# Study of the Artificial Fish Swarm Algorithm for Hybrid Clustering

**Hongwei Zhao, Liwei Tian***

*School of Information Engineering*
*Shenyang University*
*21 South Wanghua Str., Dadong District, Shenyang, China*
*E-mails: zhw30@163.com, tianliwei@163.com*

**Corresponding author*

**Received: January 7, 2015** **Accepted: June 18, 2015**

**Published: June 30, 2015**

***Abstract:*** *The basic Artificial Fish Swarm (AFS) Algorithm is a new type of an heuristic swarm intelligence algorithm, but it is difficult to optimize to get high precision due to the randomness of the artificial fish behavior, which belongs to the intelligence algorithm. This paper presents an extended AFS algorithm, namely the Cooperative Artificial Fish Swarm (CAFS), which significantly improves the original AFS in solving complex optimization problems. K-medoids clustering algorithm is being used to classify data, but the approach is sensitive to the initial selection of the centers with low quality of the divided cluster. A novel hybrid clustering method based on the CAFS and K-medoids could be used for solving clustering problems. In this work, first, CAFS algorithm is used for optimizing six widely-used benchmark functions, coming up with comparative results produced by AFS and CAFS, then Particle Swarm Optimization (PSO) is studied. Second, the hybrid algorithm with K-medoids and CAFS algorithms is used for data clustering on several benchmark data sets. The performance of the hybrid algorithm based on K-medoids and CAFS is compared with AFS and CAFS algorithms on a clustering problem. The simulation results show that the proposed CAFS outperforms the other two algorithms in terms of accuracy and robustness.*

***Keywords:*** *Artificial Fish Swarm, Particle Swarm Optimization, Swarm Intelligence, Data clustering.*

## Introduction

Swarm Intelligence (SI) is an innovative artificial intelligence technique for solving complex optimization problems. In recent years, many SI algorithms have been proposed, such as Ant Colony Optimization (ACO), Particle Swarm Algorithm (PSO), Bacterial Foraging Optimization (BFO), etc. Artificial Fish Swarm (AFS) algorithm is a new swarm intelligence algorithm, imitating the behaviors of real fishes when finding a food source and sharing the information of it, which has been applied successfully to some engineering problems, such as constrained optimization problems, neural networks and clustering.

A novel Cooperative optimization model, the Artificial Fish Swarm (AFS) [17] algorithm, is designed in this paper. As a generalized neighborhood search algorithm, AFS uses swarm intelligence of biosphere to solve optimization problems, by means of heuristic search strategy, whose capacity of tracking changes rapidly gives the algorithm the ability of global optimization, because of the characteristics of global convergence itself, and the initial value can be set as fixed or random, allowing parameters to be set in a wider scope. AFSA has strong adaptability and parallelism; many behavior combinations can be selected due to its good flexibility, and it can get better optimization performance, which genetic algorithm and particle swarm optimization do not possess. This artificial intelligence model, based on biological behavior, is different from the classical pattern. Firstly, design single entity perception behavioral mechanisms, and then place a group of entities in the environment so that they can solve the problems in environment interaction [1, 2]. However, making the best reaction under the stimulation of the environment is the basic idea of AFSA. Liu and Zhou [9] proposed reducing the search field to accelerate the local search of an artificial fish individual. This optimization only took convergence speed into account rather than its quality, by making severe limitation of swarming and following behaviors of AFS, thus affecting the quality of the optimization. Tao et al. [10] introduced the *K*-means algorithm to speed up the iteration, but the performance was unstable because of many random processes in AFSA which affected the practical application of the method. Using simulated annealing algorithm to improve AFS, He and Qu [6] modified the preying behavior approach to avoid the degradation of artificial fish, with a relatively long convergence time, which was not suitable for analysis of huge data, although it overcame the shortcoming of easily falling into local minima. Combining AFS with a clustering analysis algorithm based on grid and density, Xie [14] obtained clusters automatically for the amount of *K* and applied them to arbitrarily-shaped data, achieving better parallelism, but the ultimate clustering quality was affected by the number and the size of grids, which led to some limitations [3-7].

As an important research direction of data mining, the clustering algorithm is a suitable means of classifying data for different patterns based on the different characteristics of different objects [8]. But the traditional *K*-medoids has a greater ability for a local search, for it is very sensitive to the initial cluster centers and easily falls into the local optimum. If outliers are randomly selected as the initial centers, the whole quality of classification will decline [9, 11]. AFSA is less sensitive to initial values, even in case of global optimization, which has a bad convergence and a slower iteration rate in a late period. Aiming at the advantages and disadvantages of both algorithms, this paper presents a global optimization idea to improve the *K*-medoids clustering algorithm based on AFSA, the result of which on a small data set shows that the improved algorithm obtains clear classifications and better performance [10].

This paper applies *K*-medoids and AFS algorithm to solve clustering problems which have been tested on a series of datasets, then compares the performance of CAFS on clustering with results of AFS, PSO and CAFS on the same data sets. The above data sets are provided from the UCI database [12-14].

## Optimized AFS algorithm

### *The original AFS algorithm*

Population of AFS is *N*, individual state of AF: $F = \left(f_1, f_2, ..., f_n\right)$, (where $f_i$ is optimization variables), the largest moving step is *Step*, vision is *Visual*, test time of preying behavior is *Try_number*, crowd factor is $\delta$, food consistence $Y = f\left(F\right)$ (*Y* is the value of objective function) [15].

*a. Preying behavior*

As one of the basic habits of AFS, the main principle is to find an area where there is large food concentration by using the senses of sight and taste. The current state of AF is $F_j$, then select a state $F_j$ randomly around the current location within its visual field, in the process of seeking an optimal solution, if $Y_i < Y_j$, $F_j$ will be a better state than the current one and move one step to this direction. By default choose a new state and judge again, and test *Try_number* times repeatedly. If it is unable to get a better solution, then move to a random step [16].

*b. Swarming behavior*

To ensure the survival of fish populations, AF will gather to the center of adjacent partners. While $F^i$ is still corresponding to the current state, perceive the AF number $n_f$ nearby and its central location $F_c$. If $Y_c / n_f > \delta \bullet Y_i$ is satisfied, which means the position was at a level with less congestion and more food, then step forward to $F_c$ or implement preying behavior.

*c. Following behavior*

In nature, when one or a few fish explore food, their neighbors will follow the swarm to reach the food position.

If the perception of the best state $F_j$ within the vision satisfies $Y_j / n_f > \delta \bullet Y_i$, this displays that the location is less crowded and with more food; then make a step to $F_j$ or do preying behavior. The main steps of the AFS algorithm are as follows:

1. cycle = 1
2. Initialize the food source positions $x_i$, $i$ = 1, ..., $SN$
3. Evaluate the food sources (fitness function $fit_i$)
4. Repeat
5. Preying behavior's Phase
   - For each Artificial Fish
     - Produce new food source positions $v_i$
     - Calculate the value $fit_i$
     - Apply greedy selection mechanism
   - EndFor
6. Calculate the probability values $p_i$ for the solution.
7. Swarming behaviors' Phase
   - For each Artificial Fish Swarm
     - Choose a food source depending on $p_i$
     - Produce new food source positions $v_i$
     - Calculate the value $fit_i$
     - Apply greedy selection mechanism
   - EndFor
8. Following behaviors Phase
   - If there is an Artificial Fish becoming follow
     - Then replace it with a new random source positions
9. Memorize the best solution achieved so far
10. cycle = cycle + 1
11. until cycle = Maximum Cycle Number

## *The cooperative artificial fish swarm (CAFS) algorithm*

In order to search for every best dimension for all individuals, each one's contribution to the best solution is needed. As a result, cooperative search is applied to solve the problems by AFS algorithm, and cooperative AFS algorithm emerges as required. In the CAFS algorithm, a best solution vector has been set, namely *gbest* and each component of it is the best in all populations. For *gbest* = [$g_1$, $g_2$, ..., $g_i$, ..., $g_D$], $g_i$ corresponds to the $i$-th component of the solution vector. The algorithm of the improved AFS is given below:

a) In preying behavior, when the state of a randomly selected $F_j$ does not satisfy the moving condition, it will choose random behavior, which makes it difficult to obtain high precision. AFS searches nearby the globally extreme points circuitously at the anaphase of convergence, which leads to an invalid calculation. In this paper, when preying fails, AFS chooses to move a step to a better value compared to the bulletin board records:

b) $$F_i(k+1) = F_i(k) + Step \bullet [F_{better}(k+1) - F_i(k)] \quad (1)$$

where $F_i(k + 1)$ and $F_i(k)$ denote, respectively, the current position and the next position after the movement. $F_{better}$ is the better state recorded by the bulletin board, compared to a random method, which gives the possibility for a better advance and thus jumps out of the local optima, preventing AFS in the local concussion at a standstill.

c) In AFSA, the parameter crowding factor $\delta$ is used to avoid the overcrowding of AFS, the fixed value (constant) $\delta$ will lead to the mutual exclusion of individuals adjacent to the global optimization solution, so AFS cannot get to extreme points accurately and contrast crowding condition after every iteration will also increase the computational cost. The improved method defines the parameter $\delta = 0.75$, when *Try_number* = 180, ignoring the congestion factor, namely, in the initial stages. It needs to limit the size of the artificial fish, but in the latter part fishes have already gathered in optimum, default can reduce the calculation amount and execution time of the algorithm. In this way not only does it improve the operation efficiency of AFS but it also has no effect on convergence.

d) In order to solve the problem of the centers of *K*-medoids by AFS, when swarming and following behavior have failed, preying behavior is carried out, thus increasing the convergence time and its computation. So we renew the behavior as follows: substitute random swim for preying behavior after failing in movement. Moreover, the step is adaptive step-size. The method overcomes the problem that AFS has aggregated at the local solution and missed the global ones and enhances the quality of solutions. The main steps of CAFS algorithm are given below:

---

1. cycle = 1
2. Initialize the food source positions $x_i$, $i = 1, ..., SN$
3. Evaluate the amount (fitness $fit_i$) of food sources and find the best food source which is the initial value of *gbest*
4. repeat
5. For each component $j \in (1, 2, ..., D)$
6. Preying behaviors' Phase

    For each Artificial Fish $i = 1, ..., SN$

        Replace the *j*-component of the *gbest* by using the *j*-component of Artificial Fish *i*

        Calculate the $f[newgbest([g_1, g_2, ..., x_{ij}, ..., g_D])]$

        If $f(newgbest)$ better than $f(gbest)$

        Then *gbest* is replaced by *newgbest*

        For Artificial Fish *i* produce new food source positions $v_i$ by using (2)

        Calculate the value $fit_i$

        Apply greedy selection mechanism

    EndFor

---

7. Calculate the probability values $p_i$ for the solution.
8. Swarming behaviors' Phase
   For each Swarm $i$ = 1, ..., $SN$
      Choose a food source depending on $p_i$
      Replace the $j$-component of the *gbest* by using the $j$-component of fish $i$
      Calculate the $f[newgbest(g_1, g_2, ..., x_{ij}, ..., g_D)]$
      If $f(newgbest)$ better than $f(gbest)$
      Then *gbest* is replaced by *newgbest*
      For Swarm's fish $i$ produce new food source positions $v_i$ by using (1)
         Calculate the value $fit_i$
         Apply greedy selection mechanism
      EndFor
   EndFor
9. Following behaviors' Phase
   If there is a fish becomes follow
   Then replace it with new random source positions
10. Memorize the best solution achieved so far
11. Compare the best solution with *gbest* and memorize the better one.
12. cycle = cycle + 1.
13. until cycle = Maximum Cycle Number

## Benchmark tests

### *Benchmark functions*

In order to compare the performance of the proposed CAFS algorithm with AFS and PSO, we used 6 well-known benchmark functions. One of them is unimodal and the minima [7].

*Sphere function*

$$f_1(x) = \sum_{i=1}^{n} x_i^2 \tag{2}$$

*Rosenbrock function*

$$f_2(x) = \sum_{i=1}^{n} 100 \times (x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \tag{3}$$

*Quadric function*

$$f_3(x) = \sum_{i=1}^{n} \left( \sum_{j=1}^{i} x_j \right)^2 \tag{4}$$

*Sum of different powers*

$$f_4(x) = \sum_{i=1}^{n} |x_i|^{i+1} \tag{5}$$

*Ackley function*

$$f_6(x) = -20\exp(-0.2\sqrt{\tfrac{1}{n}\sum_{i=1}^{n} x_i^2}) - \exp(\tfrac{1}{n}\sum_{i=1}^{n}\cos 2\pi x_i) + 20 + e \tag{6}$$

*Rastrigin function*

$$f_7(x) = \sum_{i=1}^{n}(x_i^2 - 10\cos(2\pi x_i) + 10 \tag{7}$$

## *Parameter settings*

In the experiment, all functions are tested on 30 dimensions and the population size of all algorithms is 100. The PSO algorithm we used is the standard PSO. In PSO algorithm, inertia weight varies from 0.9 to 0.7 linearly, and the iterations and the acceleration factors $c_1$ and $c_2$ are both 2.0. The dimensions, initialization ranges, global optima *, and the corresponding fitness value $f$(*) of each function are listed in Table 1 [3].

Table 1. Parameters of tested functions

| | **Dimensions** | **Initial range** | $x*$ | $f(x*)$ |
|---|---|---|---|---|
| $f_1$ | 30 | [-100, 100]D | [0, 0, …, 0] | 0 |
| $f_2$ | 30 | [-30, 30]D | [1, 1, …, 1] | 0 |
| $f_3$ | 30 | [-65.536, 65.536]D | [0, 0, …, 0] | 0 |
| $f_4$ | 30 | [-1, 1]D | [0, 0, …, 0] | 0 |
| $f_5$ | 30 | [-32.768, 32.768]D | [0, 0, …, 0] | 0 |
| $f_6$ | 30 | [-5.12, 5.12]D | [0, 0, …, 0] | 0 |

## *Simulation results for benchmark functions*

The experimental results, including the best, worst, average, and standard deviation of the function values found in 30 runs, are presented in Table 2 and all algorithms are terminated after 100,000 function evaluations.

As listed in Table 2, the CAFS algorithm performs superior to the others on Sphere, Ackley and Rastrigin benchmark functions, while on Quadric benchmark functions AFS algorithm performs the best. As illustrated in Fig. 1, PSO does worst not only in terms of its convergence speed, but also in terms of its performance on all benchmark functions.

On Sphere function, all algorithms perform very well. Table 2 tells that CAFS is superior to the others, especially on convergence speed, which can be seen in Fig. 1.

From Table 2 and Fig. 1, the convergence speed of CAFS is much higher than the others as to finding good results within relatively few generations, and the AFS is the fastest one. All algorithms are able to consistently find the minimum to functions $f_1$, $f_2$ and $f_3$ within 1000 generations.

After comparing CAFS with PSO algorithms, an obvious result can be seen that the performance of CAFS is significantly superior to the others on continuous unimodal functions

$f_1 \sim f_5$. From the rank values illustrated in Table 2, the order of each performance among the above 3 algorithms is CAFS > AFS > PSO.

Table 2. Results comparison of different optimal algorithms for 30D

| 30D | | AFS | CAFS | PSO |
|---|---|---|---|---|
| **Sphere** | Average | 1.1426e-014 | **1.2346e-018** | 2.2345e-008 |
| | Best | 2.11269e-015 | 5.9142e-019 | 1.7865e-009 |
| | Worst | 3.2378e-014 | **2.7426e-018** | 2.6754e-007 |
| | Std | 8.1226e-015 | **5.3234e-019** | 3.8776e-008 |
| **Rosenbrock** | Average | 3.2313e-001 | **7.3246e+000** | 2.3442e+001 |
| | Best | 1.3680e-002 | 2.8654e-002 | 7.3455e+000 |
| | Worst | 1.3357e+000 | 7.5632e+001 | 9.3535e+001 |
| | Std | 2.9874e-001 | **1.0864e+001** | 1.4356e+001 |
| **Quadric** | Average | **6.2342e-007** | 3.9523e-003 | 4.1956e+002 |
| | Best | 1.3549e-011 | 1.2465e-001 | 3.4355e+002 |
| | Worst | 1.7767e-005 | 1.2665e-001 | 4.3454e+002 |
| | Std | **3.2344e-006** | 2.7866e-002 | 2.9238e+001 |
| **Sum of different powers** | Average | 1.9897e+002 | **3.3453e-004** | 9.5252e+003 |
| | Best | 4.3453e-004 | 3.8183e-004 | 8.3453e+003 |
| | Worst | 4.7389e+002 | 3.3454e-004 | 1.0151e+004 |
| | Std | 1.1697e+002 | **6.3455e-009** | 3.3455e+002 |
| **Ackley** | Average | 6.3454e-006 | **8.3455e-012** | 4.2520e+000 |
| | Best | 1.5905e-006 | 2.5553e-012 | 2.3453e+000 |
| | Worst | 1.34535e-005 | 2.9208e-011 | 5.7625e+000 |
| | Std | 3.5254e-006 | **7.3455e-012** | 8.3370e-001 |
| **Rastrigin** | Average | 1.3455e-001 | **1.3732e-013** | 4.6671e+001 |
| | Best | 3.8257e-009 | 1.3453e-001 | 2.1889e+001 |
| | Worst | 9.34535e-001 | 334535e-013 | 834535e+001 |
| | Std | 3.6710e-001 | **8.3242e-014** | 1.2656e+001 |

## A hybrid clustering algorithm based on CAF clustering model

$X = (x_1, x_2, ..., x_N)$ as the $N$ data samples, $x$ is the data representative point, $C_i$ is an arbitrary cluster, $O_i$ is the center of the cluster $C_i$, ($j$ = 1, 2, …, $k$). The algorithm is presented as follows.

Select $k$ objects in a set $X$ as the initial centers arbitrarily ($O_1$, $O_2$, …, $O_i$, …, $O_k$), then assign the remaining data except for the representative centers by the proximity principle to each cluster. In each cluster ($C_i$), choose a non-central point $O_j$ randomly, calculating the total cost $\Delta E$ after using a non-center instead of the original center point. If $\Delta E < 0$, replace the original $O_i$ with a non-center $O_j$, performing the above steps repeatedly until $k$ centers are fixed. The cost function is also applied to evaluate the clustering quality improval. The function is defined as follows:

$$\Delta E = E_2 - E_1 \tag{8}$$

where $\Delta E$ represents the change of the absolute error standard, $E_2$ refers to the sum of dissimilarity degree between the representative points and the center points in the same cluster after replacing the centers, and $E_1$ represents the dissimilarity degree before replacing.

Calculate $\Delta E$, and if $\Delta E < 0$, the effect of clustering has been improved, then use the new center.

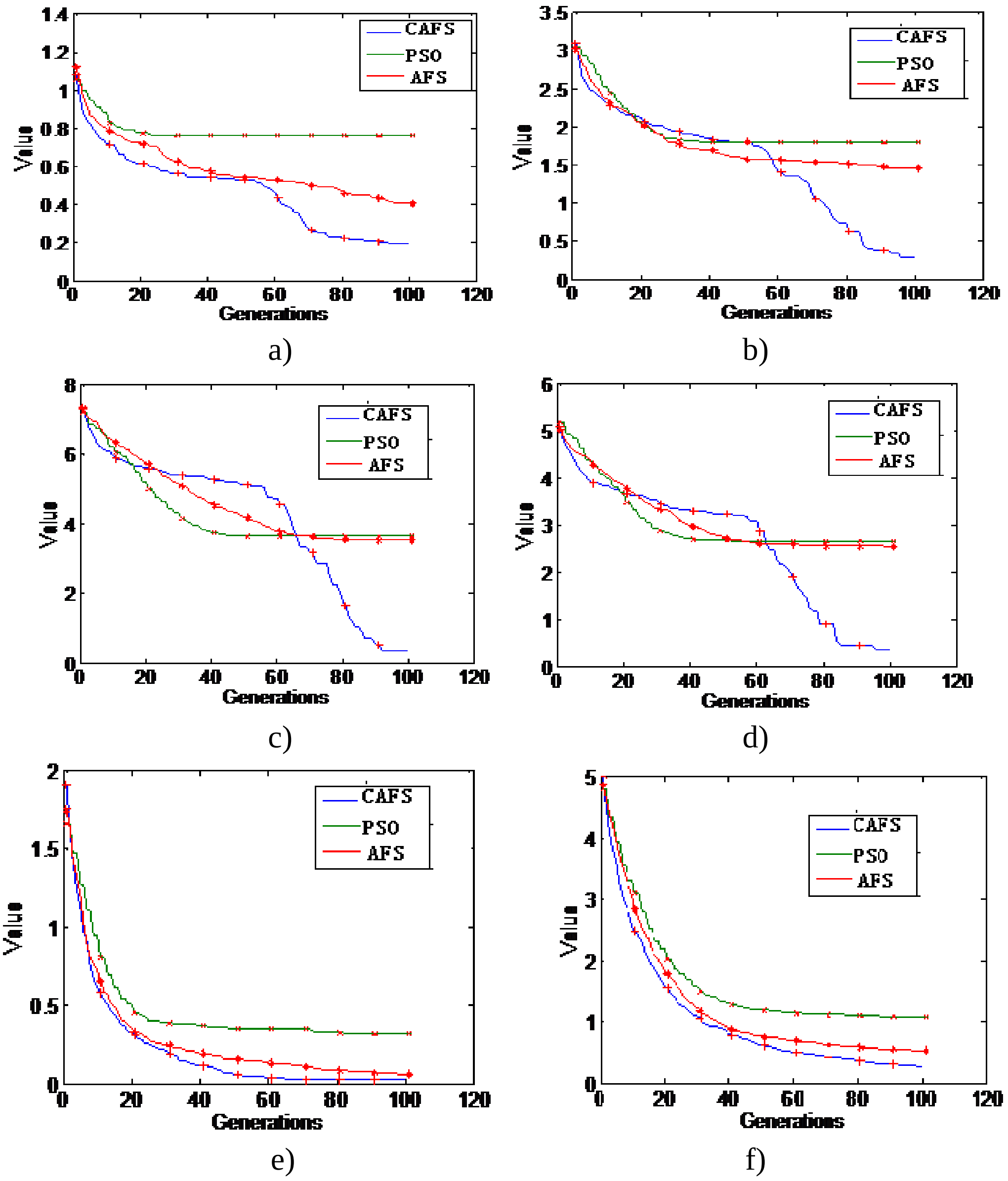



Fig. 1 The median convergence results of 30D unimodal continuous functions:
a) Sphere function; b) Rosenbrock function; c) Quadric function;
d) Sum of different powers; e) Ackley function; f) Rastrigin function.

Definition 1: (adaptive step-size of AFS) Adaptive step-size represents the moving distance of AF changing with iterations, which is defined as:

$$F_{i+1} = F_i + Step \bullet Rand() \quad (9)$$

Definition 2: (clustering evaluation criterion) Objective function measures dissimilarity between the representative points and objects, which means the compact degree of data distribution between classes, so the objective function is defined as:

$$E = \sum_{j=1}^{k} \sum_{X \in C_j} | X - O_j |^2 \tag{10}$$

Step 1: Initialize the value of AF parameters, and then calculate food consistence at a current position by an objective function.

Step 2: Carry out the algorithm through behavior's condition, update the location of AFS by preying, swarming and following behaviors, data density referring to food concentration; then contrast food consistence within vision distance to select solution, with its state recorded in the bulletin board, finally gather the fishes in the areas of high data density.

Step 3: Each state of AFS represents a decision variable; compute the fitness value by an objective function, evaluate optimization degree and record; repeat 2), 3), and update the location information of AFS until the termination condition is met.

Step 4: According to the bulletin board information and the location of fishes, choose input parameters for *K*-medoids, namely the initial center and the number of clusters; using *K*-medoids for a clustering analysis until meeting minimum within-class scatter of data. The minimum within-class *M* is presented as follows:

$$M = \min E \tag{11}$$

The flowchart in Fig. 2 shows the procedure of approach.

## Data clustering experimental results

To analyze the performance of the proposed CAFS approach for the clustering algorithm, the results of PSO and AFS with different data sets have been compared in this paper, which are selected from the UCI machine learning repository.

### *Experiment by simulation data sets*

Simulation data include 300 3D data; running environment for experiment: Pentium (R), 3.00 G; Programming environment: Matlab (2012b); AFSA parameters are set as follows: *Step* is 0.2, *Visual* is 100, $\delta$ is 0.75, *Try_number* (iteration times) is 200, *N* (the total number of AF) is 50.

In the simulation the data are classified by clustering algorithms. A comparison of the results based on the approach this paper proposed and classic hybrid clustering algorithm is presented. The operation result of the classic hybrid method is shown in Fig. 3 and performance of improved approach is shown in Fig. 4.

CAFS finds the centers in 3D data, as shown in Fig. 3, and the aggregation effect is not clear, as a few individuals move to local clusters. The optimization results approximate to global data-intensive areas that can be seen from the iteration route in Fig. 4. The comparison of the performance showing the edge of the clusters is more obvious by the improved method under the same condition. The aggregation effect is clear, so that we can obtain a higher accuracy of the division to verify the advantages of this algorithm.

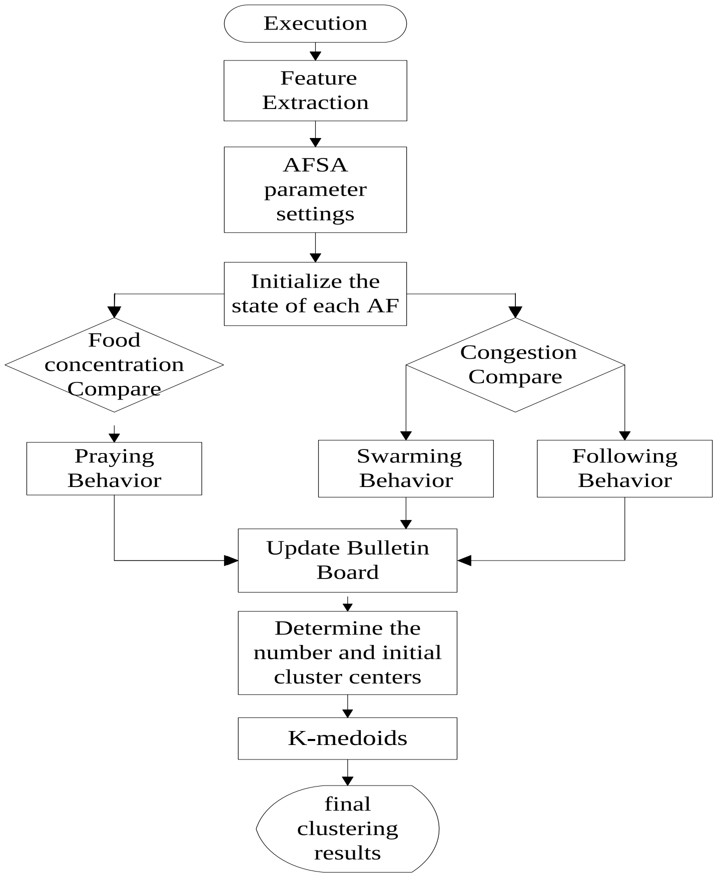



Fig. 2 Flowchart of clustering based on CFSA and *K*-medoids

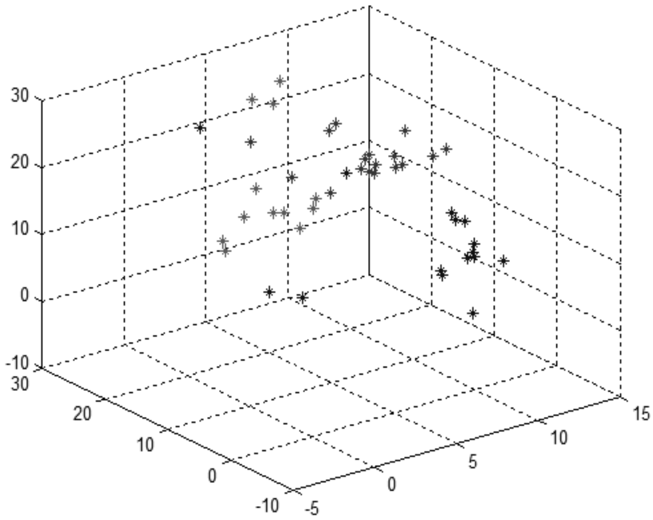


Fig. 3 Optimization based on CAFS

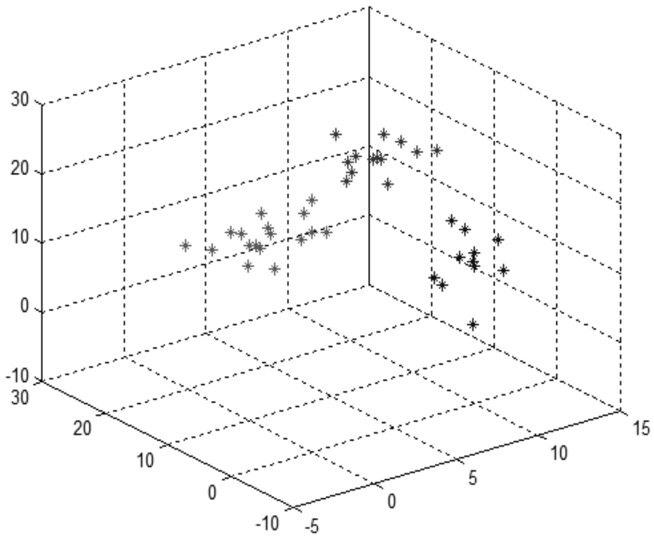



Fig. 4 Optimization based on CAFS and *K*-medoids

Table 3. The results of the two algorithms

| | **Total number of AF** | **Iteration times** | **Iteration time, ms** | **Correctness rate** |
|---|---|---|---|---|
| **General method** | 50 | 200 | 762 | 89 |
| **Proposed method** | 50 | 200 | 685 | 93 |

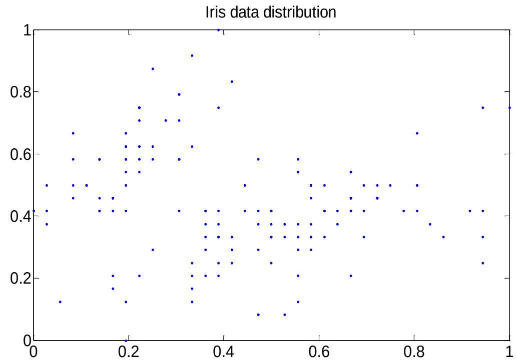
It is shown in Table 3 that the proposed method has improved the accuracy of clustering on the same condition.

### *Experiment by real data sets*

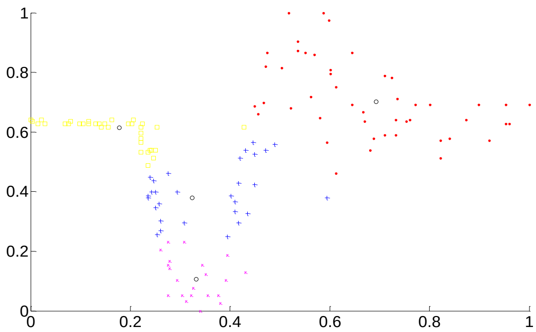
The hybrid algorithm with *K*-medoids and CAFS algorithms is used for data clustering on Iris data sets, which is able to provide the same partition of the data points in all runs. Iris data is thus selected from the UCI machine learning repository. The clustering result of these sets by CAFS and the hybrid clustering algorithm is presented in Fig. 5. From the result in Fig. 5, for all real data sets, the hybrid algorithm with *K*-medoids and CAFS outperforms the other methods.

## Conclusion

In this paper, based on the cooperative approaches, a novel Artificial Fish Swarm algorithm is presented, namely the Cooperative Artificial Fish Swarm. In order to demonstrate the performance of the CAFS algorithm, we compared it with those of AFS, PSO optimization algorithms on several benchmark functions. The comparison of experimental results shows that, firstly, the hybrid clustering algorithm based on CAFS makes similar data gather obviously; secondly, the model is more stable and accurate than the old one; thirdly, it distinguishes samples precisely while also improving the cluster quality and obtaining better centers with a clear division, which means reducing the computation amount.




a) Iris distribution



b) Iris clustering result base on CAFS


c) Iris clustering result base on hybrid algorithm

Fig. 5 The data distribution of Iris data sets and the clustering result by CAFS and hybrid algorithm

The model of a novel and modern intelligence algorithm based on animal autonomous body combines *K*-medoids, which avoids the weakness of dependency on cluster initialization and improves the accuracy of clusterin; its good parallelism can be effectively applied in various fields, which also plays a major role in knowledge discovery, information forecast and decision analysis. However, the convergence speed issue remains to be improved and researched.

## Acknowledgment

*The authors are supported financially by the Natural Science Foundation of Liaoning Province (Project No.2013020011) and this work is supported in part by the International S&T Cooperation Program of China (ISTCP) under Grant 2011DFA91810-5 and Program for New Century Excellent Talents in University of Ministry of Education of China under Grant NCET-12-1012.*

## References

1. Atia D. M., F. H. Fahmy, N. M. Ahmed, H. T. Dorrah (2012). A New Control and Design of PEM Fuel Cell System Powered Diffused Air Aeration System, TELKOMNIKA Indonesian Journal of Electrical Engineering, 10 (2), 291-302.
2. Cheng Y. M., M. Jiang, D. Yuan (2009). Novel Clustering Algorithms Based on Improved Artificial Fish Swarm Algorithm, Sixth International Conference on Fuzzy Systems and Knowledge Discovery, IEEE Press, 3,141-145.

3. Engelbrecht A. P. (2009). Fundamentals of Computational Swarm Intelligence, Wiley Publishing, Inc.
4. Fan Y., D. Wang, M. Sun (2007). Improved Artificial Fish Swarm Algorithm, Journal of Chongqing Normal University (Natural Science Edition), 24(3), 24-26.
5. Han J. W., M. Kamber, J. Pei (2012). Data Mining Concepts and Techniques, Elsevier.
6. He X. D., L. D. Qu (2009). Artificial Fish Swarm Clustering Algorithm, Application Research of Computers, 26(10), 3666-3668.
7. Karaboga D., C. Ozturk (2011). A Novel Clustering Approach: Artificial Bee Colony (ABC) algorithm, Applied Soft Computing, 11(1), 652-657.
8. Li H., Q. Zhang, F. Mult (2009) Objective Optimization Problems with Complicated Pareto Sets, MOEA/D and NSGA-II, IEEE Transactions on Evolutionary Computation, 13(2), 284-302.
9. Liu B., Y. Q. A. Zhou (2008). Hybrid Clustering Algorithm Based on Artificial Fish Swarm Algorithm, Journal of Computer Engineering and Applications, 44(18), 136-138.
10. Tao X. M., J. Xu, L. B. Yang (2010). An Improved Hybrid Algorithm Based on Particle Swarm Optimization and K-means Algorithm, Journal of Electronics & Information Technology, 32(1), 92-94.
11. Wang L., X. Fu (2005). Data Mining with Computational Intelligence (Advanced Information and Knowledge Processing), Springer-Verlag, New York.
12. Wang X. H., J. Shen, R. B. Wang (2010). A New Hybrid Algorithm Based on Ant Colony and Clustering. Journal of Hangzhou Dianzi University, 30(1), 26-27.
13. Xia N. X., Y. D. Su (2010). An Efficient K-medoids Clustering Algorithm, Journal of Application Research of Computers, 27(12), 4517-4519.
14. Xie J. Y. (2010). A Simple and Fast Algorithm for Global K-means Clustering, 2nd International Workshop on Education Technology and Computer Science (ETCS), Wuhan, China, 2, 36-40.
15. Yang B. R., D. Y. Cao (2010). An Improved K-medoids Clustering Algorithm, 2nd International Conference on Computer and Automation Engineering (ICCAE), Singapore, 132-135.
16. Zhao Y. Z., X. Liu, H. Zhang (2013). The K-medoids Clustering Algorithm with Membrane Computing, TELKOMNIKA Indonesian Journal of Electrical Engineering, 11(4), 2050-2057.
17. Zhu C. X., Z. Ying, J. X. Song (2009). Method of Image Segmentation Based on Fuzzy C-мeans Clustering Algorithm and Artificial Fish Swarm Algorithm, International Conference on Intelligent Computing and Integrated Systems (ICISS), IEEE Press, 254-257.

**Assoc. Prof. Hongwei Zhao**
E-mail: zhw30@163.com

Hongwei Zhao is an Associate Professor at Shenyang University. He is a supervisor for postgraduates at School of Information Engineering. His main research areas are swarm intelligence theory and optimization decision method.

**Prof. Liwei Tian**
E-mail: tianliwei@163.com

Liwei Tian is a Professor at Shenyang University. He is a supervisor for postgraduates at School of Information Engineering. His main research areas are computational biology and swarm intelligence theory.