Fuzzy Adaptive Artificial Fish Swarm Algorithm

Danial Yazdani[1], Adel Nadjaran Toosi[2], Mohammad Reza Meybodi[3],

[1] IAU of Qazvin, Department of Electerical, Computer and Information Technology, Qazvin, Iran.
[2] IAU of Mashhad, Department of Computer Software Engineering, Mashhad, Iran.
[3] Amirkabir University of Technology, Department of Computer Engineering and Information Technology, Tehran, Iran.

{ D_yazdani@qiau.ac.ir , NadjaranToosi@mshdiau.ac.ir , mmeybodi@aut.ac.ir }

Abstract. Artificial Fish Swarm Algorithm (AFSA) is a kind of swarm intelligence algorithms which usually employs in optimization problems. There are many parameters to adjust in AFSA like *visual* and *step*. Through constant initializing of visual and step parameters, algorithm is only able to do local searching or global searching. In this paper, two new adaptive methods based on fuzzy systems are proposed to control the visual and step parameters during the AFSA execution in order to control the capability of global and local searching adaptively. First method uniformly adjusts the visual and step of all fish while in the second method, each artificial fish has its own fuzzy controller for adjusting its visual and step parameters. The experiments and evaluations of the proposed methods were performed on eight well known benchmark functions in comparison with standard AFSA and Particle Swarm Optimization (PSO). The overall results show that proposed algorithm can be effective surprisingly.

Keywords: Artificial fish swarm algorithm (AFSA), particle Swarm Optimization (PSO), fuzzy system, global search, local search.

1 Introduction

Solving the NP-complete problems is one of the toughest issues that computer scientists always were faced with. Among the algorithms, swarm intelligence algorithms have been significantly proved their capabilities in this area. Among them Particle Swarm Optimization (PSO) [1] and Ant Colony Optimization (ACO) [2] are the most well-known algorithms that have been ever proposed. These algorithms have some characteristics that make them suitable for solving NP-complete problems, like scalability, fault tolerance, consistency, higher speed, flexibility, parallelism and etc.
Artificial fish swarm algorithm (AFSA) [3], proposed by Li Xiao Lei in 2002, is a stochastic population-based algorithm motivated by intelligent collective behavior of fish groups in nature. AFSA has characteristics of non-sensitive initial artificial fish location, flexibility and fault tolerant. It has been applied on different problems

including fuzzy clustering [4], Resource Leveling [5], optimization of PID controller parameters [6], spread spectrum code estimation [7], data mining [8], optimization of DNA encoding sequence [9] and etc.

At its time, AFSA has not been comprehensively accepted by scientists for some reasons. High computational complexity, difficult implementation of the algorithm and the results not significantly better than similar algorithms can be noted here. In fact, algorithms such as PSO with less computational complexity are easier to implement and the results obtained from different versions of PSO shows better performances than AFSA. Notice that AFSA is not a version of PSO and differs significantly from PSO. One of the main differences between these two algorithms is that particles in PSO move completely based on the past movements and their previous experiences in the problem environment, although, artificial fish movements depend on their current positions and other members of the group situations. Accordingly, movements of the fish differ from particles.

There are two important parameters in AFSA: Visual and Step. Artificial Fish search the problem environment as broad as their Visual, and then they move toward the target based on the random value of the step in the each iteration. In standard AFSA initial values for these parameters have a great affect on the final result because of the fact that these parameters remain constant and equal to the initial values until the algorithm termination. If larger initial values for Visual and Step have been selected, artificial fish swarm move faster toward the global optimum and will be more capable of passing the local optimums. Selecting lower values for these parameters causes better results in local searching.

In this paper, two fuzzy adaptive methods have been proposed that regulates the visual and step of artificial fish. First method based on two inputs, ratio of the improved fish and iteration number, generate a weight. Visual and step are multiplied by the weight to attain the next iteration values for these parameter. It is important to mention that in the first method, which is called Fuzzy Uniform Fish (FUF) here, parameters of visual and step of all artificial fish are adjusted by a global weight attained of fuzzy controller output. Fuzzy Autonomous Fish (FAF), the second method, combines inputs like distance from the best artificial fish, fitness ranking of the current fish and iteration number for each fish to create a weight value in order to adjust the visual and step in the next iteration.

Experiment results show the proposed method enhanced better outcomes in comparison to standard AFSA. To achieve better balance between local and global searching, the proposed algorithm has employed the fuzzy rules. As a consequence, the algorithm searching efficiency and convergence speed to the global optimum has been improved considerably.

Before this, parameter which is called inertia weight has been applied on particle swarm optimization (PSO) algorithm by Shi and Eberhart [10]. Output of the proposed fuzzy engines here plays a similar role to inertia weight in PSO.

This paper is organized as follows. In Section 2, standard AFSA is briefly introduced. Afterward, section 3 presents the fuzzy approaches in detail through the introduction of input parameters, output weight and fuzzy decision strategy. Section 4 experimentally compares the proposed methods with PSO algorithm and Standard AFSA using a set of benchmark functions. Finally, conclusions are drawn in Section 6.

2 Artificial Fish Swarm Algorithm

In natural world fish look for foods with local search of fish individual and global search of the group to reach the areas with higher amount of foods. The basic idea of AFSA, from long observation of fish swarm intelligence in nature, is to imitate the fish behaviors in solving the optimization problem. In fact AFSA is a random parallel search algorithm based on simulating fish swarm behaviors like preying, swarming and following.

Artificial fish, fabricated entity of true fish, is used to explore the problem environment. The environment in which the artificial fish lives is mainly the solution space. Artificial fish moves depend on its current state and its environmental state in its next behavior, and it influences the environment via its own activities and other companions' activities.

As it has been shown in the figure 1, the artificial fish realizes external perception by its vision.

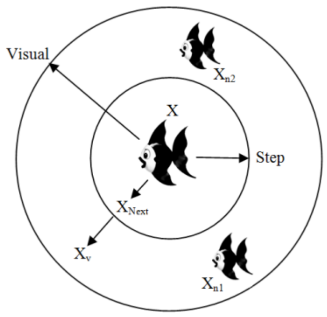


Fig. 1. Artificial Fish and the environment around it.

All The current state of an artificial fish is showed by X=$(x_1,x_2,...x_n)$ in N-dimensional space. Visual is the visual distance, and X_v is somewhere in the visual distance that the artificial fish desires to move there. If the state $X_v=(x_1^V, x_2^V, \dots, x_n^V)$ at the visual position is better (aspect of food consistence) than the current state it goes forward a step in this direction, and arrives in the X_{Next}; otherwise, if the X_v is not better than current state, it continues at inspecting tour in its vision. Step is the length of the longest possible movement. The greater number of inspecting tour the fish does, the more knowledge about overall states of the vision it obtains.

The artificial fish model includes variables and functions. The variables comprise the current position of the artificial fish (*X*), the moving step length (*Step*), visual distance (*Visual*), the maximum try number in searching behavior (*try_number*) and the crowd factor (δ) that ensures the artificial fish move to optimum in a wide field. The functions include behaviors of the artificial fish, like searching food (prey), swarm and following. Implementation of these behavior functions in each iteration of the algorithm execution includes the following behavior and after that swarm behavior. If each of them does not contain the favorable result, searching food behavior will be employed.

2.1 Searching Behavior

Intention of finding places with larger amount of food is a biological behavior among fish. Generally, fish based on their senses like sight (vision) look for places with higher density of food in water, and then they decide to move in a direction. Accordingly, we simulate the behaviors of fish based on this characteristic to find the global optimum, which is the basic idea of the AFSA.

Let X_i be the current state of artificial fish. A random state X_j in the visual distance is selected. X_j is computed by the formula 1:

$$X_j = X_i + Visual.Rand() \tag{1}$$

Where *Visual* represents the visual distance and *Rand()* is a random numbers between 0 and 1. The objective function value, $Y=f(X)$, shows the amount of food. If $Y_i<Y_j$ (in maximizing problems) then the artificial fish goes forward a step in this direction, based on the formula 2:

$$X_i^{(t+1)} = X_i^{(t)} + \frac{X_j - X_i^{(t)}}{\left\| X_j - X_i^{(t)} \right\|}.Step.Rand(\ ) \tag{2}$$

Where $d_{ji}=\|X_j-X_i\|$ is the distance between state X_i and X_j. if $Y_i \geq Y_j$, another X_j is selected randomly by artificial fish and then it is examined whether it satisfies the forward condition($Y_i<Y_j$) or not. If it cannot satisfy the condition after *try-number* times, it moves a step freely according to the formula number 3:

$$X_i^{(t+1)} = X_i^{(t)} + Step.Rand(\ ) \tag{3}$$

2.2 Swarm Behavior

The fish will assemble in groups naturally in the moving process, which is a kind of living habits in order to guarantee the existence of the colony and avoid dangers. Let X_i be the current state of artificial fish. X_c is the center position of swarm. There are n_f neighbors within the visual area of X_c and N is the total number of artificial fishes. If $n_f/n < \delta$ and $Y_c>Y_i$,which means center is not very crowded and has better fitness function value, artificial fish will swim a random distance towards X_c based on the formula number 4. Otherwise, if $n_f=0$ or $Y_c \leq Y_i$ then artificial fish will resume the behavior of searching.

$$X_i^{(t+1)} = X_i^{(t)} + \frac{X_C - X_i^{(t)}}{\left\| X_C - X_i^{(t)} \right\|}.Step.Rand(\ ) \tag{4}$$

2.3 Follow Behavior

In the process of fish swarm moving when one or some of fishes find food, the other neighbors and close fishes go after them and reach food quickly. Let X_i be the current state of artificial fish and X_j be an artificial fish in neighborhood of X_i ($d_{ij} \leq Visual$) and n_f be the number of fishes in the neighborhood of Xj. If $Y_j > Y_i$ and $n_f/n < \delta$ then X_j will have more amount of food (better fitness function value) and will not be very crowded. So artificial fish will swim a random distance towards X_j based on the formula number 5. Otherwise, artificial fish will resume the behavior of searching.

$$X_i^{(t+1)} = X_i^{(t)} + \frac{X_j - X_i^{(t)}}{\left\| X_j - X_i^{(t)} \right\|} .Step.Rand(\ ) \tag{5}$$

3 Proposed Algorithm

In nature, fish swarms members have a certain visual that directly depends on the fish type, environment conditions (e.g. water fog) and around obstacles (e.g. water plants and other fish). When the swarm moves towards a target (e.g. food) as much as it converges on, visibility is reduced due to density. Here, the main motivation of our work is to implement this natural reality for the artificial fish.

In AFSA, artificial fish search the problem environment based on their visual and then they move towards the target by a random value of their step. In standard AFSA, determination of the initial values of the step and visual influence on the final result essentially. Values of these parameters remain constant and equal to the initial values during the algorithm execution. If greater initial values have been considered for these parameters, artificial fish swarm will move faster towards the global optimum because artificial fish are able to search bigger environment around them and move with bigger step in the each iteration. Under such circumstances, artificial fish are more powerful in escaping from the local optimums. Incidentally, there are some deficiencies in larger values of step and visual. Accuracy and consistency of the algorithm will decrease in such situation.

In fact, the algorithm acts better in global searching, but after approaching the global optimum, it is incapable of an appropriate local search because of the fact that the visual is larger than it should be. Therefore, owing to the large value of the visual, positions with better fitness are unlikely to find and fish will pass the global optimum, even they may go far from it. Considering smaller values for these parameters make algorithm more consistent and accurate but it causes the algorithm to move towards the target more slowly and incapable of escaping local optimums.

Based on the above facts, in order to attain better results, larger initial value for visual and step is selected at first. Afterward, it is reduced during the algorithm execution adaptively. As a result, fish move towards the target quickly and are more capable of escaping local optimums. Then by approaching the target, artificial fish can accurately investigate the environment by smaller Visual and Step.

In order to control values of step and visual and balancing between global search and local search, a novel parameter, called *Constriction Weight* here, has been proposed. Weight must be greater than 0 and smaller than 1. Current iteration visual and step values calculate according to the following formulas in presence of weight parameter:

$$Visual_{itr} = CW \times Visual_{itr-1} \tag{6}$$

$$Step_{itr} = CW \times Step_{itr-1} \tag{7}$$

Where *CW* is *Constriction Weight* which is generated as a output of the proposed fuzzy engines described in the following parts of this paper. $Visual_{itr}$ and $Step_{itr}$ stand for the current iteration visual and step and $Visual_{itr-1}$ and $Step_{itr-1}$ is the pervious iteration visual and step respectively.
With the purpose of attaining better values for the visual and step two different fuzzy methods for calculating the weight have been proposed here:

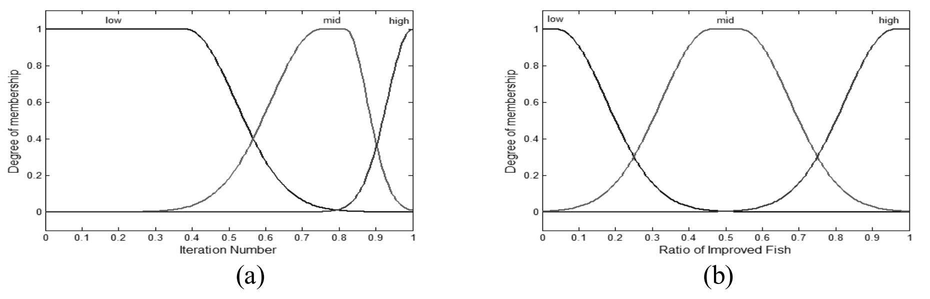
3.1 Fuzzy Uniform Fish (FUF)

In this method, weight is a value between 0 and 1 that is calculated as an output of the fuzzy engine. All of the fish in the swarm then adjust their visual and step based on the output weight. The proposed fuzzy engine has two inputs and one output: Iteration number and ratio of improved fish as inputs and constriction weight as an output. *Iteration Number*, normalized between 0 and 1, is the proportion of current iteration number to the final iteration number. In fact, visual and step parameters must be larger in initial iterations to achieve better global searching. Therefore, visual and step values decreases very smooth in initial iterations of the algorithm execution. Progress of the algorithm causes the artificial fish come close to the global optimum of the problem. So, in order to increasing the local search capability of the algorithm, visual and step parameters must be reduced by larger amounts. As a result artificial fish are able to search the global optimum more keenly. Considering the above facts, by approaching to the final iterations, the proposed fuzzy engine increases the constriction weight to reduce the visual and step more sharply.

Ratio of Improved Fish is the proportion of the number of fish that find better positions in problem space (points with higher fitness) to the total number of fish in comparison with previous iteration.

When most of the artificial fish find better positions compared with previous iteration, visual and step parameter are suitable. Therefore, there is no need to reduce them. In this situation *Ratio of Improved Fish* value is a number close to 1. Conversely, when most of the artificial fish do not experience any improvement rather than previous iteration the constriction weight must be increased to reduce the visual and step. Act of reducing visual and step raising the probability of the finding positions with better fitness values due to the fact that it increase of the local capability of the fish. Considering the above facts, the proposed fuzzy engine increases the constriction weight by reduction of the ratio of the improved fish value and vice versa. Figures 2(a) and 2(b) show the membership functions for Inputs: *Iteration Number* and *Ratio of Improved Fish.*

Constriction Weight is the output of the fuzzy engine which has the membership functions of figure 2(c).The proposed fuzzy engine, which is a Mamdani fuzzy inference system with centroid of area defuzzification strategy, uses the rules shown in the fuzzy associative memory in Table 1.



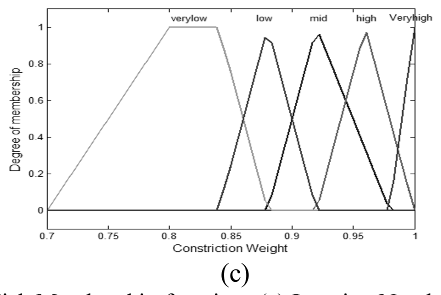


Fig. 2. Fuzzy Uniform Fish Membership functions.(a) Iteration Number. (b) Ratio of Improved Fish. (c) Constriction Weight.

Table 1. Fuzzy associative memory for the proposed FUF engine. VL: very low, L: low, M: mid, H: high and VH: very high.

Iteration Number	Ratio of Improved Fish	Constriction Weight
L	H	VH
L	M	H
L	L	M
M	H	H
M	M	M
M	L	L
H	H	M
H	M	L
H	L	VL

3.2 Fuzzy Autonomous Fish (FAF)

The main difference between the Fuzzy Autonomous Fish (FAF) method and Fuzzy Uniform Autonomous (FUF) is that each artificial fish adjust its visual and step parameter individually and independent of the rest fish in FAF.

In this method, initial values of the visual and step parameters for all the fish is identical. But through the algorithm execution, for each artificial fish, these parameters decrease based on the output of the fuzzy engine which is obtained according to the position of the fish in the group. Consequently, each fish has its own visual and step parameter independent and different from the others. The fuzzy inference system based on three inputs calculates the Constriction Weight. These inputs are *Distance from Best*, *Fitness Ranking* and *Iteration Number*.

Distance from Best is a normalized rank-based value between 0 and 1 for each fish. It is calculated based on Euclidian distance of the artificial fish from the best artificial fish (fish with best fitness value). Afterward, all fish are sorted and ranked based on the Euclidian Distance from the best fish. Proportion of the ranking number to a total number of artificial fish is considered as the *Distance from Best* input parameter of the fuzzy engine.

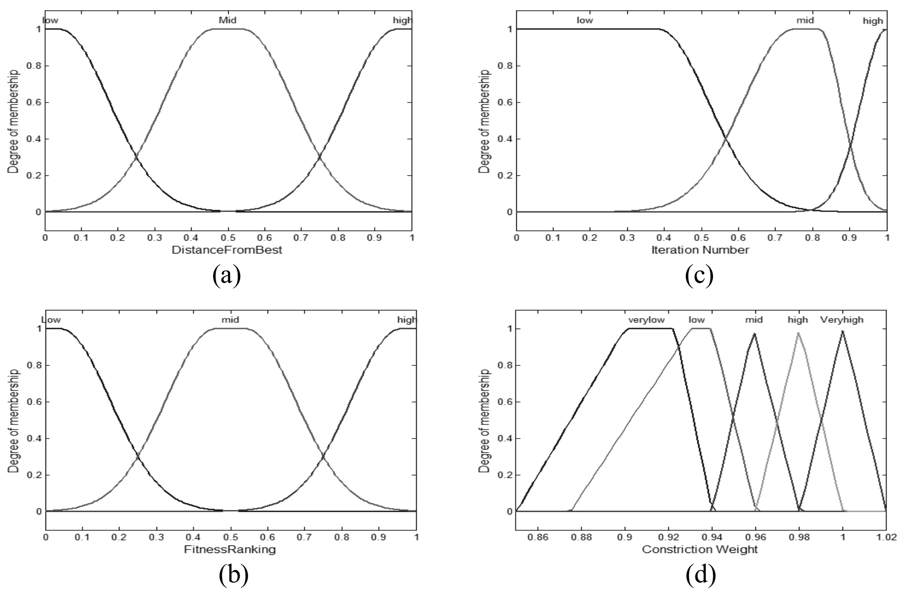


Fig. 3. Inputs and Outputs of the Fuzzy Autonomous Fish inference system membership functions. (a)Distance from Best. (b)Fitness Ranking. (c)Iteration Number. (d)Constriction Weight

In this situation, as long as the distance of the fish from the best fish is large, parameter of visual and step must be reduced lesser. If a fish is far from the best fish then the amount of visual and step parameters should be reduced more sharply in order to perform the search more accurately. This policy leads to fish cannot step out from the area with high level of appropriateness with long step.

Fitness Ranking is equal to the proportion of the ranking number, calculated based on the fitness value for the artificial fish, to the total number of artificial fish. In order to calculate this parameter, all artificial fish are sorted based on their fitness value and then ranked based on the index of their position on the list. Like *Distance from Best,* the worse ranking that artificial fish have, lesser amount visual and step parameters should be reduced and vice versa. And Iteration number has the same definition of the FUF engine. Figure 3 depicts the membership functions of FAF engine. The FAF engine is a Mamdani fuzzy inference system with centroid of area defuzzification strategy. FAF uses 27 rules have been shown in the fuzzy associative memory in Table 2.

Table 2. Fuzzy associative memory for the proposed FAF engine. VL: very low, L: low, M: mid, H: high and VH: very high.

Distance from Best	Fitness Ranking	Iteration Number	Weight	Distance from Best	Fitness Ranking	Iteration Number	Weight
L	L	H	VL	M	M	L	H
L	L	M	VL	M	H	H	M
L	L	L	L	M	H	M	H
L	M	H	L	M	H	L	VH
L	M	M	M	H	L	H	L
L	M	L	H	H	L	M	M
L	H	H	L	H	L	L	H
L	H	M	M	H	M	H	M
L	H	L	VL	H	M	M	H
M	L	H	VL	H	M	L	VH
M	L	M	L	H	H	H	M
M	L	L	M	H	H	M	H
M	M	H	L	H	H	L	VL
M	M	M	M				

4 Experimental Results

In this section, eight benchmark functions are tested for comparison. All of them are standard test functions. Table 3 shows name of the functions, functions' equation and search space for each function [11].

Table 3. Eight test functions used in this paper.

Name of Function	Function	Search Space
Ackly	$f_1(x) = 20 + e - 20e^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2}} - e^{\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)}$	$-32 < x_i \leq 32$
Sphere	$f_2(x) = \sum_{i=1}^{D} x_i^2$	$-100 \leq x_i \leq 100$
Griewank	$f_3(x) = \sum_{i=1}^{n}\left(\frac{x_i^2}{4000}\right) - \prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-600 < x_i \leq 600$
Schwefel's2.22	$f_4(x) = \sum_{i=1}^{D}\lvert x_i\rvert + \prod_{i=1}^{D}\lvert x_i\rvert$	$-10 < x_i \leq 10$
Step	$f_5(x) = \sum_{i=1}^{D}\left(\lfloor x_i + 0.5\rfloor\right)^2$	$-100 < x_i \leq 100$
Rastrigin	$f_6(x) = \sum_{i=1}^{D}\lvert x_i\rvert + \prod_{i=1}^{D}\lvert x_i\rvert$	$-5.12 < x_i \leq 5.12$
Non continuous Rastrigin	$f_7(x) = \sum_{i=1}^{D}\left(y_i^2 - 10\cos(2\pi y_i) + 10\right)$ $Where \begin{cases} x_i & \lvert x_i\rvert < 0.5 \\ \frac{round(2x_i)}{2} & \lvert x_i\rvert \geq 0.5 \end{cases}$	$-5.12 < x_i \leq 5.12$
Generalized Penalized	$f_8(x) = \frac{\pi}{D}\left\{10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2\left[1 + 10\sin^2(\pi y_{i+1})\right] + (y_D - 1)^2\right\}$ $+ \sum_{i=1}^{D} u(x_i, 10, 100, 4)\ ,\quad u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m & x_i > a \\ 0 & -a \leq x_i \leq a \\ k(-x_i - a)^m & x_i < -a \end{cases}$	$-50 < x_i \leq 50$

In experiments, Initial values for Visual and Step has been considered 40% and 25% of the range length of the fitness function variables respectively (For example range length of the fitness function variables in Ackly function is 64). Crowd factor is 0.5, maximum try-number is 10 and population number in standard AFSA, FAF and FUF is 30. In global version of PSO (GPSO) [10], inertia weight value linearly decreases from 0.9 to 0.4 during the algorithm execution. The population size is equal to 5 * D where D is the problem dimensions. At last, parameters c_1 and c_2 have been set in form of $c_1 = c_2 = 2$. Experiments repeated 100 times; best, mean and standard deviation obtained from running of standard AFSA, FUF, FAF, and GPSO in 30-dimensional spaces on eight benchmark functions for 1000 Iterations have been shown on Tables 4.

As it is shown in table 4 and figure 4, in standard-AFSA, since the visual and step parameter are constant during the algorithm execution, algorithm is not able to reach the acceptable results in none of the benchmark functions. The main reason for this deficiency is that the algorithm is not adequate flexible in different situations with which group are faced in problem space. In other words, algorithm shows a uniform behavior in every possible situation and is not able to keep the balance between global and local searching.

Table 4. Comparison of Standard AFSA, GPSO, FAF and FUF on Ackly function.

Function	Algorithm	Min	Mean	Std.Dev
Ackly	Std.AFSA	5.7284	6.8145	0.5602
	GPSO	1.8082e-04	7.4578e-04	3.6421e-04
	FUF	1.3322e-14	1.6875e-14	3.2601e-15
	FAF	1.3322e-14	1.7941e-14	3.3704e-15
Sphere	Std.AFSA	4.1020+02	7.2819+02	1.8326e+02
	GPSO	3.3124e-07	5.1620e-06	9.0477e-06
	FUF	4.4040e-49	7.0516e-49	1.5620e-49
	FAF	1.2361e-68	8.8204e-63	2.7849e-62
Griewank	Std.AFSA	4.0010	6.9563	1.5048
	GPSO	1.2028e-05	0.0390	0.1008
	FUF	2.2204e-16	3.6637e-16	1.0865e-16
	FAF	4.4408e-16	7.54951e-16	1.8724e-16
Schwefel's2.22	Std.AFSA	7.2255	9.7531	1.0017
	GPSO	1.1192e-04	3.0191	4.1696
	FUF	7.1387e-21	9.5280e-16	3.9264e-15
	FAF	1.8153e-08	3.9340e-04	0.0010
Step	Std.AFSA	383	667.6500	184.6787
	GPSO	0	0	0
	FUF	0	0	0
	FAF	0	0	0
Rastrigin	Std.AFSA	77.5419	162.0464	34.2185
	GPSO	21.9819	41.6566	9.5776
	FUF	0	0.5472	1.0447
	FAF	0	0.5969	1.3846
Noncontinuous Rastrigin	Std.AFSA	76.3466	139.1457	18.8128
	GPSO	28.3333	74.3343	32.0512
	FUF	0	1.0500	1.1909
	FAF	0	2.3773	2.6676
Generalized Penalized	Std.AFSA	2.6573	5.1480	1.2510
	GPSO	1.9245e-06	0.0618	0.1120
	FUF	3.6357e-32	1.8239e-30	4.7910e-30
	FAF	3.7648e-32	1.3889e-17	4.7272e-17

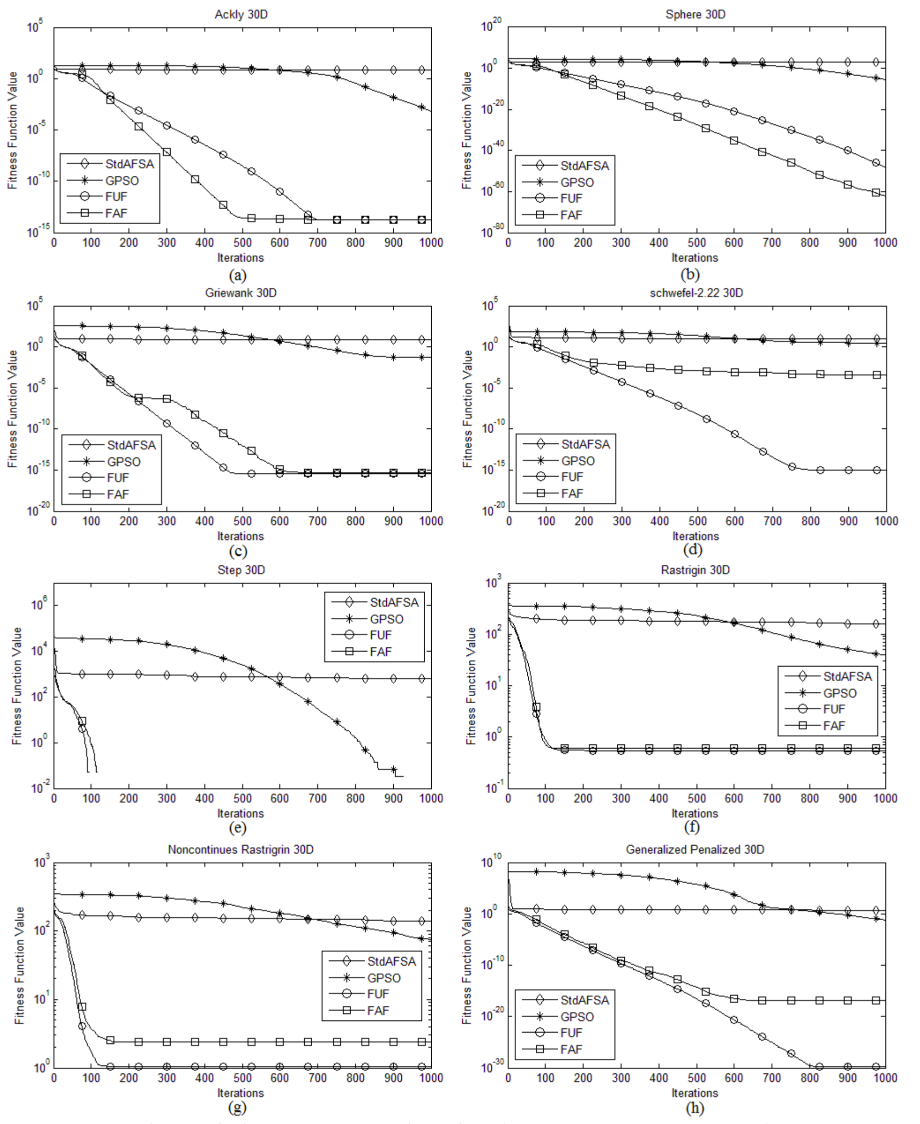


Fig. 4. Comparison of the average results of Std-AFSA, GPSO, FAF and FUF on 30-Dimensional (a) Ackly, (b) Sphere, (c) Griewank, (d) Schwefel 2.22, (e) Step, (f) Rastrigin, (g) Noncontinues Rastrigin and (g) Generalized Penalized functions in 1000 iterations.

Results show that proposed algorithms, FAF and FUF, considerably improved the effectiveness. The main reason is that the proposed algorithms dynamically decrease the visual and step parameter based on the output of the fuzzy systems which act according the group position. Therefore, during the algorithm execution, capability of the algorithm in global searching gradually decreases and capability of it in local searching increases. Artificial fish pass the local optimum more rapidly then they search around global optimum more accurately after converging to the global optimum.

Both of the proposed algorithms attain similar results on Ackly, Griewank, Step and Rastrigin. But in Sphere which a function without a local optimum, FAF produces better results than FUF. Actually, because of the diversity in visual and step values among the artificial fish, FAF is more capable in local searching. This can be adverse in some functions like Generalized Penalized and Schwefel's2.22. In FAF, when a artificial fish continually experiences higher ranks, it reduce its visual and step faster, subsequently, it become frozen finally.
Totally speaking, proposed algorithms are more efficient than Standard-AFSA and Global version of PSO.

4 Conclusion

In this paper, two modified AFSA algorithms have been proposed. Proposed algorithms take the members group position into account in adjusting the visual and step parameter. It can be easily seen that AFSA with an appropriate adaptive visual and step can acts better than standard AFSA and other similar optimization algorithms like PSO. Two introduced fuzzy engine bring a significant improvement on the AFSA performance. Various simulations have been done to support it.

1. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: IEEE International Conference on Neural Network, pp. 1942--1948, Perth (1995)
2. Dorigo, M., Birattari, M., Stutzle, T.: Ant Colony Optimization. In: IEEE Computational Intelligent Magazine, Vol. 1, pp. 28--39 (2006)
3. Li, L.X., Shao, Z. J., Qian, J.X.: An Optimizing Method Based on Autonomous Animate: Fish Swarm Algorithm. In: Proceeding of System Engineering Theory and Practice, Vol. 11, pp. 32--38 (2002)
4. Hi, S., Belacel, N., Hamam, H., Bouslimani, Y.: Fuzzy Clustering with Improved Artificial Fish Swarm Algorithm. In: International Joint Conference on Computational Sciences and Optimization 09, Vol. 2, pp. 317--321, Hainan (2009)
5. Tian, W., Tian, Y.: An Improved Artificial Fish Swarm Algorithm for Resource Leveling. In: International Conference on Management and Service Science, pp. 1--4, Wuhan (2009)
6. Luo, Y., Zhang, J., Li, X.: The Optimization of PID Controller Parameters Based on Artificial Fish Swarm Algorithm. In: IEEE International Conference on Automation and Logistics, pp. 1058--1062, Jinan (2007)
7. Jiang, M., Wang, Y., Rubio, F., Yuan, D.: Spread Spectrum Code Estimation by Artificial Fish Swarm Algorithm. In: IEEE International Symposium on Intelligent Signal Processing, pp. 1--6, Alcala de Henares (2007)
8. Zhang, M., Shao, C., Li, M., Sun, J.: Mining Classification Rule with Artificial Fish Swarm. In: 6th World Congress on Intelligent Control and Automation, Vol. 2, pp. 5877--5881, Dalian (2006)
9. Cui, G., Cao, X., Zhou, J. , Wang, Y.: The Optimization of DNA Encoding Sequences Based on Improved Artificial Fish Swarm Algorithm. In: IEEE International Conference on Automation and Logistics, pp. 1141--1144, Jinan (2007)
10. Shi, Y., Eberhart, R.: A Modified Particle Swarm Optimization. In: IEEE International Conference on Evolutionary Computation Proceedings, pp. 69--73, Anchorage (1998)
11. Zhan, Z.H., Zhang, J., Li, Y., Chung, H.S.H.: Adaptive Particle Swarm Optimization. In: IEEE Transaction on System, Man and Cybernetics, Part B: Cybernetics, Vol. 39, No. 6, pp. 1362--1381 (2009)