



Normative fish swarm algorithm (NFSA) for optimization

Weng-Hooi Tan¹ · Junita Mohamad-Saleh¹

Published online: 9 May 2019

© Springer-Verlag GmbH Germany, part of Springer Nature 2019

Abstract

In this paper, a swarm-based optimization algorithm, normative fish swarm algorithm (NFSA) is proposed as an effective global and local search technique to obtain effective global optima at superior convergence speed. Artificial fish swarm algorithm is a recent swarm-based algorithm that imitates the behavior of fish swarm in the real environment. Many improvements and modifications have been proposed regularly on fish swarm algorithm to improve the performance of optimization, but to date, existing fish swarm algorithms have not yet obtained a global optimum at extremely superior convergence rates. Hence, there still remains a huge potential for the development of fish swarm algorithm. NFSA hybridizes the characteristics of PSOEM-FSA with the normative knowledge as the complementary guidelines for more accurate and precise global optimum approaching. NFSA further improves the adaptive parameters in term of *visual* and *step* to balance the contradiction between the exploration and exploitation processes. Random initialization of the initial population is introduced to spread out the solution candidates of artificial fishes over the solution space. For the purpose of experiments, ten benchmark functions have been used in the evaluation process. The proposed algorithm is then compared with other related algorithms published in the literature. The results proved that the proposed NFSA achieved superior results in terms of convergence rate and best optimal solution on a majority of the tested benchmark functions in comparison with other comparative algorithms.

Keywords Artificial fish swarm algorithm (AFSA) · Adaptive visual and step · Particle swarm optimization (PSO) · PSO with extended memory (PSOEM) · Normative knowledge · Cultural algorithm (CA) · *t* test

1 Introduction

An optimization problem is a problem of finding the best solution from all feasible solutions. Every optimization problem has a unique global optimal solution, where the output is optimized. Global search and optimization algorithm (GSOA) has been designed to solve the optimization problems as it plays the role to find the global optimal solution for a given optimization problem. Swarm-based algorithms, categorized under the group of global search and optimization algorithm, are commonly developed by

imitating the behavior of creatures in nature. They have been widely involved in different fields of application, such as knapsack problem solving, text document clustering analysis (Abualigah et al. 2018a, b; Karol and Mangat 2013) and maximum power point tracking (MPPT). There exists another option of mathematical optimization problem, named multi-criteria optimization which involves more than one objective function to be optimized simultaneously. Multi-criteria optimization is an area of multiple criteria decision making. Although it is not implicated in this paper, the works of the literature (Amin et al. 2018a, b; Fahmi et al. 2018; Shakeel et al. 2018) are reviewed to study the respective employed methods in solving multi-criteria optimization problems.

The relatively popular swarm-based algorithms include ant colony optimization algorithm (ACO) (Colomi et al. 1991; Dorigo et al. 1999), particle swarm optimization (PSO) algorithm (Bai 2010; Eberhart and Kennedy 1995), artificial bee colony (ABC) algorithm (Basturk and Karaboga 2007; Karaboga 2005; Karaboga and Basturk 2008) and many more. The artificial fish swarm algorithm

Communicated by V. Loia.

✉ Junita Mohamad-Saleh
jms@usm.my

Weng-Hooi Tan
tanwenghooi@hotmail.com

¹ School of Electrical and Electronic Engineering, Engineering Campus, Universiti Sains Malaysia, 14300 Nibong Tebal, Penang, Malaysia

(AFSA) is one of the most recent swarm-based algorithms [originally proposed by Li et al. (2002)].

Naturally, fish swarm lives in an environment with greater food density. AFSA imitates the preying behavior process of fish swarm in the real environment. In this algorithm, each artificial fish (AF) is trained to take action according to the real-time situation. Each AF learns to perform four kinds of basic behaviors: follow, swarm, prey and random behaviors. Prey behavior is the foundation of the algorithm's convergence, swarm behavior enhances the stability and global convergence of the algorithm, follow behavior speeds-up the algorithm's convergence and the random behavior balances the contradiction of the other three behaviors. Input parameters given to each AF are *visual*, as the perception, and *step*, as the moving step length. AFs will gather the information and take a move using *visual* and *step*. These behaviors of AFs tend to influence each other's fitness as well.

The beneficial advantages of AFSA over other swarm-based algorithms are as follows: parallelism, simplification, strong global search ability, fast convergence and less sensitive to the requirements of the objective functions (Wang et al. 2016). With these advantages, AFSA has been successfully applied in a wide range of optimization problems.

Although many improvements and modifications have been proposed regularly on fish swarm algorithm, they mostly focused on balancing the process of exploration and exploitation, but not appointed for detailed navigation to guide toward accurate and precise direction. The fact is that the existing fish swarm algorithms have reached neither the real optimal solution nor an extremely good convergence rate.

In this paper, an improved algorithm, named normative fish swarm algorithm (NFSA), is proposed. NFSA hybridizes the characteristics of AFSA optimized by PSO with extended memory (PSOEM-FSA) (Duan et al. 2016) with the normative knowledge that was implemented in the cultural artificial fish swarm algorithm (CAFSA) (Wu et al. 2011). The normative knowledge is employed to prepare the complementary guidelines to be merged with the behavioral patterns of PSOEM-FSA. The generated guidelines lead the artificial fishes (AFs) to cite an extremely accurate and precise direction when they perform the modified search behaviors. The exchanging and sharing of information are employed in the population. NFSA further enhances the adaptive parameters to further balance the contradiction between the global search and local search abilities. The aim of the proposed algorithm is to obtain

better performance in terms of best optimal solution and convergence rate by solving the presented optimization benchmark functions. The specific contributions are as follows:

- i. Present a comprehensive survey on the optimization methods to solve the optimization benchmark functions with a focus to achieve better best optimal solution and convergence rate.
- ii. Introduce a normative communication behavior that was inspired by the behavioral pattern in PSOEM-FSA. Normative communication behavior is mainly employed to enhance the global search ability with a focus to provide detailed navigation to the candidates.
- iii. Introduce a normative memory behavior that has similar characteristics with normative communication behavior. Just that, normative communication behavior emphasizes the global search action, while normative memory behavior emphasizes the delicate local search.
- iv. Improve the adaptive parameters, mainly on the perceptions of *visual/step* and $visual_{min}/step_{min}$ along the iterative process, to solve the imbalance problem occurred in manipulating the process of exploration and exploitation.
- v. Introduce the parallelism in executing the behaviors of the proposed algorithm to make sure that every behavior is utilized in any given situation.

The proposed algorithm is simulated on ten nonlinear benchmark functions. Out of the ten benchmark functions, seven of them are the multimodal optimization functions while the other three are the unimodal optimization functions. Without a doubt, multimodal functions are much more difficult to be solved due to the higher possibility of candidate solutions being trapped inside the local optima. The parameters of NFSA are adopted from the literature (Mao et al. 2017). For the purpose of validation, a fair comparison with other comparative algorithms published in the literature (Mao et al. 2017) is employed. The results obtained by the proposed algorithm include the best optimum on each benchmark function, the means of the statistical data and the standard deviations. Collected results reveal that the NFSA is highly robust and precise due to the relatively superior standard deviations. NFSA outperforms other comparative algorithms on almost all the datasets. The convergence characteristic of NFSA was examined by comparing it with comparative algorithms. The convergence rate of NFSA was found to

be extremely outstanding. It can be deduced that the respective contributions solve their problems in either way of global or local search.

This paper is outlined as follows. Section 2 reviews the works of the literature, Sect. 3 lays out the proposed algorithm, Sect. 4 explains the experimental settings, Sect. 5 analyzes the computational results, and Sect. 6 describes the conclusion of this research.

2 Literature review

2.1 Standard artificial fish swarm algorithm (AFSA)

Recently, there have been plenty of researches done on AFSA, in various ways of modification and a variety of application. The works of Zhang et al. (2018); Zhou et al. (2018); and Zhu et al. (2018) have been cited to prove that AFSA tends to maintain its popularity in 2018.

Artificial fish swarm algorithm (AFSA) was inspired by the characteristic of fish swarm’s behavior in searching for a maximum food density. Let the environment where AF lives in is the problem space given by $[L, U]$ (Azizi 2014). Suppose that the state vector of the artificial fish swarm is $X = (X_1, X_2 \dots X_n)$, where $X_1, X_2 \dots X_n$ is the position of the population (Huang and Chen 2013) and n is the total number of artificial fishes (AFs) involved. The food concentration is determined by the object function $Y = f(X)$, where Y represents the fitness value at position X . Each AF has been given a perception, in term of *visual* to gather the information in order to seek for a better food solution and determine the current situation of other companions. *step* represents the maximum step size of an artificial fish (AF) (Huang and Chen 2013) to approach a certain targeted position. The other necessary parameters include crowding factor, *try_number* and iteration number t . In AFSA, each AF has been taught to perform different behaviors (i.e., follow, swarm, prey and random) according to its current situation.

Figure 1 shows the process flow of standard AFSA for a count of iteration. The entire behaviors of an AF have been mentioned, with the conditions stated. The pseudo code of AFSA is given in Sumathi et al. (2016).

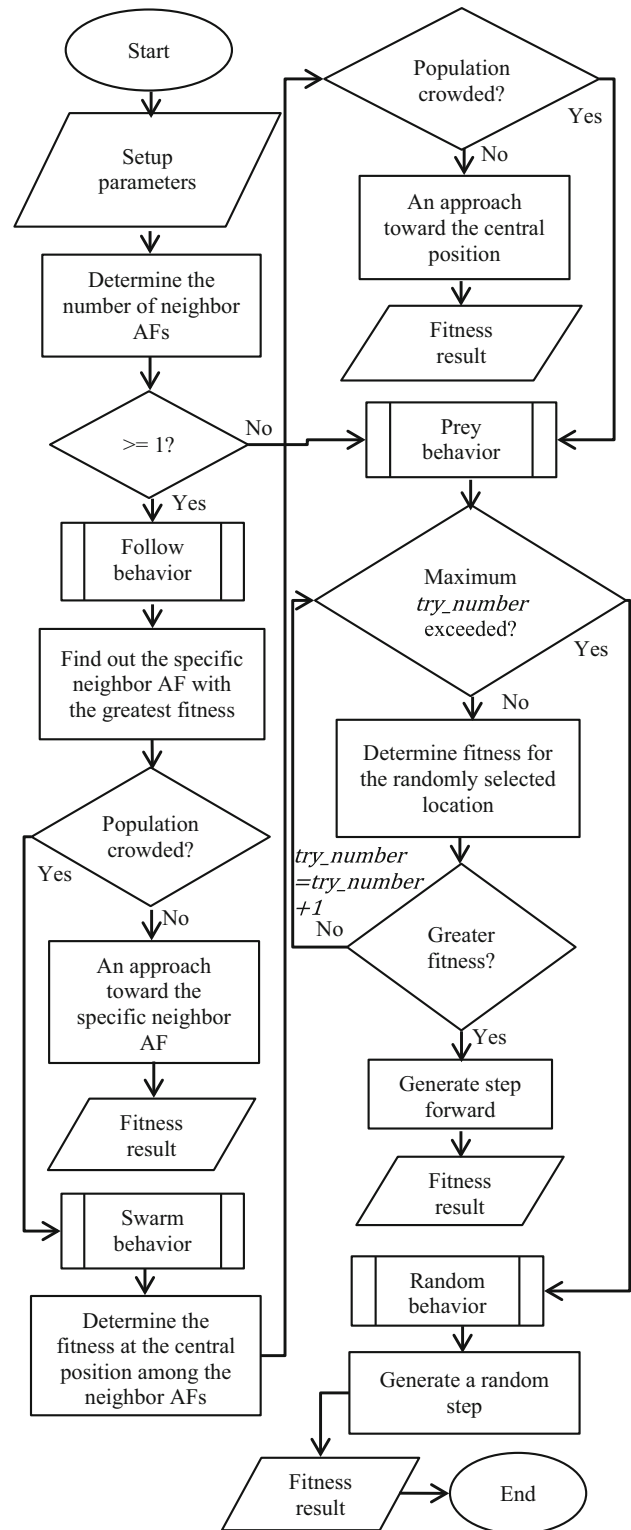


Fig. 1 Flowchart of AFSA

The pseudo code of standard AFSA

```

Set Up Parameters;
Random initialize Fish Population;
WHILE (is terminated by maximum iteration)
FOR (NumFish)
Determine Fitness of AFs' Current Position;
  DO AF_FOLLOW
  IF (AF_FOLLOW Fail) THEN
    DO AF_SWARM
    IF (AF_SWARM Fail) THEN
      DO AF_PREY
      IF (AF_PREY Fail) THEN
        DO AF_RANDOM
      END
    END
  END
END
NumFish+1;
END FOR
END WHILE
Output Optimal Solution

```

2.1.1 Follow behavior

The current position of i -th artificial fish (AF) is X_i , and the food concentration at this position is Y_i (Zhang et al. 2014). The total number of neighbor companions which fulfills the condition

$$d_{ij} < visual$$

is denoted as n_f value, where d_{ij} is the distance between the i -th AF and the j -th neighbor companion. As long as " $n_f > 0$," there is at least one neighbor companion located within its perceived *visual* and follow behavior is ready to be executed. Among the neighbor companions, the one with the best fitness Y_{\min} is referred to and its position is denoted as X_{\min} . If $\frac{Y_{\min}}{n_f} < \delta Y_i$, it indicates that the food density at X_{\min} is greater than at X_i . In this case, the i -th AF will chase the reliable companion using the following expression (Li et al. 2002):

$$X_i^{t+1} = X_i^t + \frac{X_{\min} - X_i^t}{|X_{\min} - X_i^t|} rand \times step \quad (1)$$

where X_i^t represents the position vector of i -th individual at the t -th iterative process.

2.1.2 Swarm behavior

Similar to follow behavior, the total number of neighbor companions which fulfills the condition

$$d_{ij} < visual$$

is denoted as n_f . If " $n_f > 0$," swarm behavior is ready to be executed. By including all neighbor companions, the center position of the swarm is calculated and denoted as X_C , which yield a center fitness value Y_C . If $\frac{Y_C}{n_f} < \delta Y_i$, it indicates the food density at X_C is greater than at X_i . In this case, i -th AF will follow the expression to swarm together (Li et al. 2002):

$$X_i^{t+1} = X_i^t + \frac{X_C - X_i^t}{|X_C - X_i^t|} rand \times step \quad (2)$$

2.1.3 Prey behavior

Without consulting to any information from companions, AF performs a random preying behavior. Random position X_j is selected within its perceived *visual* given by (Li and Qian 2003).

$$X_j = X_i^t + rand[-1, 1] \times visual \quad (3)$$

If $Y_j < Y_i$, it indicates a greater food density at X_j , and AF will move toward the selected position. The expression is given below (Li et al. 2002).

$$X_i^{t+1} = X_i^t + \frac{X_j - X_i^t}{|X_j - X_i^t|} rand \times step \quad (4)$$

In case of X_j fail to yield a better food solution, it performs again the preying behavior as long as the *try_number* does not reach its limit.

2.1.4 Random behavior

In general, random behavior is conducted after frustration in other behaviors. AF abides the expression to move a random step (Li et al. 2002):

$$X_i^{t+1} = X_i^t + rand \times step \quad (5)$$

2.2 AFSA optimized by PSO with extended memory (PSOEM-FSA)

Generally, PSOEM (Duang et al. 2011) upgraded PSO (Eberhart and Kennedy 1995) with the extension of memory, which employed the information at previous iterations. The expressions of PSOEM are denoted as (Duan et al. 2016):

$$X_{t+1} = X_t + v_{t+1} \quad (6)$$

and

$$\begin{aligned}
 \mathbf{v}_{t+1} = & \omega \mathbf{v}_t + \alpha_t^l [\zeta_t (\boldsymbol{\rho}_t^l - \mathbf{X}_t) + \zeta_{t-1} (\boldsymbol{\rho}_{t-1}^l - \mathbf{X}_{t-1})] \\
 & + \alpha_t^g [\zeta_t (\boldsymbol{\rho}_t^g - \mathbf{X}_t) + \zeta_{t-1} (\boldsymbol{\rho}_{t-1}^g - \mathbf{X}_{t-1})]
 \end{aligned} \tag{7}$$

where t denotes the index of iteration; \mathbf{v}_t represents the speed of the particle at the t -th iterative process, and hence, \mathbf{v}_{t+1} represents its updated speed vector; \mathbf{X}_t represents the position of the particle at the t -th iterative process, \mathbf{X}_{t+1} represents the updated position of the particle; $\boldsymbol{\rho}_t^l$ represents the current individual extreme value point of the particle at the t -th iterative process; $\boldsymbol{\rho}_{t-1}^l$ represents extreme value point of the particle at the $(t - 1)$ -th iterative process; $\boldsymbol{\rho}_t^g$ represents the current global extreme value point of the population at the t -th iterative process; $\boldsymbol{\rho}_{t-1}^g$ represents the global extreme value point of the population at the $(t - 1)$ -th iterative process; α_t^l and α_t^g are the acceleration factors; ω is known as the inertia weight; ζ_t is called current effective factor; ζ_{t-1} is called the effective factor of extended memory; $\sum \zeta = \zeta_t + \zeta_{t-1} = 1$.

Integrating the behaviors of the standard AFSA (Li et al. 2002) into PSOEM produced PSOEM-FSA. From the statements of PSOEM-FSA in the literature (Duan et al. 2016), $\boldsymbol{\rho}_t^g$ and $\boldsymbol{\rho}_t^l$ were extracted and combined exclusively with the fish swarm behavior. Hence, the communication behavioral pattern and memory behavioral pattern were introduced, respectively.

The updated speed vector of communication behavior in PSOEM-FSA is expressed as follows (Duan et al. 2016):

$$\begin{aligned}
 \mathbf{v}_{t+1} = & \omega \mathbf{v}_t + rand[0, 1] \\
 & \times step \left[\left(\frac{\zeta_t (\mathbf{X}_{gbest}^t - \mathbf{X}_i^t) + \zeta_{t-1} (\mathbf{X}_{gbest}^{t-1} - \mathbf{X}_i^{t-1})}{\left| \zeta_t (\mathbf{X}_{gbest}^t - \mathbf{X}_i^t) + \zeta_{t-1} (\mathbf{X}_{gbest}^{t-1} - \mathbf{X}_i^{t-1}) \right|} \right) \right]
 \end{aligned} \tag{8}$$

where \mathbf{v}_t represents the speed of the individual at the t -th iterative process; hence, \mathbf{v}_{t+1} represents its updated speed vector; \mathbf{X}_{gbest}^t represents the current global extreme value point of the population at the t -th iterative process; \mathbf{X}_{gbest}^{t-1} represents the global extreme value point of the population at the $(t - 1)$ -th iterative process; \mathbf{X}_i^t represents the current position vector of i -th individual at the t -th iterative process.

Communication behavioral pattern trains the artificial fishes (AFs) to swim while referring to the current and previous global optimal positions (i.e., \mathbf{X}_{gbest}^t and \mathbf{X}_{gbest}^{t-1}) of the entire communities. This helps to strengthen the ability in exchanging and sharing information between the individuals in the search process and further reduces the blindness of the fish in the search process (Duan et al. 2016).

On the other side, the updated speed vector of memory behavior is expressed as follows (Duan et al. 2016):

$$\begin{aligned}
 \mathbf{v}_{t+1} = & \omega \mathbf{v}_t + rand[0, 1] \\
 & \times step \left[\left(\frac{\zeta_t (\mathbf{X}_{lbest}^t - \mathbf{X}_i^t) + \zeta_{t-1} (\mathbf{X}_{lbest}^{t-1} - \mathbf{X}_i^{t-1})}{\left| \zeta_t (\mathbf{X}_{lbest}^t - \mathbf{X}_i^t) + \zeta_{t-1} (\mathbf{X}_{lbest}^{t-1} - \mathbf{X}_i^{t-1}) \right|} \right) \right]
 \end{aligned} \tag{9}$$

where \mathbf{X}_{lbest}^t represents the current individual extreme value point of the particle at the t -th iterative process; \mathbf{X}_{lbest}^{t-1} represents the global extreme value point of the population at the $(t - 1)$ -th iterative process.

Memory behavioral pattern trains the AFs to swim while referring to its own optimal positions (i.e., \mathbf{X}_{lbest}^t and \mathbf{X}_{lbest}^{t-1}). It definitely helps to reduce the blindness of the fish in the search process (Duan et al. 2016).

Communication behavioral pattern [Eq. (8)] mainly strengthens the global search ability, while memory behavioral pattern [Eq. (9)] mainly strengthens the local search ability. The implication of $(t - 1)$ -th iterative information indicates the extended memory used.

2.3 Normative knowledge in cultural artificial fish swarm algorithm (CAFSA)

CAFSA involves the implementation of normative knowledge, which describes the feasible solution space of an optimization problem (Wu et al. 2011). Normative knowledge is a set of promising variable range that provides standards for individual behaviors and guidelines within which individual adjustments can be made (Reynolds and Peng 2004). It is a set of information for each variable and is given by:

$$N = \mathbf{I}, \mathbf{U}, \mathbf{L}, \mathbf{D} \tag{10}$$

where \mathbf{U} , \mathbf{L} and \mathbf{D} are N -dimensional vectors and \mathbf{I}_k denotes the closed interval for variable $k \in (1, 2 \dots N)$, which is expressed as (Reynolds and Peng 2004):

$$\mathbf{I}_k = [l_k, u_k] = \{ \mathbf{X} | l_k \leq \mathbf{X} \leq u_k \} \tag{11}$$

where l_k and u_k are the lower and upper bounds of feasible space, respectively, for k -th variable. l_k and u_k are initialized with lower and upper bounds of the population (Wu et al. 2011) and are updated based on the following expressions (Wu et al. 2011):

$$l_k^{t+1} = \begin{cases} X_{i,k}, & X_{i,k} < l_k^t \text{ or } f(X_i^t) < L_k^t \\ l_k^t, & \text{otherwise} \end{cases} \quad (12)$$

$$u_k^{t+1} = \begin{cases} X_{j,k}, & X_{j,k} \geq u_k^t \text{ or } f(X_j^t) < U_k^t \\ u_k^t, & \text{otherwise} \end{cases}$$

where X represents the state (position) vector of the population, i -th individual affects the lower bound for variable k , j -th individual affects the upper bound for variable k , L_k^t and U_k^t are the values of the fitness function associated with the bounds l_k^t and u_k^t , respectively, at the t -th iteration (Wu et al. 2011). The fitness functions L_j^t and U_j^t are updated as follows (Wu et al. 2011):

$$L_k^{t+1} = \begin{cases} f(X_i^t), & X_{i,k} < l_k^t \text{ or } f(X_i^t) < L_k^t \\ L_k^t, & \text{otherwise} \end{cases} \quad (13)$$

$$U_k^{t+1} = \begin{cases} f(X_j^t), & X_{j,k} \geq u_k^t \text{ or } f(X_j^t) < U_k^t \\ U_k^t, & \text{otherwise} \end{cases}$$

Generally, the normative knowledge leads an AF to “jump” into a satisfactory range if it is not there yet (Reynolds and Peng 2004). In the normative knowledge, the parameters selected by the acceptance function are used to calculate the current acceptable interval for each parameter in the belief space. The idea is to be conservative when narrowing the interval and be progressive when widening the interval (Chung and Reynolds 1996). In CAFSA, normative knowledge is mainly used to adjust the current *visual* and *step* range to provide a suitable perception at every iterative process.

3 Proposed algorithm

In this section, a new normative fish swarm algorithm (NFSA) is proposed to combine the strengths of PSOEM-FSA (Duan et al. 2016) and normative knowledge. Transformed normative knowledge is used to find the complementary guidelines to be merged into the communication behavior and memory behavior to produce the newly proposed features referred to as normative communication behavior and normative memory behavior, respectively. In addition, NFSA improves the technique in adapting parameters, mainly on *visual/visual_{min}* and *step/step_{min}* along the iterative process. The improvements are ordinarily based on AFSA, PSOEM-FSA and CAFSA. The pseudo code of NFSA is given below.

The pseudo code of NFSA

```

Set Up Parameters;
Random Fish Population Initialization, AF_INIT
WHILE (is terminated by maximum iteration)
  FOR (NumFish)
    Determine Fitness of AFs' Current Position;
    DO AF_FOLLOW
    IF (AF_FOLLOW Fail) THEN
      DO AF_PREY
      IF (AF_PREY Fail) THEN
        DO AF_RANDOM
      END
    END
  END
  Output X1;
  DO AF_SWARM
  IF (AF_SWARM Fail) THEN
    DO AF_PREY
    IF (AF_PREY Fail) THEN
      DO AF_RANDOM
    END
  END
  END
  Output X2;
  DO AF_NORM_COMM
  IF (AF_NORM_COMM Fail) THEN
    DO AF_PREY
    IF (AF_PREY Fail) THEN
      DO AF_RANDOM
    END
  END
  END
  Output X3;
  DO AF_NORM_MEMORY
  IF (AF_NORM_MEMORY Fail) THEN
    DO AF_PREY
    IF (AF_PREY Fail) THEN
      DO AF_RANDOM
    END
  END
  END
  Output X4;
  Greedy Selection from X1, X2, X3 and X4;
  NumFish+1;
END FOR
Update Best Fitness
IF (Best Fitness Doesn't Upgrade after tlimit) THEN
  Gradient Decline visualmin;
  Gradient Decline stepmin;
END
END
Improved Adapt visual;
Improved Adapt step;
END WHILE
Output Optimal Solution

```

Figure 2 shows the flowchart of the proposed NFSA algorithm. It involves the following stages:

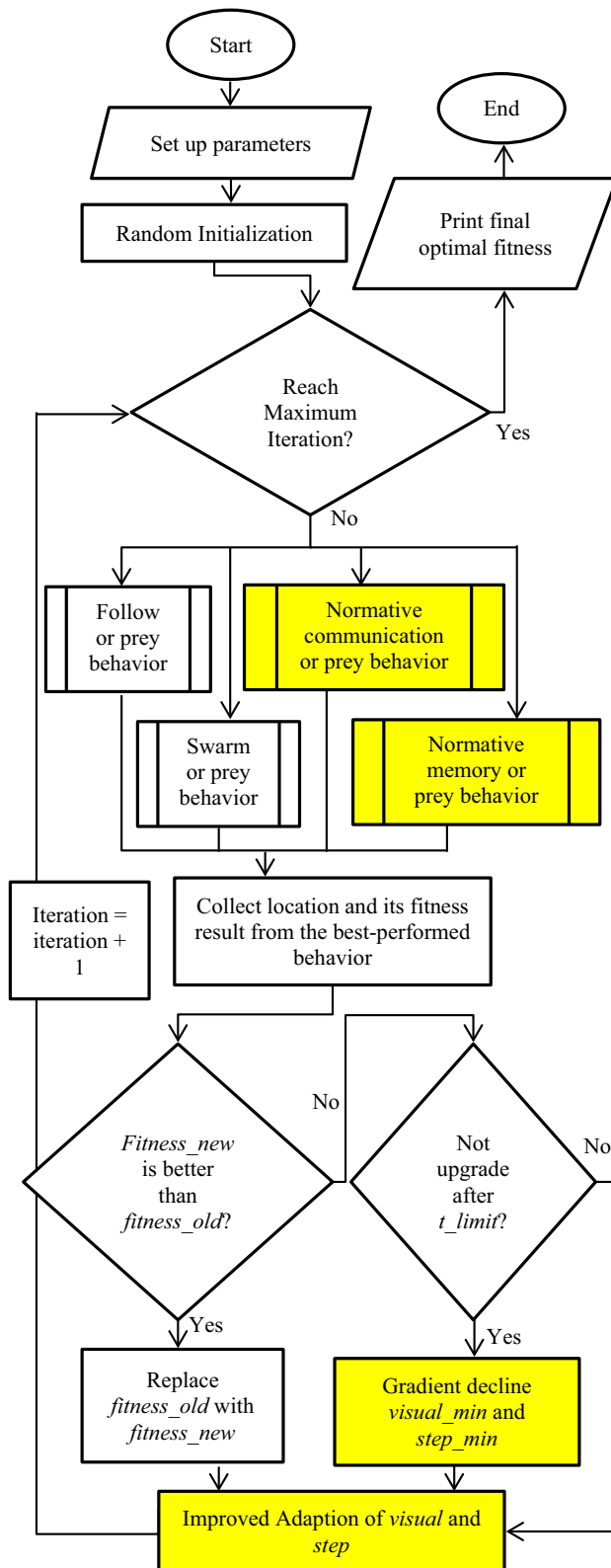


Fig. 2 Flowchart of NFSA

- Step 1 Random initialization of artificial fishes (AFs) population, the position of the fishes, the optimal locations of each fish’s memory and the global optimal position parameters.
- Step 2 Execute the four association behavior models which run in parallel:
 - (a) Follow or prey behavior.
 - (b) Swarm or prey behavior.
 - (c) Normative communication or prey behavior.
 - (d) Normative memory or prey behavior.
- Each behavior model yields distinct updated location and solution. The possible number of executions per iteration is decided by the given number of tries.
- Step 3 Apply greedy selection to select the optimal association behavior and update the current location of AFs.
- Step 4 Update the current global, local optima and current global, local extreme points in the bulletin board.
- Step 5 If the global optimal value is not upgraded for t_{limit} times, gradient decline the $visual_{min}$ and $step_{min}$, otherwise go to Step 6.
- Step 6 Adapt the $visual$ and $step$ parameters.
- Step 7 If the maximum iteration number is reached, the optimization process ends; otherwise, go back to Step 2.

The following sub-sections explain the structure of the newly designed features in NFSA. Sect. 3.1 explains normative communication behavior, Sect. 3.2 explains normative memory behavior, Sect. 3.3 describes the improved adaptive parameters and Sect. 3.4 depicts the structure of a bulletin board in NFSA.

3.1 Normative communication behavior (AF_NORM_COMM)

The global and local searches in PSOEM-FSA employed extended memory. AF tends to refer to the current and previous guidelines to decide on an accurate direction. Yet, some insufficiencies have been found in Eq. (8), regarding the communication behavior. Since the search direction is totally dependent upon the global extreme points X_{gbest}^t and X_{gbest}^{t-1} , it causes a lack of flexibility in the global search action. Hence, further improvements are proposed to enhance the communication behavioral pattern [Eq. (8)].

In the original communication behavioral pattern, the current location of AFs is updated by substituting the updated speed vector from Eq. (8) into Eq. (6). The connection of speed vector is removed and the current location

of AFs will be updated using only position vector features. In addition to that, the reconnaissance is executed around the current and previous global extreme points X_{gbest}^t and X_{gbest}^{t-1} to find the complementary guidelines (i.e., X_{cg}^t and X_{cg}^{t-1}) other than global optima. This helps AFs to cite an even more accurate direction and further reduces the blindness of AFs. Let the reconnaissance radius is set to be a variable range of feasible space $[l_j, u_j]$ based on the theorem of normative knowledge. To accommodate the enhanced behavioral pattern, some transformations are made regarding the variation of feasible space $[l_j, u_j]$. Yet, the idea is still the same, to be conservative when narrowing the search interval and be progressive when widening the search interval. l_j and u_j are initialized with initial lower and upper bounds, $l_j^{\text{initial},t=1}$ and $u_j^{\text{initial},t=1}$ respectively. Parameters l_j and u_j are updated as follows:

$$l_k^{t+1} = \begin{cases} \arg \min(X_k^{t+1}), & \arg \min(X_k^{t+1}) < l_k^t \\ l_k^t + |\Delta X_{\min,k}|, & \text{otherwise} \end{cases} \quad (14)$$

$$u_k^{t+1} = \begin{cases} \arg \max(X_k^{t+1}), & \arg \max(X_k^{t+1}) > u_k^t \\ u_k^t - |\Delta X_{\max,k}|, & \text{otherwise} \end{cases}$$

where k denotes the variable of dimension, $k \in (1, 2 \dots N)$; t denotes the index of iteration; X_k^{t+1} represents the position vector of the population for k -th dimension at the $(t + 1)$ -th iterative process; l_k^{t+1} is the lower bound for k -th dimension at the $(t + 1)$ -th iterative process; u_k^{t+1} is the upper bound for k -th dimension at the $(t + 1)$ -th iterative process; $\Delta X_{\min,k} = \arg \min(X_k^{t+1}) - l_k^t$; $\Delta X_{\max,k} = \arg \max(X_k^{t+1}) - u_k^t$.

Hence, the reconnaissance radius and the selective method of the complementary guidelines, X_{cg}^t and X_{cg}^{t-1} , are expressed as follows:

$$\text{radius}^t = \text{distance}[l^t, u^t] \quad (15)$$

and

$$X_{\text{cg}}^t = X_{\text{gbest}}^t + \text{rand}[-1, 1] \times \text{radius}^t$$

$$X_{\text{cg}}^t = \begin{cases} X_{\text{cg}}^t, & f(X_{\text{cg}}^t) < f(X_{\text{gbest}}^t) \\ X_{\text{gbest}}^t, & \text{otherwise} \end{cases} \quad (16)$$

and

$$X_{\text{cg}}^{t-1} = X_{\text{gbest}}^{t-1} + \text{rand}[-1, 1] \times \text{radius}^t$$

$$X_{\text{cg}}^{t-1} = \begin{cases} X_{\text{cg}}^{t-1}, & f(X_{\text{cg}}^{t-1}) < f(X_{\text{gbest}}^{t-1}) \\ X_{\text{gbest}}^{t-1}, & \text{otherwise} \end{cases} \quad (17)$$

where t denotes the index of iteration; X_{cg}^t and X_{cg}^{t-1} are the selected guidelines within the reconnaissance radius from X_{gbest}^t and X_{gbest}^{t-1} , respectively; X_{gbest}^t represents the current

global extreme value point of the population at the t -th iterative process; X_{gbest}^{t-1} represents the global extreme value point of the population at the $(t - 1)$ -th iterative process; X_i^t represents the current position vector of i -th individual at the t -th iterative process.

As for normative communication behavioral pattern, the updated position vector X_i^{t+1} can be presented as follows:

$$X_i^{t+1} = X_i^t + \text{rand}[0, 1] \times \text{step} \left[\beta_1 \left(\frac{\xi_t (X_{\text{cg}}^t - X_i^t) + \xi_{t-1} (X_{\text{cg}}^{t-1} - X_i^{t-1})}{|\xi_t (X_{\text{cg}}^t - X_i^t) + \xi_{t-1} (X_{\text{cg}}^{t-1} - X_i^{t-1})|} \right) + \beta_2 \left(\frac{\xi_t (X_{\text{gbest}}^t - X_i^t) + \xi_{t-1} (X_{\text{gbest}}^{t-1} - X_i^{t-1})}{|\xi_t (X_{\text{gbest}}^t - X_i^t) + \xi_{t-1} (X_{\text{gbest}}^{t-1} - X_i^{t-1})|} \right) \right] \quad (18)$$

where β_1 and β_2 are known as the speed factors; ξ_t is denoted as the current effective factors; ξ_{t-1} is denoted as the effective factor of extended memory; $\sum \beta = \beta_1 + \beta_2 = 1$; $\sum \xi = \xi_t + \xi_{t-1} = 1$.

3.2 Normative memory behavior (AF_NORM_MEMORY)

Normative memory behavior is quite similar to normative communication behavior. Just that, normative communication behavior emphasizes the global search action, while normative memory behavior emphasizes the delicate local search. Some weaknesses have been found in Eq. (9) as well. Similar to the communication behavior, the search direction in memory behavior is totally dependent upon the local extreme point X_{lbest}^t and X_{lbest}^{t-1} . This leads to a lack of flexibility in the local search action. Hence, this work further improves the memory behavioral pattern [i.e., Eq. (9)].

In the original memory behavioral pattern, the current location of AFs is updated by substituting the updated speed vector from Eq. (9) into Eq. (6). The connection of speed vector is proposed to be removed and the inertia weight ω is erased. The reconnaissance is executed around the current and previous local extreme points X_{lbest}^t and X_{lbest}^{t-1} , respectively, to find the complementary guidelines other than individuals' own optima. This helps the artificial fishes (AFs) to cite an even more precise direction and further increase the local search's efficiency. Let the reconnaissance radius be a variable range of feasible space $[l_j, u_j]$ and is expressed the same as Eq. (15). l_j and u_j are initialized with initial lower and upper bounds, $l_j^{\text{initial},t=1}$ and $u_j^{\text{initial},t=1}$, respectively. Parameters l_j and u_j are updated based on Eq. (14).

Hence, the selective method of new guidelines, X_{cl}^t and X_{cl}^{t-1} , are expressed as follows:

$$X_{cl}^t = X_{lbest}^t + rand[-1, 1] \times radius^t$$

$$X_{cl}^t = \begin{cases} X_{cl}^t, & f(X_{cl}^t) < f(X_{lbest}^t) \\ X_{lbest}^t, & \text{otherwise} \end{cases} \quad (19)$$

and

$$X_{cl}^{t-1} = X_{lbest}^{t-1} + rand[-1, 1] \times radius^t$$

$$X_{cl}^{t-1} = \begin{cases} X_{cl}^{t-1}, & f(X_{cl}^{t-1}) < f(X_{lbest}^{t-1}) \\ X_{lbest}^{t-1}, & \text{otherwise} \end{cases} \quad (20)$$

where X_{cl}^t and X_{cl}^{t-1} are the new selected guidelines in normative memory behavior; X_{lbest}^t represents the current individual extreme value point of the particle at the t -th iterative process; X_{lbest}^{t-1} represents the individual extreme value point of the population at the $(t - 1)$ -th iterative process

As for normative memory behavioral pattern, the updated position vector X_i^{t+1} can be presented as follows:

$$X_i^{t+1} = X_i^t + rand[0, 1]$$

$$\times step \left[\beta_1 \left(\frac{\zeta_t (X_{i,cl}^t - X_i^t) + \zeta_{t-1} (X_{i,cl}^{t-1} - X_i^{t-1})}{|\zeta_t (X_{i,cl}^t - X_i^t) + \zeta_{t-1} (X_{i,cl}^{t-1} - X_i^{t-1})|} \right) \right]$$

$$+ \beta_2 \left(\frac{\zeta_t (X_{i,lbest}^t - X_i^t) + \zeta_{t-1} (X_{i,lbest}^{t-1} - X_i^{t-1})}{|\zeta_t (X_{i,lbest}^t - X_i^t) + \zeta_{t-1} (X_{i,lbest}^{t-1} - X_i^{t-1})|} \right) \quad (21)$$

where X_{cl}^t and $X_{i,cl}^{t-1}$ are the new selected guidelines of i -th individual; $X_{i,lbest}^t$ represents the current individual extreme value point of i -th individual at the t -th iterative process; $X_{i,lbest}^{t-1}$ represents the individual extreme value point of i -th individual at the $(t - 1)$ -th iterative process; X_i^t represents the current position vector of i -th individual at the t -th iterative process.

3.3 Improved adaptive parameters

In the standard AFSA, *visual* and *step* remain unchanged throughout the iterative processes. If the given initial parameters of *visual* and *step* are large, artificial fish swarm move faster toward the global optima, due to greater *visual* search for the larger environment and hence move with bigger *step* (Azizi 2014). This way, artificial fish are more capable of escaping from local optima traps (Azizi 2014). However, at the same time, it reduces the accuracy of global search optimization as large *visual* and *step* are proficient in approaching the global optimal region, but not in performing the accurate local search. On the other hand,

by using small *visual* and *step*, it is capable to carry on with an improved local search but has to compromise the convergence speed to reach the global optima.

To balance the contradiction between the global search ability and local search ability of AFSA, the adaptive method utilized in the literature (Li and Qian 2003) is reviewed. In the early stage, relatively large *visual* and *step* are adopted to enhance the global search ability and convergence speed of the algorithm (Zhang et al. 2014). Along the iterations, these parameters are decreased based on the following equations:

$$\begin{cases} visual^{t+1} = visual^t - visual^t * \lambda + visual_{min} \\ step^{t+1} = step^t - step^t * \lambda + step_{min} \\ \lambda = \exp \left(- \frac{\sigma}{\sqrt[4]{(t_{max})^3}} * (t_{max} - t) \right) \end{cases} \quad (22)$$

where $visual_{min}$ is the minimum *visual* value; $step_{min}$ is the minimum *step* value; $t \in (1, 2, \dots, t_{max})$; $0.5 < \sigma < 1$ at any stage. Hence, at the later stage, it improves the local search of the algorithm.

From the expression in Eq. (22), it can be obviously deduced that the *visual* and *step* are gradually decreasing along the iterations and eventually drop to the value approximately equals to $visual_{min}$ and $step_{min}$, respectively. Without a doubt, $visual_{min}$ and $step_{min}$ play the role to limit the iterative variation of *visual* and *step*, respectively. Without the presence of $visual_{min}$ and $step_{min}$, *visual* and *step* will definitely drop to an approximately zero value at a later stage. If this happens, AFs will definitely lose the ability to perform any further local search, as they fail to perform any significant move.

Yet, it is still problematic to decide the values of $visual_{min}$ and $step_{min}$. If large $visual_{min}$ and $step_{min}$ are set, it may affect the accuracy of local search, but small $visual_{min}$ and $step_{min}$ may consume more time to reach the exact optimum. Hence, the $visual_{min}$ and $step_{min}$ are proposed to be gradient declined under a certain condition, where the optimum remains the same without upgrading after a given limitation of iterations t_{limit} . The declinations of both parameters are presented as follows:

$$visual_{min}^{t+1} = \frac{visual_{min}^t}{0.5 \times \sqrt[3]{t_{max}}} \quad (23)$$

and

$$step_{min}^{t+1} = \frac{step_{min}^t}{\sqrt[3]{t_{max}}} \quad (24)$$

where $t \in (1, 2, \dots, t_{max})$ is the index number of iteration.

3.4 Bulletin board

Bulletin board is a terminal for collecting data and sharing among all the users. Generated n AFs are the users of the bulletin board. It basically records the global optima Y_{gbest} , together with its global extreme position vector X_{gbest} and the local optima Y_{lbest} , together with its local extreme position vector X_{lbest} . The data are updated along the iterations, as long as they found a better optimal solution. In other words, if $Y_i^{t+1} = \arg \min[Y^{t+1}] < Y_{gbest}^t$ at the t -th iterative process, Y_{gbest}^{t+1} and X_{gbest}^{t+1} are replaced by Y_i^{t+1} and X_i^{t+1} respectively, where $i \in (1, 2 \dots n)$.

4 Experimental setting

Although the algorithm is made to be user dependent, for the purpose of validation and a fair comparison with other works, the employed parameters are adopted from the literature (Mao et al. 2017). The parameter settings for NFSA are displayed in Table 1.

Ten optimization benchmark functions are used to evaluate the performance of NFSA. Evaluations have been carried out using maximum iteration numbers, $t_{max} = 1000$. The proposed algorithm is proposed to run 10 times independently on each benchmark function, in order to ensure that the best optimal solution, mean and standard deviation (SD) of each function are obtained.

The representation of the employed benchmark functions are as follows:

1. $f_1 = f_{Ackley}(x)$
2. $f_2 = f_{Alipine}(x)$
3. $f_3 = f_{Griewank}(x)$
4. $f_4 = f_{Levy}(x)$
5. $f_5 = f_{Quadric}(x)$
6. $f_6 = f_{Rastrigin}(x)$
7. $f_7 = f_{Rosenbrock}(x)$
8. $f_8 = f_{Elliptic}(x)$
9. $f_9 = f_{Sphere}(x)$
10. $f_{10} = f_{SumPower}(x)$

5 Computational results and discussion

5.1 Assessment of NFSA

The evaluation of NFSA mainly tests the contributions of proposed normative communication behavior, normative memory behavior and improved adaption of parameters. Table 2 displays the best optimum, mean and standard deviation (SD) obtained by NFSA on each benchmark function in the case of $N = 10$. It has to be noted that the closer the data values with reference to its global minimum, the better the performance of NFSA on a specific function. Table 2 shows the relatively superior results from NFSA on every benchmark function, especially Rastrigin function (f_6) which obtained the real global minimum. The fact that the proposed NFSA tends to reach the real global minimum on certain benchmark functions, and shows the relatively superior results on all the benchmark functions, certifies the contributions of newly proposed features in NFSA. The relatively small standard deviations (SD) reveal that NFSA is highly robust and precise.

Table 1 Parameter settings for NFSA

Parameter	Values
No. of tries, <i>try_number</i>	5
<i>visual</i>	$1.25 \times (L - U)$
<i>step</i>	$0.9 \times Visual$
Crowding factor, δ	0.75
Dimension, N	10
Fish population, n	50
ξ_t	0.5
ξ_{t-1}	0.5
β_1	0.5–1
β_2	$1 - \beta_1$
MaxD	$\sqrt{(L - U)^2 \times N}$
<i>visual</i> _{min}	$\frac{MaxD}{500}$
<i>step</i> _{min}	$\frac{MaxD}{500}$
t_{limit}	$0.02 \times t_{max}$

Table 2 Performance evaluation of NFSA

Function	Best	Mean	SD
f_1	8.88E–16	3.73E–15	1.50E–15
f_2	4.11E–34	1.29E–26	3.56E–26
f_3	9.86E–03	3.49E–02	1.68E–02
f_4	1.50E–32	1.50E–32	0
f_5	2.10E–14	4.75E–09	9.13E–09
f_6	0	0	0
f_7	0	1.94E–30	6.14E–30
f_8	2.95E–48	1.09E–46	1.58E–46
f_9	1.14E–55	1.14E–52	1.64E–52
f_{10}	1.00E–197	7.10E–179	0

Table 3 NFSA performance comparison with related algorithms on benchmark functions

Ind.	Variants			
	NFSA	PSOEM-FSA (Mao et al. 2017)	CIAFSA (Mao et al. 2017)	AFSA (Mao et al. 2017)
<i>f</i> ₁				
Best	8.88E-16	1.68E-07	1.79E-06	2.50E-04
Mean	3.73E-15	1.02E-04	3.56E-04	3.09E-02
SD	1.50E-15	1.11E-04	3.95E-04	1.94E-02
<i>f</i> ₂				
Best	4.11E-34	4.79E-07	1.94E-06	6.41E-08
Mean	3.56E-26	6.87E-06	8.98E-06	4.14E-06
SD	9.75E-08	5.39E-06	1.18E-05	3.62E-06
<i>f</i> ₃				
Best	9.86E-03	2.94E-08	7.47E-06	2.49E-09
Mean	3.49E-02	1.07E-01	1.52E-02	7.11E-06
SD	1.68E-02	1.84E-01	3.47E-01	7.57E-05
<i>f</i> ₄				
Best	1.50E-32	4.79E-07	1.94E-06	1.38E-08
Mean	1.50E-32	6.87E-06	8.98E-06	2.34E-05
SD	0	5.39E-06	1.18E-05	2.63E-05
<i>f</i> ₅				
Best	2.10E-14	1.16E-08	9.04E-05	1.98E-08
Mean	4.75E-09	1.16E-04	9.34E+00	5.44E-05
SD	9.13E-09	2.63E-05	2.11E+01	1.29E-04
<i>f</i> ₆				
Best	0	3.09E-10	4.56E-09	3.20E-10
Mean	0	8.34E-06	2.06E-06	2.38E-07
SD	0	2.40E-05	4.55E-06	4.44E-07
<i>f</i> ₇				
Best	0	1.99E-14	2.54E-14	5.65E-12
Mean	1.94E-30	6.55E-10	1.45E-05	4.26E-08
SD	6.14E-30	1.87E-10	3.81E-05	7.70E-08
<i>f</i> ₈				
Best	2.95E-48	1.03E-02	2.18E+01	1.61E-07
Mean	1.09E-46	5.73E+00	8.40E+03	3.29E-02
SD	1.58E-46	1.04E+01	3.51E+03	6.57E-02
<i>f</i> ₉				
Best	1.14E-55	3.08E-13	1.00E-12	1.58E-11
Mean	1.14E-52	6.19E-08	6.51E-03	1.21E-07
SD	1.64E-52	1.11E-07	2.59E-03	1.84E-07
<i>f</i> ₁₀				
Best	1.00E-197	1.43E-15	4.57E-13	2.94E-14
Mean	7.10E-179	9.70E-10	1.72E-09	6.47E-12
SD	0	1.69E-09	3.47E-09	1.38E-11

5.2 Comparison of NFSA with related algorithms

The results of the performance of PSOEM-FSA (Duan et al. 2016), CIAFSA (Mao et al. 2017) and AFSA (Li et al. 2002) on *f*₁₋₁₀ are adopted from the literature (Mao et al.

2017). As observed from Table 3, data obtained by NFSA (with references to Table 2) were compared with these related algorithms, using the same maximum number of iterations *t*_{max} in the case of dimension *N* = 10.

As explained, PSOEM-FSA is actually the predecessor of CIAFSA. Hence, both algorithms exhibit the same characteristics of communication behavior and memory behavior which is the origin of the proposed features in NFSA—normative communication behavior and normative memory behavior. CIAFSA is proposed after PSOEM-FSA, with the additional feature of improved adaptive *visual* and *step*. They are comparatively suited to be used to test the contributions of normative communication behavior and normative memory behavior.

Some insightful conclusions can be drawn from Table 3. The bold and highlighted data represent the best results on each function among different algorithms. Apart from Griewank function (f_3), the proposed NFSA has clearly outperformed PSOEM-FSA, CIAFSA, and AFSA. Especially on Ackey, Alipine, Levy, Rastrigin, Rosenbrock, Elliptic, Sphere and SumPower functions ($f_1, f_2, f_4, f_6, f_7, f_8, f_9, f_{10}$), NFSA achieved relatively superior best optima and means over other algorithms, affirming that NFSA tends to approach closer to the respective global minimum. The relatively smaller standard deviations (SD) also prove that NFSA is way more precise than other comparative algorithms.

Figures 3 and 4 display the convergence characteristics of NFSA and other related algorithms on f_1 – f_{10} . The convergence rate of the algorithm is tightly related and influenced by *visual* and *step* variations. Due to the improved technique of adaption of *visual* and *step* in NFSA, the convergence rates become more superior to other related algorithms on most tested benchmark functions. The truth is that NFSA tends to strike closer to the global minimum, as shown in Table 2, Figs. 3 and 4.

5.3 Statistical test

Student t test is a statistical test which is widely used to compare the means of two groups of samples. It is, therefore, to evaluate whether the means of the two sets of data are statistically significantly different from each other. The t test can be commonly distributed into three types:

- The one-sample t test used to compare the mean of a population with a theoretical value.
- The unpaired two-sample t test used to compare the mean of two independent samples.
- The paired t test used to compare the means between two related groups of samples.

Using the mean (\bar{x}), standard deviation (SD) and the number of population (n) for each sample from Tables 1 and 3, the independent two-sample t test is performed. Assuming all the conditions for inference are met and using the significant level of 0.05, it is reasonable to assume that two populations have the same standard deviation as the

sample standard deviations of the algorithms are reasonably close. An alternative procedure known as the pooled t procedure will be used.

Considering the t test between NFSA and CIAFSA on Ackley benchmark function (f_1) as an example, the datasets are summarized in Table 4.

denotes the population means. The hypothesis difference between the population means is set to 0. The null hypothesis is then constructed as follows:

$$H_0 : \mu_{NFSA} = \mu_{CIAFSA} \tag{25}$$

The alternate hypothesis is written as follows:

$$H_a : \mu_{NFSA} < \mu_{CIAFSA} \tag{26}$$

Based on the sample data in Table 4, our t statistic is equal to:

$$\begin{aligned} t &= \frac{\bar{x}_{NFSA} - \bar{x}_{CIAFSA}}{s_p \sqrt{\frac{1}{n_{NFSA}} + \frac{1}{n_{CIAFSA}}}} \\ &= \frac{3.73 \times 10^{-15} - 3.56 \times 10^{-04}}{2.79307 \times 10^{-04} \sqrt{\frac{1}{50} + \frac{1}{50}}} = -6.373 \end{aligned}$$

where s_p is calculated by using the following expression:

$$\begin{aligned} s_p &= \sqrt{\frac{(n_{NFSA} - 1)(SD_{NFSA})^2 + (n_{CIAFSA} - 1)(SD_{CIAFSA})^2}{n_{NFSA} + n_{CIAFSA} - 2}} \\ &= \sqrt{\frac{(50 - 1)(1.50 \times 10^{-15})^2 + (50 - 1)(3.95 \times 10^{-04})^2}{50 + 50 - 2}} \\ &= 2.79307 \times 10^{-04} \end{aligned}$$

Next step is to obtain the p value. p value is the probability of obtaining the observed difference between the samples if the null hypothesis were true. The elements used to achieve the p value are the t statistic and degree of freedom df , where df is expressed as follows:

$$df = n_{NFSA} + n_{CIAFSA} - 2 = 50 + 50 - 2 = 98$$

The easiest way to determine the p value is by referring to the p value table or using the p value from t score calculator. The one-tailed (left-tailed) hypothesis is employed. The obtained p value is <0.0001 . The area of the reddish region in Fig. 5 indicates the p value.

Assuming the null hypothesis is true, and the p value that is rather a low probability which is below the significant level of 0.05, the null hypothesis should be rejected. Since the null hypothesis is rejected, the alternative hypothesis is suggested, where the estimated population mean of NFSA is superior to CIAFSA ($\mu_{NFSA} < \mu_{CIAFSA}$).

Some insightful conclusions can be drawn from Table 5. The “*rejected*” null hypotheses H_0 in Table 5 demonstrate convincingly that the population means of NFSA are predicted to be superior in comparison with respective

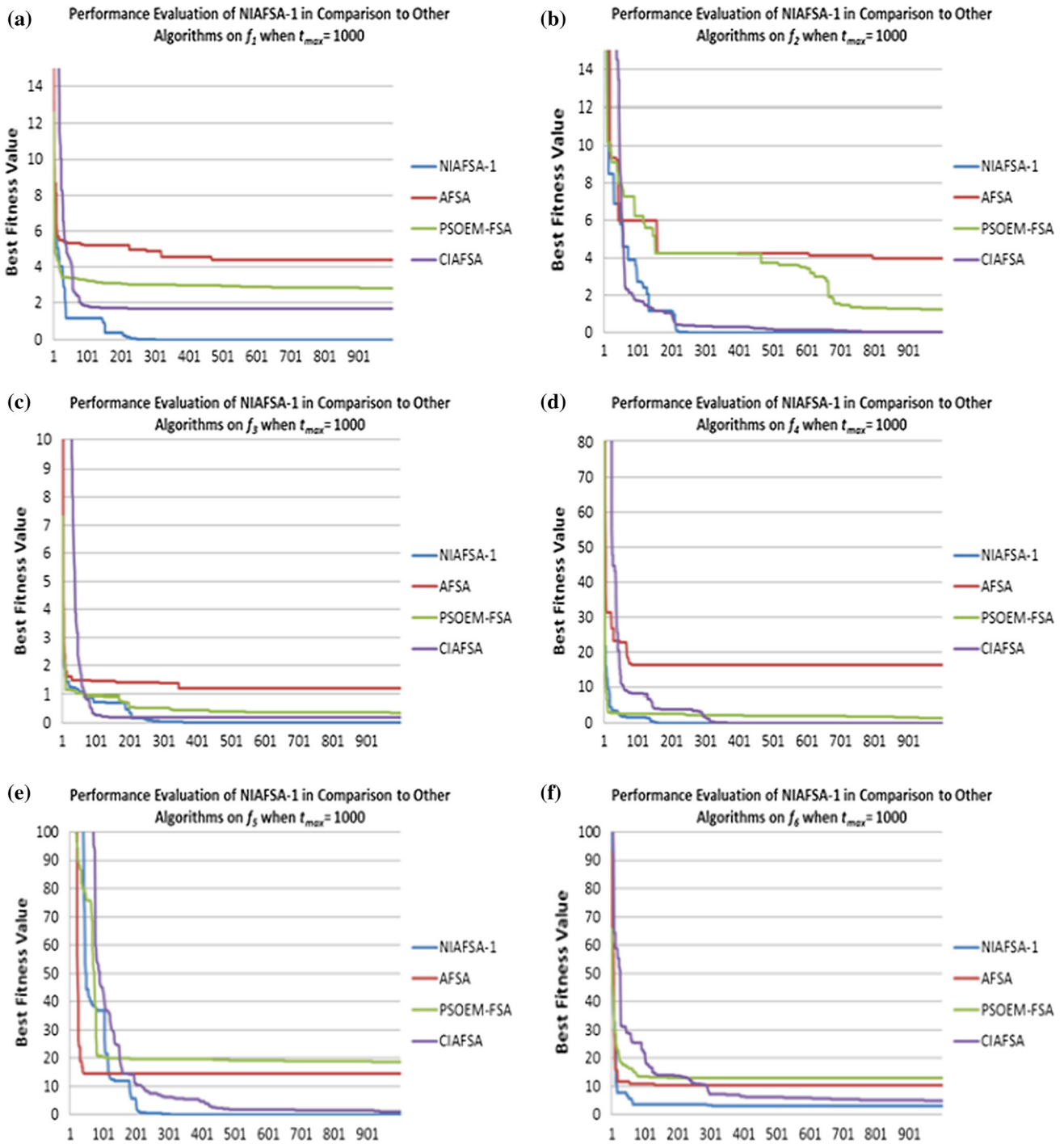


Fig. 3 Convergence characteristics of NFA in comparison with a f_1 to f_6 benchmark functions

comparative algorithms on specific benchmark function, according to the statement of alternative hypothesis H_a in Eq. (26). Most results display the “rejected” state for the null hypotheses, except on the Griewank function (f_3), where the null hypotheses are “accepted” in the t test between NFA and CIAFSA, and between NFA and AFSA, indicating the failures of NFA to overcome

CIAFSA and AFSA on Griewank function (f_3) in terms of the population means. Yet, based on the overall results in Table 5, it can be deduced that overall, the population of NFA are fitter in comparison with the population of other comparative algorithms.

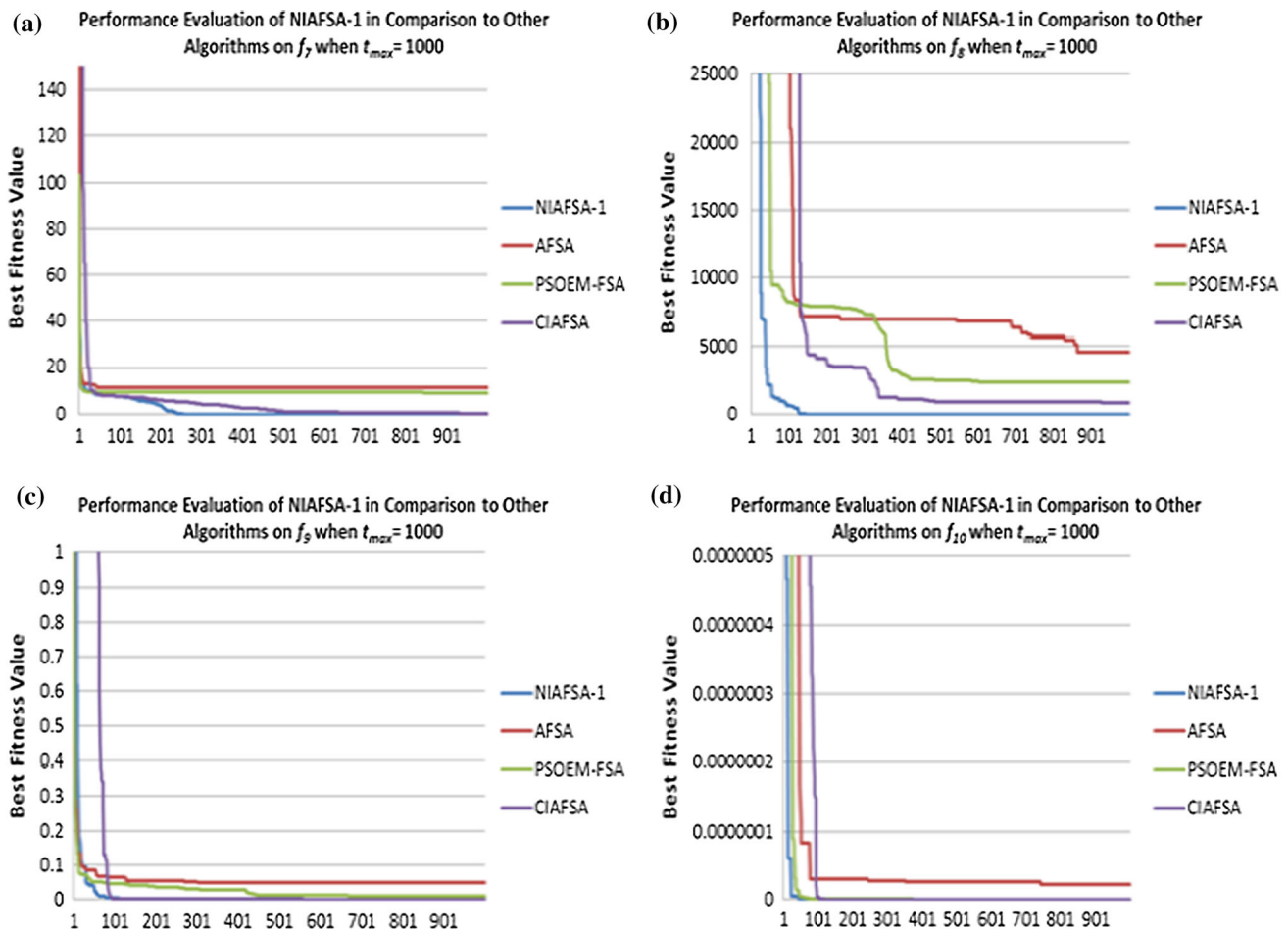


Fig. 4 Convergence characteristics of NFSA in comparison with a f_7 to d f_{10} benchmark functions

Table 4 Summarized data as taken from Tables 1 and 3

Ind.	NFSA	CIAFSA (Mao et al. 2017)
$f_1 \bar{x}$	3.73E-15	3.56E-04
SD	1.50E-15	3.95E-04
n	50	50

6 Conclusion

Although many optimization algorithms have been introduced, no matter what kind of solutions they are, none of them could claim that they have perfectly solved the respective optimization problems. Somehow it has become a challenging topic for many researchers.

This work had successfully developed the proposed normative fish swarm algorithm (NFSA) for optimization. Numerical simulation experiments were performed to compare its performance with the predecessors of the proposed algorithm: PSOEM-FSA and CIAFSA. The

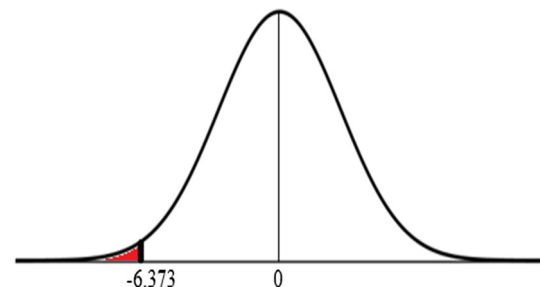


Fig. 5 The normalized t distribution sample

experimental results showed that NFSA outperformed its predecessors and standard AFSA in terms of convergence speed and global optimum achievement. The real global optima were achieved for most of the benchmark functions used, confirming the contributions attained through the modifications done on existing optimization algorithms. The results bring researchers one step closer toward more efficient solution for solving optimization problems.

Table 5 Results obtained after performing two-sample *t* test between NFSA and respective comparative algorithms

Ind.	A two-sample <i>t</i> test between NFSA and:		
	PSOEM-FSA (Mao et al. 2017)	CIAFSA (Mao et al. 2017)	AFSA (Mao et al. 2017)
f_1			
<i>t</i> statistic	− 6.498	− 6.373	− 11.263
<i>p</i> value	< 0.0001	< 0.0001	< 0.0001
Null hypothesis H_0	Rejected	Rejected	Rejected
f_2			
<i>t</i> statistic	− 9.011	− 5.381	− 8.084
<i>p</i> value	< 0.0001	< 0.0001	< 0.0001
Null hypothesis H_0	Rejected	Rejected	Rejected
f_3			
<i>t</i> statistic	− 2.759	0.401	14.686
<i>p</i> value	0.0035	0.6554	>99.9999
Null hypothesis H_0	Rejected	Accepted	Accepted
f_4			
<i>t</i> statistic	− 9.013	− 5.381	− 6.291
<i>p</i> value	< 0.0001	< 0.0001	< 0.0001
Null hypothesis H_0	Rejected	Rejected	Rejected
f_5			
<i>t</i> statistic	− 31.187	− 3.13	− 2.982
<i>p</i> value	< 0.0001	0.0012	0.0018
Null hypothesis H_0	Rejected	Rejected	Rejected
f_6			
<i>t</i> statistic	− 2.457	− 3.201	− 3.79
<i>p</i> value	0.0079	0.0009	0.0001
Null hypothesis H_0	Rejected	Rejected	Rejected
f_7			
<i>t</i> statistic	− 24.768	− 2.691	− 3.912
<i>p</i> value	< 0.0001	0.0042	< 0.0001
Null hypothesis H_0	Rejected	Rejected	Rejected
f_8			
<i>t</i> statistic	− 3.896	− 7.071	− 3.541
<i>p</i> value	< 0.0001	< 0.0001	0.0003
Null hypothesis H_0	Rejected	Rejected	Rejected
f_9			
<i>t</i> statistic	− 3.943	− 17.773	− 4.65
<i>p</i> value	< 0.0001	< 0.0001	< 0.0001
Null hypothesis H_0	Rejected	Rejected	Rejected
f_{10}			
<i>t</i> statistic	− 4.059	− 3.505	− 33.152
<i>p</i> value	< 0.0001	0.0003	< 0.0001
Null hypothesis H_0	Rejected	Rejected	Rejected

Other optimization problems can be solved in the future to ensure the capability of the proposed algorithm in this domain. Moreover, other powerful local search strategies can be employed to further improve the exploitation capability of fish swarm algorithm.

Acknowledgements This research is supported by the Ministry of Higher Education (MOHE) Malaysia Fundamental Research Grant Scheme (Grant no. 203.PELECT.6071317).

Author contributions W-HT conceived, developed and tested the proposed algorithm, analyzed the data, and wrote this manuscript.

JM-S verified the analytical methods and supervised the findings of this work. Both authors read and approved the final manuscript.

Funding This study was financially funded by the Ministry of Higher Education (MOHE) Malaysia Fundamental Research Grant Scheme (Grant No. 203.PELECT.6071317).

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflicting interest.

Ethics approval This research does not contain any studies with human participants or animals performed by any of the authors.

Consent to participate Informed consent was obtained from all individual participants included in the study.

References

- Abualigah LM, Khader AT, Hanandeh ES (2018a) A combination of objective functions and hybrid krill herd algorithm for text document clustering analysis engineering applications of artificial intelligence a combination of objective functions and hybrid krill herd algorithm for text document clusterin. *Eng Appl Artif Intell* 73:111–124. <https://doi.org/10.1016/j.engappai.2018.05.003>
- Abualigah LM, Khader AT, Hanandeh ES (2018b) Hybrid clustering analysis using improved krill herd algorithm. *Appl Intell* 48(11):4047–4071. <https://doi.org/10.1007/s10489-018-1190-6>
- Amin F, Fahmi A, Abdullah S (2018a) Dealer using a new trapezoidal cubic hesitant fuzzy TOPSIS method and application to group decision-making program. *Soft Comput*. <https://doi.org/10.1007/s00500-018-3476-3>
- Amin F, Fahmi A, Abdullah S, Ali A, Ahmad R, Ghanu F (2018b) Triangular cubic linguistic hesitant fuzzy aggregation operators and their application in group decision making. *J Intell Fuzzy Syst* 34(1):1–15. <https://doi.org/10.3233/JIFS-171567>
- Azizi R (2014) Empirical study of artificial fish swarm algorithm. *Int J Comput Commun Netw* 3(1):1–7
- Bai Q (2010) Analysis of particle swarm optimization algorithm. *Comput Inf Sci* 3(1):180–184. <https://doi.org/10.5539/cis.v3n1p180>
- Basturk B, Karaboga D (2007) A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm. *J Glob Optim* 39(3):459–471. <https://doi.org/10.1007/s10898-007-9149-x>
- Chung CJ, Reynolds RG (1996) A Testbed for solving optimization problems using cultural algorithms. In: *Evolutionary programming V: proceedings of the fifth annual conference on evolutionary programming*. San Diego, CA, pp 225–236
- Colomi A, Dorigo M, Maniezzo V (1991) Distributed optimization by ant colonies. In: *Proceedings of the first European conference on artificial life*. Paris, France, pp 134–142
- Dorigo M, Caro GD, Gambardella LM (1999) Ant algorithms for discrete optimization. *Artif Life* 5(2):137–172. <https://doi.org/10.1162/106454699568728>
- Duan Q, Mao M, Duan P, Hu B (2016) An improved artificial fish swarm algorithm optimized by particle swarm optimization algorithm with extended memory. *Kybernetes* 45(2):210–222. <https://doi.org/10.1108/K-09-2014-0198>
- Duang Q, Huang DW, Lei L (2011) Simulation analysis of particle swarm optimization algorithm with extended memory. *Control Decis* 26(7):1087–1100
- Eberhart 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. <https://doi.org/10.1109/MHS.1995.494215>
- Fahmi A, Abdullah S, Amin F, Khan MSA (2018) Trapezoidal cubic fuzzy number Einstein hybrid weighted averaging operators and its application to decision making. *Soft Comput*. <https://doi.org/10.1007/s00500-018-3242-6>
- Huang Z, Chen Y (2013) An improved artificial fish swarm algorithm based on hybrid behavior selection. *Int J Control Autom* 6(5):103–116. <https://doi.org/10.14257/ijca.2013.6.5.10>
- Karaboga D (2005) An idea based on honey bee swarm for numerical optimization (Technical Report - TR06). Erciyes University, Engineering Faculty, Computer Engineering Department
- Karaboga D, Basturk B (2008) On the performance of artificial bee colony (ABC) algorithm. *Appl Soft Comput* 8(1):687–697. <https://doi.org/10.1016/j.asoc.2007.05.007>
- Karol S, Mangat V (2013) Evaluation of text document clustering approach based on particle swarm optimization. *Cent Eur J Comput Sci* 3(2):69–90. <https://doi.org/10.2478/s13537-013-0104-2>
- Li XL, Qian JX (2003) Studies on artificial fish swarm optimization algorithm based on decomposition and coordination techniques. *Chin J Circuits Syst* 8(1):1–6
- Li XL, Shao ZJ, Qian JX (2002) An optimizing method based on autonomous animate: fish-swarm algorithm. *Chin J Syst Eng Theory Pract* 22(11):32–38. [https://doi.org/10.12011/1000-6788\(2002\)11-32](https://doi.org/10.12011/1000-6788(2002)11-32)
- Mao M, Duan Q, Duan P, Hu B (2017) Comprehensive improvement of artificial fish swarm algorithm for global MPPT in PV system under partial shading conditions. SAGE. <https://doi.org/10.1177/0142331217697374>
- Reynolds RG, Peng B (2004) Cultural algorithms modeling of how cultures learn to solve problems. In: *Proceedings of the 16th IEEE international conference on tools with artificial intelligence*. IEEE, Boca Raton, FL, USA. <https://doi.org/10.1109/ICTAI.2004.45>
- Shakeel M, Abdullah S, Fahmi A (2018) Triangular cubic power aggregation operators and their application to multiple attribute group decision making. *Punjab Univ J Math* 50(3):75–98
- Sumathi S, Ashok Kumar L, Surekha P (2016) *Computational intelligence paradigms for optimization problems using MATLAB®/SIMULINK® (illustrate)*. CRC Press, Boca Raton
- Wang HB, Fan CC, Tu XY (2016) AFSAOCP: a novel artificial fish swarm optimization algorithm aided by ocean current power. *Appl Intell* 30:992–1007. <https://doi.org/10.1007/s10489-016-0798-7>
- Wu Y, Gao XZ, Zenger K (2011) Knowledge-based artificial fish-swarm algorithm. In: *IFAC proceedings volumes*. vol 44, pp 14705–14710. IFAC. <https://doi.org/10.3182/20110828-6-IT-1002.02813>
- Zhang C, Zhang FM, Li F, Wu HS (2014) Improved artificial fish swarm algorithm. In: *Proceedings of the 2014 9th IEEE conference on industrial electronics and applications, ICIEA 2014*. pp 748–753. <https://doi.org/10.1109/ICIEA.2014.6931262>
- Zhang H, Hong Q, Shi X, He J (2018) A social tagging recommendation model based on improved artificial fish swarm algorithm and tensor decomposition. In: *Security with intelligent computing and big-data services—SICBS 2017*. Springer, Cham, pp 3–13. https://doi.org/10.1007/978-3-319-76451-1_1
- Zhou GL, Li YM, He YC, Wang XL, Yu MC (2018) Artificial fish swarm based power allocation algorithm for MIMO-OFDM

relay underwater acoustic communication. *IET Commun* 12(9):1079–1085. <https://doi.org/10.1049/iet-com.2017.0149>

Zhu X, Ni Z, Cheng M, Jin F, Li J, Weckman G (2018) Selective ensemble based on extreme learning machine and improved discrete artificial fish swarm algorithm for haze forecast. *Appl*

Intell 48(7):1757–1775. <https://doi.org/10.1007/s10489-017-1027-8>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.