Improved artificial bee colony algorithm based on damping motion and artificial fish swarm algorithm

Liyi Zhang¹, Mingyue Fu², Hongbo Li², and Ting Liu¹*

¹School of Information Engineering, Tianjin University of Commerce, Tianjin 300134, China
²School of Economics, Tianjin University of Commerce, Tianjin 300134, China
*Email: liuting@tjcu.edu.cn

Abstract. The basic artificial bee colony (ABC) algorithm is easy to fall into local optimum, and it has poor exploitation ability and slow convergence speed. According to the defects, the improved algorithm is proposed, which is based on damping motion and artificial fish swarm (AFS) algorithm. In the employed bees phase, the swarm behavior of AFS algorithm which can avoid sinking into local optimum is introduced. At the same time, considering the slow convergence speed, employed bees adopt the multidimensional updating process in the early stage of iteration. In the onlooker bees phase, an adaptive step size based on damping motion is designed to replace the random step size, which can balance the global exploration and the local exploitation. Through the simulation of six test functions, we get the average convergence algebra, the mean, the best and the variance of the optimal solution through 30 experiments. Simulation shows that the improved algorithm has a better performance than the basic one.

1. Introduction
Artificial bee colony algorithm has some advantages like fewer control parameters, stronger robustness and better search performance, but it has the disadvantages of easy sinking into local optimum and slow convergence speed [1]. Therefore, many scholars have continuously improved it. Li proposed IABC algorithm that used memory algorithm to simulate a memory mechanism. Based on the original ABC, IABC added two schemes: simple dynamic change detection scheme and simple storage scheme. These two improved schemes can guide the artificial bees to further forage and reduce unnecessary repeated search [2]. Akay et al. [3] introduced the chaotic optimization algorithm and reverse learning method to the population initialization and the search mechanism which can improve the global search ability. Cao et al. [4] proposed the genetic algorithm in the stage of onlooker bees and improved the search equation in the stage of scout bees. The algorithm improves the exploration ability and convergence speed. Li et al. [5] introduced inertial components and acceleration coefficients in order to improve the search equation. The improved algorithm is called I-ABC, which can reduce sinking into local optimum and speed up global search. ACO-ABC algorithm was proposed by combining ACO and ABC and successfully applied in the classification model of medical data [6]. SABC algorithm based on the inspiration of hybrid frog leaping algorithm was proposed. SABC divided the average population into several groups. Newborn bees in each group were compared with the worst bees [7]. CGABC algorithm was inspired by cross-mutation in the difference evolution algorithm and the genetic algorithm. It could improve population diversity and global optimization.
ability [8]. The search process of firefly algorithm which can strengthen the global search ability was introduced into ABC algorithm, and FA-ABC was proposed. [9]. In order to get better candidate solutions and search ability, CABC algorithm was proposed by improving the search equation [10].

Based on the AFS algorithm and under-damped motion in physics, this paper proposes an improved ABC algorithm. Inspired by swarm behaviors in the AFS algorithm, in employed bees phase, the perceptual range of each employed bee is calculated and crowding degree is judged, which can avoid algorithm being caught in local optimum. Furthermore, multidimensional update strategy is adopted, which can accelerate convergence speed. For the onlooker bees, the adaptive search based on under-damped motion is used to search the neighborhood. It can ensure the strong global search in the early stage and a strong local search in the late stage.

2. Algorithm principle

2.1. Artificial bee colony algorithm

ABC algorithm has been extensively studied by scholars since it was proposed by Karaboga [11]. The bee population is divided into three categories: employed bee, onlooker bee and scout bee. Artificial Bee Colony Algorithm is to find the global optimal food source through the transformation of three bee identities. Artificial Bee Colony Algorithm is mainly divided into initialization stage, employed stage, onlooker stage and scout stage.

(1) Initialization: The maximum number of cycles $MaxCycle$ and the limit value of the number of times that the employed bee does not update the food source into the scout bee. The number of employed bees in the initialization stage is equal to the number of onlooker bees and the number of food sources $SN$. Randomly initialize food source $x_i = \{x_{id} | i = 1, 2, ..., SN; d = 1, 2, ..., D\}$.

$$x_{id} = x_{id, min} + (x_{id, max} - x_{id, min})rand$$

Where, $D$ is the dimension of the optimization, $x_{id, min}$ and $x_{id, max}$ are the minimum and maximum in the dimension of the food source $i$, $rand$ is a random number between $(0,1)$.

If minimization problem is considered, the concentration of food source is calculated according to (2), which is the fitness value to be solved.

$$f_{it}(x) = \begin{cases} 1 & f(x_i) \geq 0 \\ \frac{1}{1 + f(x_i)} & f(x_i) < 0 \end{cases}$$

Where $f_{it}(x)$ is the fitness of the food source $i$, and $f(x)$ is the objective function of the corresponding problem.

(2) Employed bees: The employed bee searches its neighborhood according to (3) and then chooses it according to greedy selection. If concentrations of new food sources searched by the neighborhood are higher than those of old ones, old ones are abandoned. Otherwise, the new one is abandoned. New food source is $v_i = \{v_{id} | i = 1, 2, ..., SN; d = 1, 2, ..., D\}$.

$$v_{id} = \begin{cases} x_{id} + (x_{id} - x_{id}) * r_{id} & \text{if } d = d_{rand} \\ x_{id} & \text{else} \end{cases}$$

Where $q \in \{1, 2, ..., SN\}$ is determined randomly and it is different from $i$. $d_{rand}$ is the random integer between $[1, D]$; $r_{id} \in [-1, 1]$ is a random step size, and it controls the scope of the search.

(3) Onlooker bees: After the employed bees, all employed bees in the dance area will transmit food information by dancing swing dance to the onlookers. After obtaining the information, the onlookers choose better food sources by roulette. Following the employed bees, the onlooker bee also conducts
neighborhood search according to (3) and evaluates them with the greedy selection. When the food concentrations of the onlookers are higher than these of the employed bees, the roles of the employed bees are replaced. Otherwise, stay the same.

(4) Scout bees: limit is the number of trials. If the continuous limit of a food source concentration is not updated, the employed bee will become scout bee and its solution will be abandoned.

2.2. Artificial Fish Swarm Algorithm
AFS algorithm is proposed by Li Xiaolei [12]. It simulates the prey behavior, swarm behavior, follow behavior and random behavior of artificial fish (AF). There are SN artificial fishes. The AF $i$ is expressed as a vector $\mathbf{X}_i = \{X_{i1}, X_{i2}, \ldots, X_{iD}\}, i \in \{1, 2, \ldots, SN\}$. Each AF state is a potential solution. Visual is the perceived field of AF, delta is crowding factor of the AF, Step is the maximum moving step of AF, $Y$ is objective function value of the AF.

(1) Prey behavior: The position of AF is $\mathbf{X}_i$. Randomly select another AF $\mathbf{X}_j = \{X_{j1}, X_{j2}, \ldots, X_{jD}\}, \ j \in \{1, 2, \ldots, SN\}$ in the field of Visual. Their objective function are calculated and compared. If $Y_j$ is better than $Y_i$ (If maximization problem is considered), $\mathbf{X}_i$ moves to $\mathbf{X}_j$; otherwise, $\mathbf{X}_i$ continues to select $\mathbf{X}_j$ randomly in its field of Visual. If $\mathbf{X}_i$ does not move after trynumber times, it will make one step forward randomly.

$$
\begin{aligned}
X_i &= X_j + \text{rand Step} \cdot \frac{X_j - X_i}{\|X_j - X_i\|} & Y_j > Y_i \\
X_i &= X_i + \text{rand Step} & \text{else}
\end{aligned}
$$

(4)

(2) Swarm behavior: The position of AF is $\mathbf{X}_i$. There are $n_f$ artificial fishes in the field of Visual. $\hat{Y}_c$ is the objective function of the central position $\mathbf{X}_c$. If $Y_c / n_f > \text{delta} * Y_i$, indicating that the food concentration at the center position $\mathbf{X}_c$ is better and it isn’t crowded, and $\mathbf{X}_i$ moves a step toward the center $\mathbf{X}_c$, otherwise the prey behavior is performed.

$$
\begin{aligned}
X_i &= X_i + \text{rand Step} \cdot \frac{X_c - X_i}{\|X_c - X_i\|} & Y_c / n_f > \text{delta} * Y_i \\
\text{prey behavior} & \text{else}
\end{aligned}
$$

(5)

(3) Follow behavior: The position of the AF is $\mathbf{X}_i$. $Y_{\text{max}}$ is maximum objective function value in the field of Visual and its corresponding location is $\mathbf{X}_{\text{max}}$. If $Y_{\text{max}} / n_f > \text{delta} * Y_i$, it shows that $\mathbf{X}_{\text{max}}$ has better food concentration and it isn’t crowded around. The $\mathbf{X}_i$ moves a step in the $\mathbf{X}_{\text{max}}$ direction, otherwise the prey process is performed.

$$
\begin{aligned}
X_i &= X_i + \text{rand Step} \cdot \frac{X_{\text{max}} - X_i}{\|X_{\text{max}} - X_i\|} & Y_{\text{max}} / n_f > \text{delta} * Y_i \\
\text{prey behavior} & \text{else}
\end{aligned}
$$

(6)

(4) Random behavior: The random behavior is a defect behavior. In the field of Visual, select a position randomly and make a step forward in that direction. The purpose of the random behavior is to increase the range of the global search.

3. Improved artificial bee colony algorithm

3.1. Improved ABC algorithm based on damping motion
Vibration can be divided into critical damped motion, under-damped motion and over-damped motion in physics. Under-damped motions have more practical significance. The vibration image of
under-damped motion is shown in Figure 1. From Figure 1, the amplitude will become smaller and smaller as time goes on, and finally reach a static state.

![Figure 1: Vibration process of under-damped motion.](image)

Taking under-damped spring oscillator as an example, the vibration law of under-damped motion without external influence is:

\[ X = A_0 * e^{-\xi t} \cos(\omega t + \phi) \]  \hspace{1cm} (7)

Where \( A_0 \) is the amplitude, \( \xi \) is the damping factor, \( \omega \) is the circular frequency of vibration, \( t \) is the time, \( \phi \) is the phase.

In the standard ABC algorithm, the onlooker bee chooses its neighborhood according to the random step size \( r_{id} \in [-1,1] \). The step size in the neighborhood search has a very obvious influence on global exploration and local exploitation. In order to enhance local search ability, an improved ABC algorithm based on adaptive step size and damping motions (DMABC) is proposed.

In the early stage of the iteration, we need to take full account of the global exploration and accelerate the convergence speed. Therefore, in the early iteration, it’s better to have a larger step size. In the later stage of the iteration, we need to take full account of local exploitation. So in the later iteration, it is better for the artificial bee to have a smaller step size.

In the DMABC algorithm, the adaptive step size is designed:

\[ \beta = e^{-\frac{\xi \text{Cycle}}{\text{MaxCycle}}} \cos(\text{Cycle}) \]  \hspace{1cm} (8)

Where, \( \xi \) is the damping factor, \( \text{Cycle} \) is the current iteration’s number, \( \text{MaxCycle} \) is the maximum cycle number. The larger \( \xi \), the faster vibration attenuation. The smaller \( \xi \), the slower amplitude attenuation. The value of \( \xi \) is determined by experiment.

The improved neighborhood search based on the adaptive step in onlooker bees phase is as follows:

\[ v_{id} = \begin{cases} 
  x_{id} + (x_{id} - x_{gd}) * r_{id} * e^{-\frac{\text{Cycle}}{\text{MaxCycle}}} \cos(\text{Cycle}) & \text{if } d = d_{rand} \\
  x_{id} & \text{else} 
\end{cases} \]  \hspace{1cm} (9)

In the DMABC algorithm, as the cycle’s number increases, step size can change from large to small. In the early iteration, the step size is large, and can maintain global search and accelerate convergence speed. In the later iteration, the step size reduces, which can carry out the local exploitation.

### 3.2. Improved ABC algorithm based on damping motion and AFS

ABC algorithm is easily caught in the local optimal solution, which can affect the overall optimization of the bee colony. Based on the DMABC and inspired by the AFS, an improved ABC algorithm (FDMABC) is proposed. The perception range and crowding factor of artificial bees are introduced to set the perception range and crowding factor of artificial bees. When employed bees search for neighborhoods, they must first judge whether they are crowded in their perception ranges.
Neighborhood searches can be carried out only when there is no congestion, which can avoid algorithm falling into local optimum. At the same time, considering slow speed of one-dimensional updating, employed bees adopt multidimensional updating processes in the early iteration. In the later iteration, the updating frequency is too fast, which may lead to elimination of optimal solution. Therefore, employed bee uses the one-dimensional updating process to increase accuracy.

FDMABC algorithm is as follows:

1. Set the artificial bee’s perceived distance $\text{Visual}$, crowding factor $\delta$, $\text{limit}$, $\text{MaxCycle}$ and $SN$. Food sources are initialized randomly.

2. Employed bees conduct neighborhood searches in the perceptual range, and judge whether the central location within the perceptual range is crowded. If not crowded, an improved neighborhood search is conducted in the perceptual range. If crowded, a traditional neighborhood search is conducted in the global range.

For the employed bee, the crowding formula for judging whether the central position of the perception range is crowded or not is as follows:

$$\frac{\text{fit}(c_x)}{n_f} > \delta * \text{fit}(c_i)$$

Where, $x_c = \{x_{c1}, x_{c2}, \ldots, x_{cd}, \ldots, x_{cD}\}$ is the central position of the employed bee $x_i = \{x_{i1}, x_{i2}, \ldots, x_{id}, \ldots, x_{iD}\}$ within the perceptual range, $\text{fit}(c_x)$ and $\text{fit}(c_i)$ are the fitness values of $x_c$ and $x_i$, $\delta$ and $n_f$ are the same with the AFS algorithm.

The multidimensional neighborhood search formula for the employed bee is as follows:

$$v_i = \begin{cases} x_c + \text{rand} * \text{Step} \cdot \frac{x_i - x_c}{\|x_c - x_i\|} & \text{if not crowded} \\ x_i + (\text{rand} - 0.5) \cdot 2 \cdot (x_i - x_q) & \text{else} \end{cases}$$

Where, $x_q = \{x_{q1}, x_{q2}, \ldots, x_{qd}, \ldots, x_{qD}\}$ is determined randomly in the global range and it has to be different from $x_i$, $\text{Step}$ is the same with the AFS algorithm.

The one-dimensional neighborhood search formula of the employed bee is

$$v_{id} = \begin{cases} x_{id} + \text{rand} \cdot \text{Step} \cdot \frac{x_{id} - x_{id}}{\|x_c - x_i\|} & \text{if } d = d_{\text{rand}} \text{ and not crowded} \\ x_{id} + (\text{rand} - 0.5) \cdot 2 \cdot (x_{id} - x_{qd}) & \text{if } d = d_{\text{rand}} \text{ and crowded} \\ x_{id} & \text{else} \end{cases}$$

3. The onlooker bee chooses to employed bee by roulette, and searches the neighborhood according to equation (9) to get the new food source. If the new food source’s concentration is higher, the original food source should be abandoned and the new food source should be selected, otherwise, it will remain unchanged.

4. If the continuous $\text{limit}$ is not updated, the corresponding food source should be abandoned, at this time, an employed bee convert into a scout bee, and algorithm enters the stage of scout bee.

5. The algorithm satisfies the end condition and outputs the best solution.

4. Computer simulation

Six benchmark functions (as shown in Table 1) are selected for simulation and comparison on Matlab 2012b. ABC, DMABC, and FDMABC are compared. Initial parameters of the three algorithms are the same. The initial food sources $SN$ is 50, and it is also the number of employed bees and onlooker bees.
Maximum cycles is $\text{MaxCycle}. \text{limit}$ is defined as $\frac{1}{20}\text{MaxCycle}$. Visual range of bees is 2.5, maximum step size $\text{Step}$ is 0.3, and crowding factor $\text{delta}$ is 0.618, damping factor $\xi$ is 10.

### Table 1. Benchmark Function.

| Function | Functional expression | Range                  | $f_{\text{min}}$ |
|----------|------------------------|------------------------|------------------|
| $f_1$    | $f_1(x) = \sum_{i=1}^{D} x_i^2$ | $[-100,100]^D$          | 0                |
| $f_2$    | $f_2(x) = \sum_{i=1}^{D} \left[ x_i^2 - 10\cos(2\pi x_i) + 10 \right]$ | $[-5.12,5.12]^D$       | 0                |
| $f_3$    | $f_3(x) = \sum_{i=1}^{D} \left[ 100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2 \right]$ | $[-30,30]^D$           | 0                |
| $f_4$    | $f_4(x) = \frac{1}{4000} \sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1$ | $[-600,600]^D$         | 0                |
| $f_5$    | $f_5(x) = \sum_{i=1}^{D} \left| x_i \right|^{i+1}$ | $[-100,100]^D$         | 0                |
| $f_6$    | $f_6(x) = \sum_{i=1}^{D} i x_i^2$ | $[-10,10]^D$           | 0                |

In the experiments, the experimental dimensions are 3 and 10 respectively. Set $\text{MaxCycle}$ are 200 and 300 respectively when the optimization dimensions are 3 and 10. Each experiment was repeated 30 times. Table 2-7 give the results of the experimental operation. $\text{Mean}$, $\text{Best}$ and $\text{Std}$ are the mean, the best and the variance of the global optimal solution produced by multiple independent experiments. $\text{Mean – iter}$ is the average convergence algebra of multiple independent experiments. In the experiments, $\text{Mean}$ and $\text{Best}$ can reflect the optimization ability and convergence accuracy. $\text{Mean – iter}$ reflects the convergence speed. $\text{Std}$ reflects the stability. Table 2-7 show that DMABC algorithm and FDMABC algorithm are more effective in dealing with low-dimensional dimensions. FDMABC algorithm has better convergence accuracy and convergence speed than standard ABC algorithm and DMABC algorithm.

### Table 2. Convergence results of $f_1$ benchmark function.

| Function | Dimension | Mean | Best | Std | Mean-iter |
|----------|-----------|------|------|-----|-----------|
| $f_1$    | ABC       | 3    | 2.61E-5 | 1.15E-5 | 2.11E-5 | 162.75 |
|          |           | 10   | 1.22E-3 | 1.20E-3 | 8.76E-4 | 199.93 |
|          | DMABC     | 3    | 9.97E-6 | 3.14E-6 | 9.49E-6 | 64.13  |
|          |           | 10   | 4.15E-4 | 3.53E-4 | 3.97E-4 | 151.75 |
|          | FDMABC    | 3    | 3.80E-6 | 1.52E-7 | 6.20E-6 | 47.14  |
|          |           | 10   | 8.92E-5 | 5.17E-5 | 6.84E-5 | 120.08 |

### Table 3. Convergence results of $f_2$ benchmark function.

| Function | Dimension | Mean | Best | Std | Mean-iter |
|----------|-----------|------|------|-----|-----------|
| $f_2$    | ABC       | 3    | 4.25E-1 | 3.60E-1 | 9.49E-2 | 120.71 |
|          |           | 10   | 1.43  | 8.96E-1 | 4.67E-1 | 198.41 |
|          | DMABC     | 3    | 2.66E-1 | 1.14E-1 | 7.62E-2 | 117.57 |
| Dimension | Mean  | Best   | Std    | Mean-iter |
|-----------|-------|--------|--------|-----------|
| 3         | 1.67  | 4.85E-1| 4.15E-1| 184.35    |
| 10        | 5.32E-2| 1.89E-2| 3.97E-2| 78.32     |
| 10        | 1.11  | 2.90E-1| 7.24E-1| 165.14    |

Table 4. Convergence results of $f_3$ benchmark function.

| Dimension | Mean  | Best   | Std    | Mean-iter |
|-----------|-------|--------|--------|-----------|
| 3         | 6.26E-02| 4.73E-02| 5.76E-02| 162.41    |
| 10        | 11.12 | 8.232  | 4.01   | 210.78    |
| 10        | 1.42E-02| 3.69E-03| 8.85E-03| 119.25    |
| 10        | 5.78  | 1.25   | 4.97   | 189.15    |
| 10        | 1.12E-02| 4.41E-03| 9.17E-03| 80.65     |

Table 5. Convergence results of $f_4$ benchmark function.

| Dimension | Mean  | Best   | Std    | Mean-iter |
|-----------|-------|--------|--------|-----------|
| 3         | 2.75E-2| 1.92E-2| 8.72E-2| 103.64    |
| 10        | 6.91E-1| 9.54E-2| 3.72E-2| 240.75    |
| 10        | 1.78E-2| 9.52E-3| 1.69E-3| 74.48     |
| 10        | 2.32E-1| 2.43E-2| 6.16E-2| 210.09    |
| 10        | 1.62E-2| 2.11E-3| 6.09E-3| 68.65     |
| 10        | 4.78E-2| 7.60E-3| 3.98E-2| 200.98    |

Table 6. Convergence results of $f_5$ benchmark function.

| Dimension | Mean  | Best   | Std    | Mean-iter |
|-----------|-------|--------|--------|-----------|
| 3         | 1.78E-6| 4.65E-7| 2.65E-6| 107.21    |
| 10        | 1.25E-2| 8.65E-3| 1.37E-2| 148.24    |
| 10        | 7.23E-7| 2.48E-7| 3.20E-7| 95.05     |
| 10        | 1.62E-3| 2.25E-3| 1.07E-3| 104.61    |
| 10        | 1.07E-7| 3.76E-8| 2.43E-7| 92.12     |
| 10        | 2.67E-4| 3.54E-5| 1.03E-4| 99.45     |

Table 7. Convergence results of $f_6$ benchmark function.
Figure 2-7 shows the average convergence processes of the six benchmark functions when the experimental dimension is 3. In Figure 2-7, the convergence speed of FDMABC is quickened, compared with the standard ABC and DMABC. It is mainly due to the multidimensional neighborhood search. At the same time, FDMABC algorithm has better convergence accuracy because it has better local exploitation.
5. Conclusion
The FDMABC proposed in this paper is based on the AFS and the damping motion. It reduces the probability of local optimum, increases local search performance, and further improves the convergence speed. The experiments on six test functions in different dimensions prove the improvement. The FDMABC algorithm has been further improved than DMABC and ABC.

ABC algorithm has been studied and developed for more than ten years since it was put forward in 2005. It has been proved that ABC algorithm plays an important role in solving optimization problems and practical applications. But at the same time, the algorithm itself has defects. In the future, we should further expand the field of application, especially in the optimization of discrete and dynamic uncertain environments.

References
[1] Karaboga, D., Basturk, B. (2008) On the performance of artificial bee colony (ABC) algorithm. Applied Soft Computing, 8(1): 687-697.
[2] Li, X.N., Yang, G.F. (2016) Artificial bee colony algorithm with memory. Applied Soft Computing, 41: 362-372.
[3] Akay, B., Karaboga, D. (2012) A modified artificial bee colony algorithm for real-parameter optimization. Information Sciences, 192(1): 120-142.
[4] Cao, Y.C., Lu, Y., Pan, X.Q., Sun, N. (2018) An improved global best guided artificial bee colony algorithm for continuous optimization problems. Cluster Computing, 22(36): 1-9.
[5] Li, G.Q., Niu, P.F., Xiao, X.J. (2012) Development and investigation of efficient artificial bee colony algorithm for numerical function optimization. Applied Soft Computing, 12(1): 320-332.
[6] Almuhaideb, S., Menai, M.E.B. (2014) A new hybrid metaheuristic for medical data classification. International Journal of Metaheuristics, 3(1): 59-80.
[7] Zhao, H.X., Chang, X.G. (2018) Improvement of artificial bee colony algorithm. Computer Engineering and Design, 39(1): 260-265.
[8] Zhang, P.H., Li, J.M., Hu, X.D., Hu, J. (2017) Research on global artificial bee colony algorithm based on crossover. Journal of Shandong University of Technology (Social Sciences Edition), 31(5): 6-17.
[9] Tuba, M., Bacanin, N. (2014) Artificial bee colony algorithm hybridized with firefly algorithm for cardinality constrained mean-variance portfolio selection problem. Applied Mathematics & Information Sciences, 8(6): 2831-2844.
[10] Gao, W.F., Liu, S.Y. (2012) A novel artificial bee colony algorithm based on modified search equation and orthogonal learning. Computers and Operations Research, 39(3): 687-697.
[11] Karaboga, D. (2005) An idea based on honey bee swarm for numerical optimization. Technical Report, Kayseri: Erciyes university, Computer Engineering Department.
[12] Li, X.L., Shao, Z.J., Qian, J.X. (2002) An optimizing method based on autonomous animals: fish-swarm algorithm. Systems Engineering-theory & Practice, 22(11): 32-38.