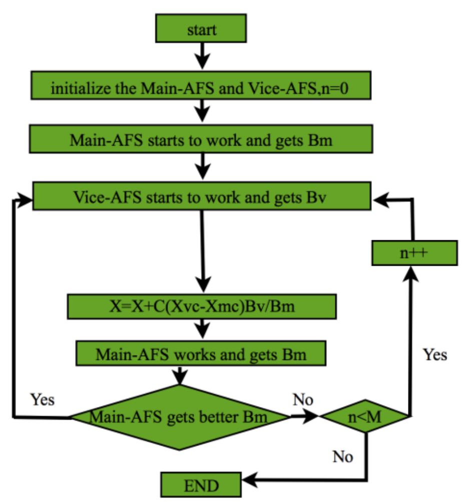


Fig.3 Flowchart of Scout-AFSA's first implementation

The second implementation is described as following. At the beginning, it is the same with the general AFSA, the Main-AFS will work until it stops and when it stops, the Vice-AFS will begin to work. The number of Vice-AFS can be various due to different situations. The size of Vice-AFS should be smaller than the Main-AFS in order to speed up the execution and save computing resource. When those Vice-AFS begin to work, the Main-AFS may have fallen into local optimum, and the Vice-AFS can search in a range, which has been decided in the initialization, around the Main-AFS. If those vice swarms find a better area with more nutrition than the current area with local optimum found by the main swarm, it implies that the main swarm has fallen into the local optimum and the next movement is that the main swarm should move and arrive at the area with more food found by those vice swarms. After this movement, the algorithm will go back to the beginning. If those vice swarms can't find a better area with more nutrition in M times, then the current optimum found by the main swarm can be regarded as the global optimum and maybe the algorithm has found the global optimum to a certain degree.

The main steps of the second implementation are as following:

1. *Initialize one Main-AFS swarm and several Vice-AFS (vice swarms' location can be in a certain range of the main swarm and generated randomly);*
2. *Main-AFS starts to work;*
3. *When the Main-AFS stops, let the vice swarms begin to work;*
4. *Through comparing the optimum found by the main swarm with the biggest optimum found by the vice swarms, the main swarm can judge whether it should go on or stops right now;*
5. *If the biggest optimum found by the vice swarms is bigger than the one found by the main swarm, then the main swarm should move and arrive at the area of the vice swarm who found the biggest optimum and the algorithm jumps back to the 2$^{nd}$ step, otherwise the vice swarms will try again to find a better area until they have tried this for M times already. After M times, if they can't find a better area with more food than the area of main swarm, then the optimum found by main swarm can be considered as the global optimum to a certain degree.*
6. *The end.*

In order to understand this algorithm more easily, it can be described with some mathematic elements:

$\boldsymbol{X_{mc}}$ represents the fish which is in the center position of Main-AFS swarm;

$\boldsymbol{X_{vc}}$ represents the fish which is in the center position of Vice-AFS swarm which found the best place;

$\boldsymbol{M}$ represents the try number how many times all the Vice-AFS swarms will work at most before the Main-AFS swarm really stops and the whole algorithm stops;

$\boldsymbol{B_m}$ represents the best local optimum found by Main-AFS;

$\boldsymbol{B_v}$ represents the best local optimum found by the all Vice-AFS in the **M** times;

$\boldsymbol{X}$ represents each fish in the Main-AFS;

In the beginning, the Main-AFS works like AFSA and then stops for a moment when it finds one local optimum, at the same time the Vice-AFS will begin to work, if $B_v>=B_m$, then $X=X+(X_{vc}-X_{mc})$, and the algorithm will go back to the beginning. If $B_v<B_m$, then Vice-AFS swarms will work again until it has tried this for M times. If in the M times the Vice-AFS swarms can find no area with more nutrition, the algorithm will stop and the optimum found by the Main-AFS swarm will be considered as the global optimum.

The algorithm flowchart:

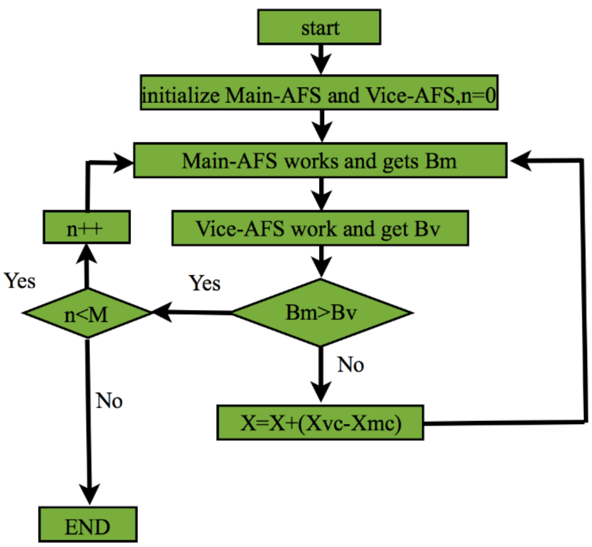


Fig.4 Flowchart of Scout-AFSA's First implementation

In order to give demonstration to the new improvement of AFSA, Scout-AFSA, an experiment has been done, in order to make a comparison between AFSA and Scout-AFSA in the capability of global search and it will be introduced in detail as following:

First, the test function should be designed to be able to test the algorithm's ability of jumping out of the local optimum and finding the global optimum. So the function is defined as following:

$$F(x,y) = \begin{cases} 30x-y & x<m,\ y<m \quad (1) \\ 30y-x & x<m,\ y\geq m \quad (2) \\ x^2-y/2 & x\geq m,\ y<m \quad (3) \\ 20y^2-500x & x\geq m,\ y\geq m \quad (4) \end{cases}$$

In the test program, AFSA and Scout-AFSA are simulated and the Scout-AFSA simulation has combined the first and the second implementation together. Here are some screen shots to show the results visually. In the test, m=30, 0<x<60, 0<y<60. And in the figure 5, 7, 9, 11, the left photo is the start status of AFSA and Scout-AFSA, the middle photo is the end status of AFSA, the right photo is the end status of Scout-AFSA. The red area represents the artificial fish, the other represents the environment and the yellower area represents the area with more nutrition.

We will conduct this test in four different cases of the start position of the main swarm of AFSA and Scout-AFSA.

1. *In the case of that the main fish swarm begins at the area of function (1)*

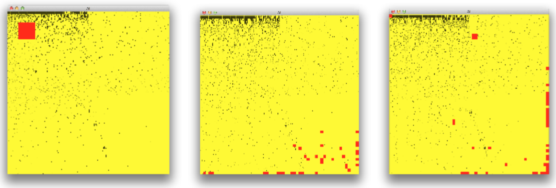

Figure 5. Fish swarm begins at the up-left area

| times | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| AFSA | 44280 | 42780 | 45280 | 41780 | 46280 |
| Scout | 52280 | 50780 | 52280 | 52280 | 52280 |

Figure 6. Data results of AFSA and Scout-AFSA in Figure 5

2. *In the case of that the main fish swarm begins at the area of function (2)*

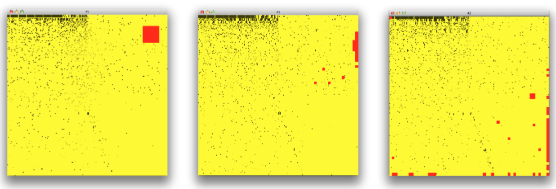

Figure 7. Fish swarm begins at the up-right area

| times | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| AFSA | 1733 | 1733 | 1734 | 1734 | 1734 |
| Scout | 52280 | 44780 | 50780 | 51780 | 52280 |

Figure 8. Data results of AFSA and Scout-AFSA in Figure 7

3. *In the case of that the main fish swarm begins at the area of function (3)*

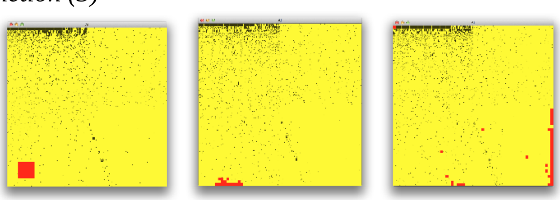

Figure 9. Fish swarm begins at the down-left area

| times | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| AFSA | 3360 | 3361 | 3360 | 3360 | 3360 |
| Scout | 47780 | 52280 | 50780 | 52280 | 50780 |

Figure 10. Data results of AFSA and Scout-AFSA in Figure 9

4. *In the case of that the main fish swarm begins at the area of function (4)*

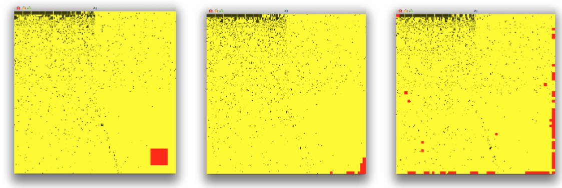

Figure 11. Fish swarm begins at the down-right area

| times | 1 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|
| AFSA | 41280 | 41280 | 42280 | 41280 | 42280 |
| Scout | 44280 | 52280 | 51780 | 52280 | 52280 |

Figure 12. Data results of AFSA and Scout-AFSA in Figure 11

Through analyzing the data above, it is obvious that AFSA can easily fall into local optimum in most conditions. However Scout-AFSA performs much better than AFSA, and it always be able to find the global optimum or desired global optimum. So Scout-AFSA, the new improvement of AFSA, does have a better capability in global search and its efficiency is good enough.

## IV. Concludes

AFSA is designed on the basis of the mechanism of natural fish swarm, it is capable to solve complex optimization problems and acceptable results have been obtained with many advantages such as high convergence speed, error tolerance and flexibility. At the same time there are some disadvantages like advanced convergence. Among those disadvantages, falling into local optimum usually can't be put up with. In most cases, we just want to get the global optimum in the first place. So the new improvement of AFSA, Scout-AFSA, is proposed in this paper to improve the capability of global search. In the test, simulating AFSA and Scout-AFSA through programming has proved that Scout-AFSA always be able to find the global optimum or desired global optimum in most cases and the general AFSA always falls into local optimum in the same cases or can't get desired global optimum.

Of course, this improvement can be optimized by using some existing improvement methods or combined with other optimization methods like Fuzzy Logic, Cellular Learning Automata and so on.

## References


[1] Mehdi Neshat, Ali Adeli, Ghodrat Sepidnam, Mehdi Sargolzaei, and Adel Najaran Toosi, "A Review of Artificial Fish Swarm Optimization Methods and Applications," *International Journal on Smart Sensing and Intelligent Systems*, vol. 5, no. 1, pp. 3-6, March 2012.

[2] Mitsuharu Higashitani, Atsushi Ishigame, Keiichiro Yasuda, "Pursuit-Escape Particle Swarm Optimization," *Tansactions on Electrical and Electronic Engineering*, IEEJ Trans 2008, vol. 3, pp. 136-142.