IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 14, 2019, accepted November 5, 2019, date of publication November 14, 2019,
date of current version November 25, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2953544

Advanced Fruit Fly Optimization Algorithm and Its Application to Irregular Subarray Phased Array Antenna Synthesis

WENTAO LI, (Member, IEEE), YUDONG ZHANG, AND XIAOWEI SHI, (Senior Member, IEEE)

Science and Technology on Antenna and Microwave Laboratory, Department of Electronic Engineering, Xidian University, Xi'an 710071, China

Corresponding author: Wentao Li (wtli@mail.xidian.edu.cn)

This work was supported in part by the National Nature Science Foundation of China under Grant 61571356, in part by the Shaanxi Natural Science Foundation under Grant 2018JM6049, in part by the Fundamental Research Funds for Central Universities under Grant JB180203, in part by the Ningbo Natural Science Foundation, and in part by the Foundation of Science and Technology on Electromechanical Dynamic Control Laboratory, China.

ABSTRACT In this paper, an advanced fruit fly algorithm (FOA) is proposed and applied in subarray phased array antenna synthesis. The proposed algorithm introduces orthogonal crossover, quantum selection and simulated annealing operations on the individuals, and then combines them by using an adaptive expansion-contraction factor. Accordingly, a linear generation mechanism of candidate solution based fruit fly algorithm (LGMS-FOA) is generated, in which individuals are selected in a highly balanced way, and the poor solutions are still accepted with a varying probability during the iteration. These mechanisms help the proposed algorithm enhance the population diversity and global searching capability but avoid falling into local optimum. Numerical classical unimodel benchmark functions are provided to test the proposed algorithm (OLFOA) in comparison with other advanced algorithms. In addition, to further validate its superiority, the proposed algorithm is applied to handle the subarray array synthesis of several tough planar and circular apertures with different array sizes and subarray shapes. Simulation results show that the proposed OLFOA can achieve better performance than other improved evolutionary algorithms.

INDEX TERMS Irregular subarray, fruit fly algorithm, orthogonal crossing, quantum behavior, simulated annealing, array synthesis.

I. INTRODUCTION

Recently, subarray technology has attracted considerable interests in radar and communication systems, because it can greatly reduce the cost and complexity of the system by reducing the number of antenna splitters and phase shifters [1]. Currently, the subarray partition and optimization problem remains to be a large barrier in the development of subarray technology. Various solutions have been explored to solve this problem. The subarray partitioning method based on genetic algorithm (GA) was proposed in [2]–[5]. These methods could solve the partitioning of irregular subarray, but got poor performance in solving the exact partition problem. Then, a method based on analytic schemata-driven optimization, which achieves exact partition in low-/medium-sized rectangular subarrays, was introduced in [6]. The X algorithm-based subarray tiling method, which exhibits great efficiency and accuracy in irregular subarray partitioning, was proposed in [7]. Although subarray technology applied in array antenna design helps to alleviate the high grating lobe to certain extent, exploring high-performance intelligent optimization algorithms for array synthesis, especially for large array or huge elements, is urgently desired.

Evolutionary algorithms have long been considered as efficient candidates to solve antenna synthesis problems. Genetic algorithm (GA) and particle swarm optimization (PSO) are two well-known optimization algorithms for array antenna synthesis [8], [9]. Simulated annealing algorithm (SA), ant colony optimization algorithm (ACO), and differential evolution algorithm (DE) are also widely used in array antenna optimization [10]–[12]. In addition, several recently proposed algorithms, such as invasive weed optimization

The associate editor coordinating the review of this manuscript and approving it for publication was Yanhui Liu.

algorithm (IWO) [13] and artificial bee colony algorithm technique (ABC) [14], have been successfully implemented in antenna design and optimization. These algorithms appear to be less and less effective in solving complex problems with high dimension and large computation overhead.

In 2012, motivated by the behavior of fruit fly in seeking food, a new fruit fly optimization algorithm (FOA) [15] was proposed. As a new global evolutionary algorithm, FOA is well-known for its simple principle, few adjustable parameters, strong global optimization capability, and fast convergence speed. Similar to other evolutionary algorithms, FOA also shows the disadvantages of easily falling into the local optimum and limited performance when calculating multi-dimensional problems. To address these shortcomings, a modified fruit fly algorithm integrated with the linear generation mechanism of candidate solution (LGMS) was proposed in [16], showing that the algorithm performance was greatly improved without changing any natural concept. However, the shortcomings of the fruit fly algorithm were still not addressed completely.

In this study, an advanced LGMS-FOA, namely, OLFOA, is proposed. By introducing orthogonal cross [17] and quantum selection [19] operations on the individuals, the proposed OLFOA combines them with an adaptive expansion-contraction factor to improve the convergence performance in solving high-dimensional complex problems. On this basis, the simulated annealing [20] strategy is introduced to further enhance the diversity of the population. The superior performance of the proposed algorithm has been validated via both numeral benchmark function tests and high-dimensional subarray phased array antenna synthesis problems with different array sizes and subarray shapes.

This paper is organized as follows: Section II introduces the concepts of FOA and LGMS-FOA. Section III presents a detailed architecture of the proposed OLFOA. Section IV makes comparisons between OLFOA and several recently proposed algorithms. Section V describes the application of OLFOA in solving high-dimensional subarray phased array antenna synthesis problems. Section VI draws the conclusion.

II. FOA AND LGMS-FOA

A. FOA

Fruit fly algorithm is a global optimization method based on the food-finding behavior of the fruit fly population. In the process of searching for food in the search space, other fruit flies are randomly searched for a certain range from the current location of fruit fly closest to the food. The search process is repeated starting from the fruit fly that is currently closest to the food until the food is found. Fig. 1 illustrates the food-finding iterative process of the fruit fly swarm. The *x*- and *y-axis* are the search spaces for the fruit fly population. The circles indicate the fruit fly swarms. The solid arrows indicate the food-finding direction of the fruit fly population, whereas the dashed arrows indicate the food-finding direction of fruit fly individuals. x^i_{best}, y^i_{best} indicates the fruit fly that

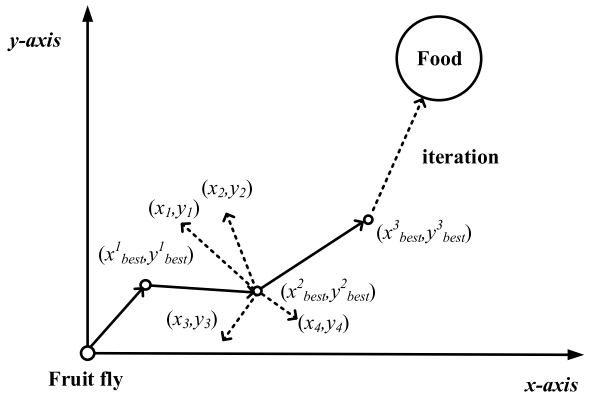


FIGURE 1. Food-finding iterative process of fruit fly swarm.

is closest to the food in the *i*th iteration. x_n, y_n indicates the location of *n*th fruit fly individual in the *x*- and *y-axis*, respectively.

The specific steps of the FOA algorithm are shown as follows:

Step 1. The maximum number of iterations (*gen*max), population size (*popsize*), range of random fruit flies (*LR*), and random direction of fruit fly (*FR*) in the fruit fly algorithm are initialized. Then, the location of the fruit fly population is defined.

$$x\_axis = rand(LR) \tag{1}$$
$$y\_axis = rand(LR) \tag{2}$$

where the *x_axis* and *y_axis* represent the searching center of the fruit fly population in *x*- and *y-axis*, respectively.

Step 2. Each fruit fly is updated with a random search direction and distance by

$$x_i = x\_axis + rand(FR) \tag{3}$$
$$y_i = y\_axis + rand(FR) \tag{4}$$

Step 3. The distance (Dist) between the location of the fruit fly and the origin and the taste decision value (*S*) of the fruit fly (*S* characterizes the distance from the food) are calculated as follows:

$$\text{Dist}_i = \sqrt{x_i^2 + y_i^2} \tag{5}$$
$$S_i = \frac{1}{Dist_i} \tag{6}$$

Step 4. The taste decision value is substituted into the fitness function to calculate the corresponding fitness value (Smell).

$$\text{Smell}_i = \text{fitness function}(S_i) \tag{7}$$

Step 5. The optimal fitness value (*bestsmell*) and corresponding location (*bestindex*) of the fruit fly individual are obtained by

$$[bestsmell, bestindex] = \min(\text{Smell}) \tag{8}$$

Step 6. The optimal fitness is set as the global optimal value (*gsmell*), and the corresponding location of fruit fly is used

as the searching center coordinate of the next iteration.

$$gsmell = bestsmell \tag{9}$$
$$x\_axis = x(bestindex) \tag{10}$$
$$y\_axis = y(bestindex) \tag{11}$$

Step 7. ***Steps 2–6*** are repeated until the terminal criterion is met or the maximum number of iterations is reached.

B. LGMS-FOA

From the description of the fruit fly algorithm, it is discovered that, given that the taste determination value (S) is always greater than zero, the fruit fly algorithm cannot express the negative number in the dimension level. Moreover, given that the taste judgment value cannot satisfy the uniform distribution, the searching of the fruit fly algorithm in the dimension is not uniform; thus, the fruit fly algorithm cannot effectively solve the complex optimization problem [16].

With this inspiration, the LGMS-FOA proposed by D. Shan in 2014 [16] expands the range of taste judgment values by introducing the LGMS mechanism. By doing so, the new taste judgment value satisfies the uniform distribution. The process of LGMS-FOA is shown as follows:

Step 1. The maximum number of iterations (genmax), population size ($popsize$), searching coefficient (n), initial weight (w_0), and weight coefficient (α) are initialized. Then, the location of the fruit fly population is defined.

$$x\_axis = n \times rand(\text{domain}) \tag{12}$$

where x_axis represents the searching center of the fruit fly population

Step 2. Each fruit fly is updated with a random search direction and distance by

$$x_i = x\_axis + w \times rand(\text{domian}) \tag{13}$$
$$w = w_0 \times \alpha^{gen} \tag{14}$$

where gen represents the current number of iterations.

Step 3. The taste decision value (S) of individual flies is calculated on the basis of LGMS.

$$S_i = x_i \tag{15}$$

Step 4. The taste decision value is substituted into the fitness function to calculate the corresponding fitness value (Smell).

$$\text{Smell}_i = \text{fitness function}(S_i) \tag{16}$$

Step 5. The optimal fitness value ($bestsmell$) and corresponding location ($bestindex$) of the fruit fly individual are obtained by

$$[bestsmell, bestindex] = \min(\text{Smell}) \tag{17}$$

Step 6. The optimal fitness is set as the global optimal value ($gsmell$), and the corresponding location of the fruit fly is used as the searching center of the next iteration.

$$gsmell = bestsmell \tag{18}$$
$$x\_axis = x(bestindex) \tag{19}$$

Step 7. *Steps 2–6* are repeated until the terminal criterion is met or the maximum number of iterations is reached.

III. OLFOA

For the proposed OLFOA, on the basis of the LGMS, the orthogonal cross and quantum selection mechanism (OQSM) and simulated annealing strategies (SM) are employed to achieve good performance and highly stable convergence. The two mechanisms are presented below in details.

A. OQSM

1) QUANTUM BEHAVIOR MECHANISM

Quantum behavior mechanism [17] has been already used in quantum particle swarm optimization (QPSO) to improve the performance of the algorithm. In the original PSO, the velocity of the particles and the search space of each iteration are limited, which means that the searching area cannot cover the entire feasible searching space. Therefore, the PSO algorithm cannot guarantee the absolute convergence to the global optimal solution [18]. In the quantum space, particles with quantum motion characteristics can appear at any point in space with a certain probability and can reach the entire feasible solution space. As a result, the capability of the PSO algorithm to obtain the global optimum is greatly improved.

On the basis of the principle of quantum behavior, the locations of the particles are updated by

$$X_{r,j}(gen+1) = gbest_{r,j} \pm \frac{W(gen)}{2} \ln(\frac{1}{u(gen)}) u \sim u(0,1) \tag{20}$$

where X means the location of particle, $gbest$ means the best location of the particle in the last iteration, gen means the number of current iteration, r and j represent the number of individuals and the dimension of the locations, respectively. W means the probability that a particle will appear at a relative point and is expressed by the following formula:

$$W(gen) = 2\beta \cdot \left| mbest(gen) - X_{r,j}(gen) \right| \tag{21}$$

where $mbest$ means the mean value of the best position and is expressed by

$$mbest(gen) = \frac{1}{popsize} \sum_{r=1}^{popsize} gbest_r(gen) \tag{22}$$

where $popsize$ represents the population size. β in (21) means expansion-contraction factor and is expressed by

$$\beta = 0.5 \times (gen\max - gen)/gen\max + 0.5 \tag{23}$$

where genmax represents the maximum number of iterations.

The expansion-contraction factor is vital in the QPSO because it controls the convergence speed and search range of the particle. In QPSO, the expansion–contraction factor scales linearly and lacks flexibility. In this study, a weighting factor wf with values between 0 and 4 is introduced in

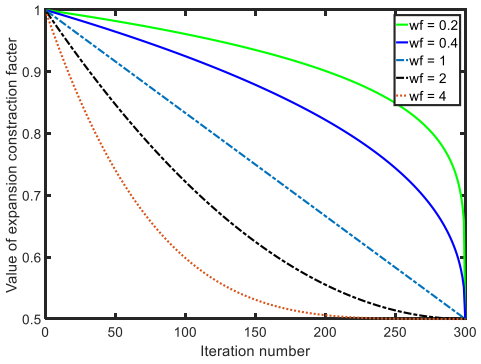


FIGURE 2. Expansion contraction factor versus iteration for different values of *wf*.

OLFOA. The adaptive expansion–contraction factor value is determined as follows, which replaces (23) in OLFOA.

$$\beta = 0.5 \times ((gen\max - gen)/gen\max)^{wf} + 0.5 \quad (24)$$

Fig. 2 shows the adaptive expansion-contraction factor versus iterations for different values of *wf*. The corresponding expansion–contraction factors have different convergence trends due to the different values of *wf*. A small *wf* value results in a slow convergence speed and a wide searching range. On the contrary, a large *wf* value leads to a fast convergence speed.

2) ORTHOGONAL DESIGN MECHANISM

As a multi-factor optimized experimental design method, the orthogonal design [19] introduces a series of orthogonal arrays for different numbers of factors and levels. The level represents the variable that affects the experimental results, whereas the factor represents the value of these variables. Different level and factor combinations will generate different samples and obtain different experimental results. The orthogonal crossover mechanism shows particular efficiency in handling the optimization problem, which doesn't require an exhaustive search over all the possible combinations to find the optimal solution.

By using orthogonal arrays, the levels and factors are rearranged in a certain order in accordance with the orthogonality, and only the most representative combinations are selected instead of calculating all possible combinations to find a suboptimal solution, thereby greatly reducing the computational burden. Specifically, the process of orthogonal crossover mechanism is established as follows:

I. Orthogonal array $L_M(Q^N)$ is selected for the orthogonal crossing, where L denotes a Latin square, Q and N denote the numbers of levels and factors, which are used for orthogonal crossing, respectively. M represents the number of combinations that is obtained by orthogonal crossing. Generally, a large number of M and Q will obtain good experiment results with an increase in the complexity. After numeral experiments, the $L_9(3^4)$ has been proven to be an efficient orthogonal array by considering performance and calculation complexity [19]. Table 1 shows the orthogonal array $L_9(3^4)$.

TABLE 1. Orthogonal array $L_9(3^4)$.

Combination	Factor1	Factor2	Factor3	Factor4
1	1	1	1	1
2	1	2	2	2
3	1	3	3	3
4	2	1	2	3
5	2	2	3	1
6	2	3	1	2
7	3	1	3	2
8	3	2	1	3
9	3	3	2	1

In Table 1, each row represents a combination of different factors and levels. For example, the sixth combination consists of the Factor 1 of Level 2, Factor 2 of Level 3, Factor 3 of Level 1, and Factor 4 of Level 2. The combinations are the most representative candidates for all possible combinations.

The orthogonality of the orthogonal array is referred to the following: (1) for the factor in any column, each level occurs for the same number of times; (2) for the two factors in any two columns, each combination of two levels occurs for the same number of times; and (3) the selected combinations are uniformly distributed over the entire space of all possible combinations [19].

II. The three individuals belonging to the same experiment are divided into four factors:

$$E_1 = (e_{1,1}, e_{1,2}, e_{1,3}, e_{1,4}) \quad (25)$$
$$E_2 = (e_{2,1}, e_{2,2}, e_{2,3}, e_{2,4}) \quad (26)$$
$$E_3 = (e_{3,1}, e_{3,2}, e_{3,3}, e_{3,4}) \quad (27)$$

where E means the level of experiments and e means the factor of experiments.

III. Three individuals are taken into the orthogonal array $L_9(3^4)$ and nine combinations are obtained.

$$O_d = (e_{m_d,1}, e_{m_d,2}, e_{m_d,3}, e_{m_d,4}) \quad 1 \le d \le 9 \quad (28)$$

where O_d means the dth combination and m_d means the parameter determined by the orthogonal array.

B. SM

The simulated annealing algorithm [20] is originally designed to search the metal state with the lowest energy during metal temperature dropping, i.e., metal annealing process. To avoid the situation that the optimal state cannot be reached due to rapid cooling, the simulated annealing algorithm accepts the solution, which is worse than the current with a certain probability.

In the proposed OLFOA, the similar mechanism named simulated annealing mechanism (SM) is introduced. Specifically, if the current solution is better than the previous one, the SM does not work; otherwise, the current solution will be

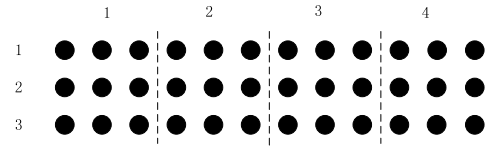


FIGURE 3. Diagram of division process.

still taken as the global optimum with a certain probability determined by SM.

C. OLFOA STEPS

With the improved mechanisms, the process of the OLFOA is described as follows:

Step 1. Initialization

The population size (*popsize*), searching coefficient (n), annealing temperature (T), final temperature (*Tfin*), maximum iteration number (*gen* max), and weighting factor (*wf*) are initialized. Then, the locations of the fruit fly swarm are initialized and updated as follows.

$$x = n \times rand(\text{domain}) \tag{29}$$

Step 2. Evaluation of the swarm

In this step, the S parameters of the fruit fly individual, which are calculated by LGMS, will be substituted into the objective function to obtain the smell values of the swarm. The optimal smell value is selected as the global optimal value (*gsmell*), and the corresponding fruit fly location is denoted as *gbest*.

$$S_i = x_i \tag{30}$$

$$\text{Smell}_i = \text{fitness function}(S_i) \tag{31}$$

$$[gsmell, gbest] = \min(\text{Smell}) \tag{32}$$

Step 3. OQSM update of the swarm

The locations for each individual of the fruit fly population are calculated three times on the basis of (20), and the three locations are set as levels of orthogonal array. Each level is divided into four parts and set as four factors. To describe the division process visually, its diagram is presented as follows:

The S parameters of the individual flies are assumed to have 12 dimensions. Each black circle represents the S-parameter of each dimension of the fruit fly individuals. Each row represents a fruit fly individual, the vertical numbers represent the levels, and the horizontal numbers represent the factors respectively. As shown in Fig. 3, the taste decision value (S) of the three fruit fly individuals are divided into four equal parts (as equal as possible). Each of the divided parts is brought into Table 1 on the basis of its number of levels and factors, and nine new individuals are reconstituted after the crossover.

Then, the smell value of all nine combinations of fruit fly individuals can be obtained on the basis of the fitness function (31). The fruit fly individual corresponding to the best value of smell among the nine individuals will be selected as the new fruit fly individual. Then, this process is repeated until the entire fruit fly swarm are updated.

TABLE 2. Comparison algorithms.

Algorithm	Method	Authors and reference
FOA	Fruit fly optimization algorithm	Pan et al. [15]
IFFO	Improved fruit fly optimization algorithm	Pan et al. [22]
CMFOA	Chaotic fruit fly optimization algorithm	L. Wu et al. [23]
MSFOA	Multi-scale cooperative mutation fruit fly optimization algorithm	Y. Zhang et al. [24]
CEFOA	Co-evolution fruit fly optimization algorithm	X. Han et al. [25]
MFOA	Mixed modified fruit fly optimization algorithm	Pan et al. [26]
RMPSO	Repository and mutation-based particle swarm optimization	B. Jana et al.[27]
BLPSO	Biogeography-based learning particle swarm optimization algorithm	X. Chen et al. [28]
JADE	Adaptive differential evolution algorithm	J. Zhang et al. [29]
NGHS	Novel global harmony search algorithm	D. Zou et al. [30]

Step 4. SM update of global optimum

Calculate the smell value of the updated fruit fly swarm. The best smell value (*bestsmell*) and corresponding location of the specific fruit fly (*bestindex*) is found:

$$[bestsmell, bestindex] = \min(\text{Smell}) \tag{33}$$

The SM triggering conditions are defined as follows:

$$\exp(-(bestsmell - gsmell)/T) > rand(0, 1) \tag{34}$$

If the current *bestsmell* is better than the *gsmell*, or the *gsmell* and *bestsmell* meet the SM triggering conditions in (34), set *bestsmell* as the global optimal value (*gsmell*), and the corresponding location of specific fruit fly (*bestindex*) will be set as *gbest*.

Then, T is updated by

$$T = T - (T - Tfin)/(gen\max - gen) \tag{35}$$

Step 5. Termination

Steps 3–4 is repeated until the maximum number of iterations is reached or the optimal value is found.

IV. EXPERIMENTS

In this section, OLFOA will be evaluated via the classical benchmark functions. The test results are compared with those of other improved algorithms to show the superior performance of the proposed OLFOA algorithm. Table 2 lists the improved algorithms in the experiments.

A. EXPERIMENTAL RESULTS

In this subsection, the capability of OLFOA is compared with the algorithms via 16 classic benchmark functions presented in Table 3. These algorithms are implemented by MATLAB 2016a and run on an i5 3.3 GHz CPU with 8 GB memory

TABLE 3. Benchmark functions.

ID	Function name	Equation	Global optimum	Domain
F1	Exponential problem	$f(x)=-\exp(-0.5\sum_{i=2}^{n}x_i^2)$	$x^*=0$ and $f(x^*)=-1$	$-1\le x_i\le 1$
F2	Quatic	$f(x)=\sum_{i=1}^{n}ix_i^4+rand()$	$x^*=0$ and $f(x^*)=0$	$-5.12\le x_i\le 5.12$
F3	Schwefels	$f(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i}x_j)^2$	$x^*=0$ and $f(x^*)=0$	$-100\le x_i\le 100$
F4	Schwefels 2.21	$f(x)=\max\lvert x_i\rvert, 1\le i\le n$	$x^*=0$ and $f(x^*)=0$	$-100\le x_i\le 100$
F5	Schwefels 2.22	$f(x)=\sum_{i=1}^{n}\lvert x_i\rvert+\prod_{i=1}^{n}\lvert x_i\rvert$	$x^*=0$ and $f(x^*)=0$	$-100\le x_i\le 100$
F6	Sphere	$f(x)=\sum_{i=1}^{n}x_i^2$	$x^*=0$ and $f(x^*)=0$	$-100\le x_i\le 100$
F7	Axis	$f(x)=\sum_{i=1}^{n}ix_i^2$	$x^*=0$ and $f(x^*)=0$	$-5.12\le x_i\le 5.12$
F8	Shifted Sphere	$f(x)=\sum_{i=1}^{n}z_i^2+f\_bias$	$z=x-o, o=(o_1,o_2,...,o_n), x^*=0$ $f(x^*)=f\_bias$	$-100\le x_i\le 100$
F9	Shifted Schwefels 1.2	$f(x)=\sum_{i=1}^{n}(\sum_{j=1}^{i}z_j)^2+f\_bias$	$z=x-o, o=(o_1,o_2,...,o_n), x^*=0$ $f(x^*)=f\_bias$	$-100\le x_i\le 100$
F10	Ackley	$f(x)=-20\exp(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2})$ $-\exp(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i))+20+e$	$x^*=0$ and $f(x^*)=0$	$-32\le x_i\le 32$
F11	Alpine	$f(x)=\sum_{i=1}^{n}\lvert x_i\sin x_i+0.1x_i\rvert$	$x^*=0$ and $f(x^*)=0$	$-100\le x_i\le 100$
F12	Griwank	$f(x)=\frac{1}{4000}\sum_{i=1}^{n}x_i^2-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	$x^*=0$ and $f(x^*)=0$	$-600\le x_i\le 600$
F13	Cosine Wave	$f(x)=-\sum_{i=1}^{n-1}(\exp(-\frac{x_i^2+x_{i+1}^2+0.5x_ix_{i+1}}{8}))$ $\times\cos(4\sqrt{x_i^2+x_{i+1}^2+0.5x_ix_{i+1}})$	$x^*=0$ and $f(x^*)=1\text{-}n$	$-5\le x_i\le 5$
F14	Neumaier 3 problem	$f(x)=\sum_{i=1}^{n}(x_i-1)^2-\sum_{i=2}^{n}x_ix_{i-1}$	$x^*=i(n+1-i)$ and $f(x^*)=-\frac{n(n+4)(n-1)}{6}$	$-n^2\le x_i\le n^2$
F15	Rastrign	$f(x)=\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i)+10)$	$x^*=0$ and $f(x^*)=0$	$-5.12\le x_i\le 5.12$
F16	Salomon	$f(x)=1-\cos(2\pi\sqrt{\sum_{i=1}^{n}x_i^2})+0.1\sqrt{\sum_{i=1}^{n}x_i^2}$	$x^*=0$ and $f(x^*)=0$	$-100\le x_i\le 100$

capacity. These algorithms are applied to minimize 16 benchmark functions with dimensions Dim $=$ 30, 50. The population size is set as 80. In order to make a fair comparison, the maximum generation number is set to 300 for each function, which is consistent with the references. The experiment results are obtained by independently running 10 times for fairness. The control parameters of all improved algorithms are set as the same as in [15] and [22]–[30]. In OLFOA, T is set as 900, *Tfin* is set as 0.001×10^{-10}, and the weighting factor *wf* is set as 0.4.

Table 4 shows the mean values and standard deviations of different fruit fly algorithms evaluated via numerical

TABLE 4. Results of algorithms on basic benchmark functions with dimension = 30, 50.

	Dim	F1		F2		F3		F4	
		Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
MFOA	30	−6.24E−2	2.34E−2	1.98E-3	1.05E-3	1.36E1	6.51	0.152	0.274
	50	-9.92E-1	1.86E−2	1.04E-2	1.16E-2	2.01E1	5.73	1.70	1.05
IFFO	30	−0.3674	0.5879	7.66E1	4.59E1	3.58E2	1.72E2	2.86E1	2.58E1
	50	−5.68E-2	0.9113	3.01E2	2.29E2	1.78E2	1.04E2	2.60E2	1.79E2
FOA	30	−9.54E−2	2.16E−2	1.05E-7	1.77E-7	5.29E1	2.45	3.98E-2	4.46E-2
	50	−8.37E−2	2.86E−2	1.21E-6	2.98E-6	5.07E1	2.44	2.10E-1	5.91E-1
MSFOA	30	−0.9999	9.14E−5	**6.24E−8**	**6.55E−8**	3.1462	4.5233	0.4139	0.4549
	50	−0.9999	9.03E−5	**7.60E−7**	**7.96E−7**	2.15E-8	2.20E-8	0.7171	4.17E-3
CMFOA	30	−0.9998	0.0001	3.35E−7	4.17E−7	0.0017	0.025	14.6682	15.3765
	50	−0.9906	0.036	6.53E−5	1.31E−5	0.0056	0.022	20.1622	11.2438
CEFOA	30	-1.00	0.00	1.2E-4	1.79E-4	**7.92E−14**	**1.98E-14**	1.98E−9	3.46E-8
	50	-1.00	0.00	9.52E-3	2.08E-4	**1.02E-13**	**2.10E-12**	4.99E-7	3.50E-7
OLFOA	30	**-1.00**	**0.00**	3.26E-3	3.68E-3	6.37E-1	5.13E-1	**7.35E-10**	**2.84E-8**
	50	**-1.00**	**0.00**	7.78E-3	5.59E-3	5.35	7.56	**3.26E-7**	**4.53E-6**
	Dim	F5		F6		F7		F8	
		Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
MFOA	30	4.74E-4	2.56E-4	1.97	1.09E−1	2.82E1	5.07E1	−4.45E2	5.45E1
	50	9.64E-4	1.75E-3	2.07	1.28E−1	2.01E1	5.27E1	−4.45E2	5.45E1
IFFO	30	3.28E−1	1.58E−1	0.94	1.76E-1	2.25E1	5.63E1	−4.43E2	3.38E−3
	50	0.5367	0.2219	2.56	1.16	1.50E2	3.12E2	−4.34E2	2.13E−1
FOA	30	1.03E2	4.26E2	1.05E−12	5.80E−1	8.64E1	1.45E1	−4.45E2	1.05E−1
	50	3.36E2	2.55E1	1.43E−11	4.26E−1	9.05E1	1.03E1	−4.41E2	4.33E−1
MSFOA	30	2.04E−5	2.39E−5	1.7160	1.7222	0.2159	0.2204	−448.26	1.8145
	50	2.46E−5	2.46E−5	2.1005	1.1212	0.787	0.588	−442.739	1.1184
CMFOA	30	0.0057	0.0078	0.0928	0.1	0.0269	0.0296	−449.916	0.0896
	50	0.0081	0.0114	0.1422	0.927	0.5809	0.5496	−442.976	0.1578
CEFOA	30	**1.95E−28**	**2.08E-10**	1.95E−12	2.08E−10	0.00	0.00	−450	0.00
	50	**1.02E-12**	**1.01E-6**	1.95E−12	1.88E-10	0.00	0.00	−450	0.00
OLFOA	30	1.50E-7	2.12E-7	**0.00**	**0.00**	**0.00**	**0.00**	**−450**	**0.00**
	50	3.25E-6	8.19E-6	**0.00**	**0.00**	**0.00**	**0.00**	**−450**	**0.00**
	Dim	F9		F10		F11		F12	
		Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
MFOA	30	−4.44E2	3.67	1.87E1	1.32E−1	2.65E-3	4.22E-3	1.89E2	4.31E1
	50	−4.34E2	2.58	2.05E1	3.45E−1	4.82E-2	3.52E-2	3.63E2	5.31E1
IFFO	30	−4.35E2	4.57E−4	5.17E−1	9.19E−1	1.1222	2.56	1.52E−2	2.75E−2
	50	−4.15E2	4.70E−3	3.2973	1.3559	1.02E1	2.76E1	5.11E−1	1.32E−2
FOA	30	−4.36E2	3.56	1.26E1	5.01E−1	4.15E1	1.58	5.07E1	1.70E1
	50	−4.35E2	1.16	3.02E1	1.05E−1	3.99E1	1.38	6.89E1	1.11E1
MSFOA	30	−447.014	3.1573	0.5921	5.98E−01	0.0536	5.42E−02	0.9661	9.67E−01
	50	−450	1.24E−08	0.6491	7.11E−01	0.1741	1.56E−01	1.6335	1.36
CMFOA	30	-449.18	0.0335	0.1205	0.1235	0.1313	0.1536	0.2594	0.2759
	50	−448.982	0.04273	0.4475	0.4387	0.4520	0.5165	33.1151	40.7375
CEFOA	30	−450	0.00	**1.60E−10**	**4.01E−10**	9.15E-14	1.55E-12	**8.89E-16**	**4.70E-15**
	50	−450	0.00	**1.69E-10**	**2.13E-10**	1.05E-13	1.95E-13	**1.09E−15**	**8.70E-14**
OLFOA	30	**−450**	**0.00**	1.32E-4	5.33E-4	**2.87E-20**	**5.91E-18**	6.44E-4	2.82E-4
	50	**−450**	**0.00**	3.72E-2	7.28E-2	**2.21E-16**	**4.39E-15**	7.17E-3	8.33E-3
	Dim	F13		F14		F15		F16	
		Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
MFOA	30	−5.57	6.11E-1	7.30E5	2.41E5	1.60E-4	2.27E-3	2.98E−1	2.24E−2
	50	−6.02	5.51E-1	2.46E5	1.41E5	6.11E-4	1.50E-3	4.00E−1	2.74E−2
IFFO	30	−17.25	1.15E−1	3.15E3	6.29E2	3.15E2	2.98E2	2.63	1.75
	50	−13.99	2.65E−1	2.265E2	5.56E2	4.18E3	1.69E3	2.44	2.71
FOA	30	−6.09	0.71	1.24E6	6.60E5	2.76E1	3.20E1	4.06E−1	7.73E−3
	50	−6.57	0.98	1.68E6	3.98E5	4.12E2	2.89E1	4.76E−1	7.60E−3
MSFOA	30	**−28.9297**	**7.15E−2**	15.9872	4.95E3	0.882	9.00E−01	0.1965	1.97E−01
	50	**−48.864**	**1.37E−1**	15.9872	4.95E3	1.635	4.37E−01	0.3122	1.16E−01
CMFOA	30	−23.0247	6.1269	827.128	4103.259	0.3156	0.3368	1.4198	1.4512
	50	−38.6263	10.5073	931.128	3357.637	2.2025	1.1627	1.0772	1.7336
CEFOA	30	-27.3134	1.05	**-1.89E4**	**3.60E3**	9.46E-8	4.42E-8	**0.00**	**0.00**
	50	−44.5835	1.52	**-1.69E4**	**2.64E3**	**1.06E-8**	**2.27E-8**	**0.00**	**0.00**
OLFOA	30	-27.7434	1.56E-3	1.08E5	2.26E5	**6.60E-8**	**8.85E-7**	5.13E-3	4.45E-3
	50	-46.9264	5.56E-3	1.97E5	5.50E6	7.87E-5	3.67E-4	3.32E-2	1.32E-2

benchmark functions with dimensions of 30 and 50. OLFOA exhibits computational benefit and superior performance compared with FOA, MFOA, IFFO, and CMFOA. OLFOA performed better than MSFOA and CEFOA with a faster convergence speed for most of the benchmark functions (F1, F4, F6, F7, F8, F9, F11, and F15), whereas it does not show outstanding performance compared to MSFOA in F2 and F13. Compared with CEFOA, the OLFOA does not

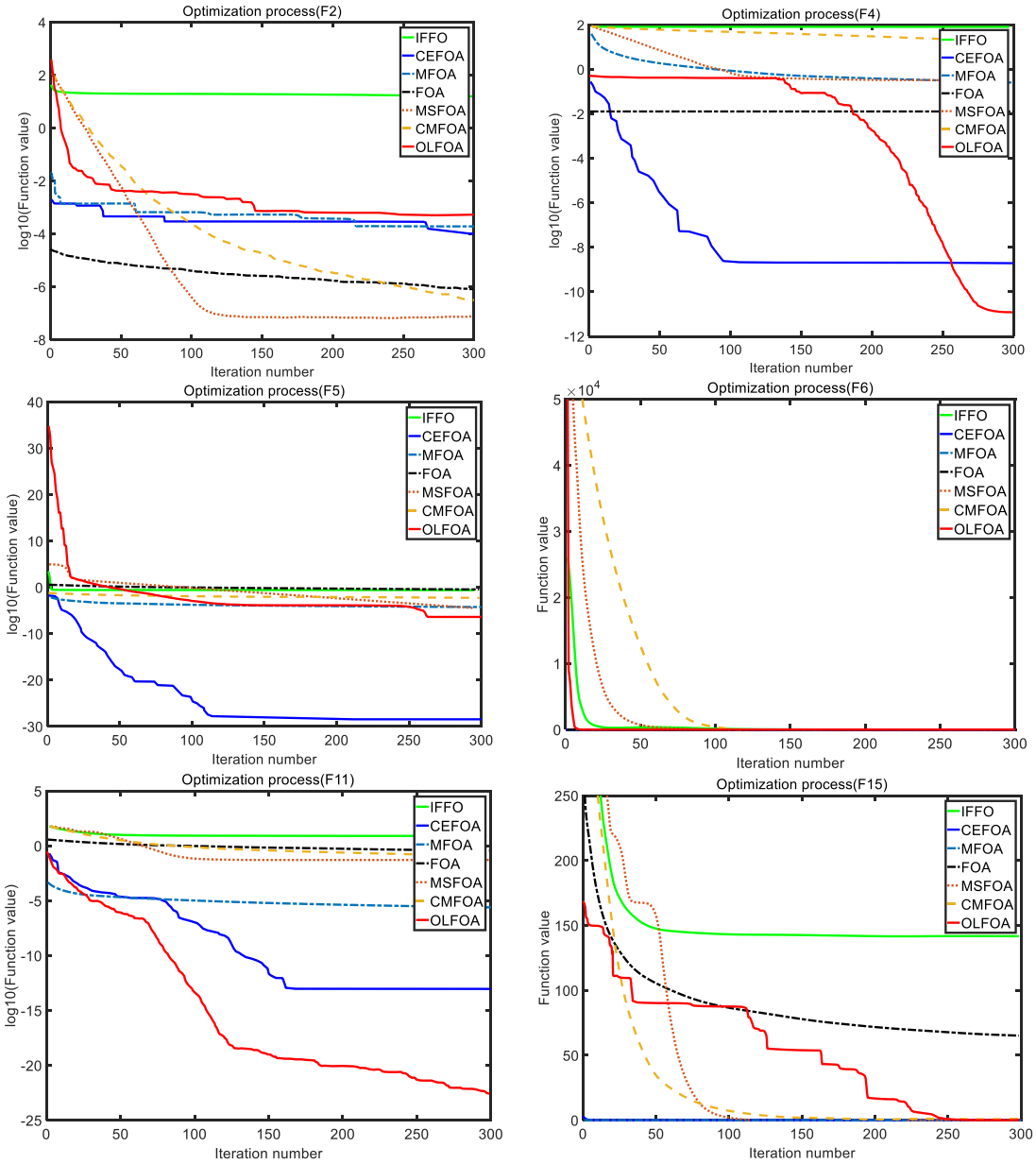


FIGURE 4. Convergence curve of multimodal functions for different algorithms with dimension = 30.

work well in F3, F5, F10, F12, F14 and F16. Thus, conclusion can be drawn that the proposed OLFOA prevails over other algorithms for most of the benchmark functions. In the case with a dimension of 50, similar results have been acquired except both the mean value and standard deviation of all algorithms have certain deterioration.

B. CONVERGENCE ANALYSIS

To describe the convergence performance of the algorithms visually, Figs. 4 and 5 provide the average convergence results of randomly selected test functions (F2, F4, F5, F6, F11, and F15) after 300 iterations with dimensions of 30 and 50, respectively. The dB fitness value is used and displayed in the *y-axis* of the figures in the optimization of F2, F4, F5, and F11. Only for benchmark functions F2 and F5, OLFOA does not show outstanding performance compared to other algorithms in terms of the convergence and optima. However, for the other unimodel benchmark functions, OLFOA shows its obvious advantage in stability and efficiency in acquiring the global optimum, especially for benchmark functions F11 and F15. For benchmark functions F4 and F6, OLFOA obtained the best fitness value and showed a fast convergence speed. These results indicate that the proposed OLFOA algorithm outperforms numerous improved FOA methods (IFFO, CEFOA, CMFOA, MFOA, and MSFOA) and the original FOA in global optimum and convergence capability.

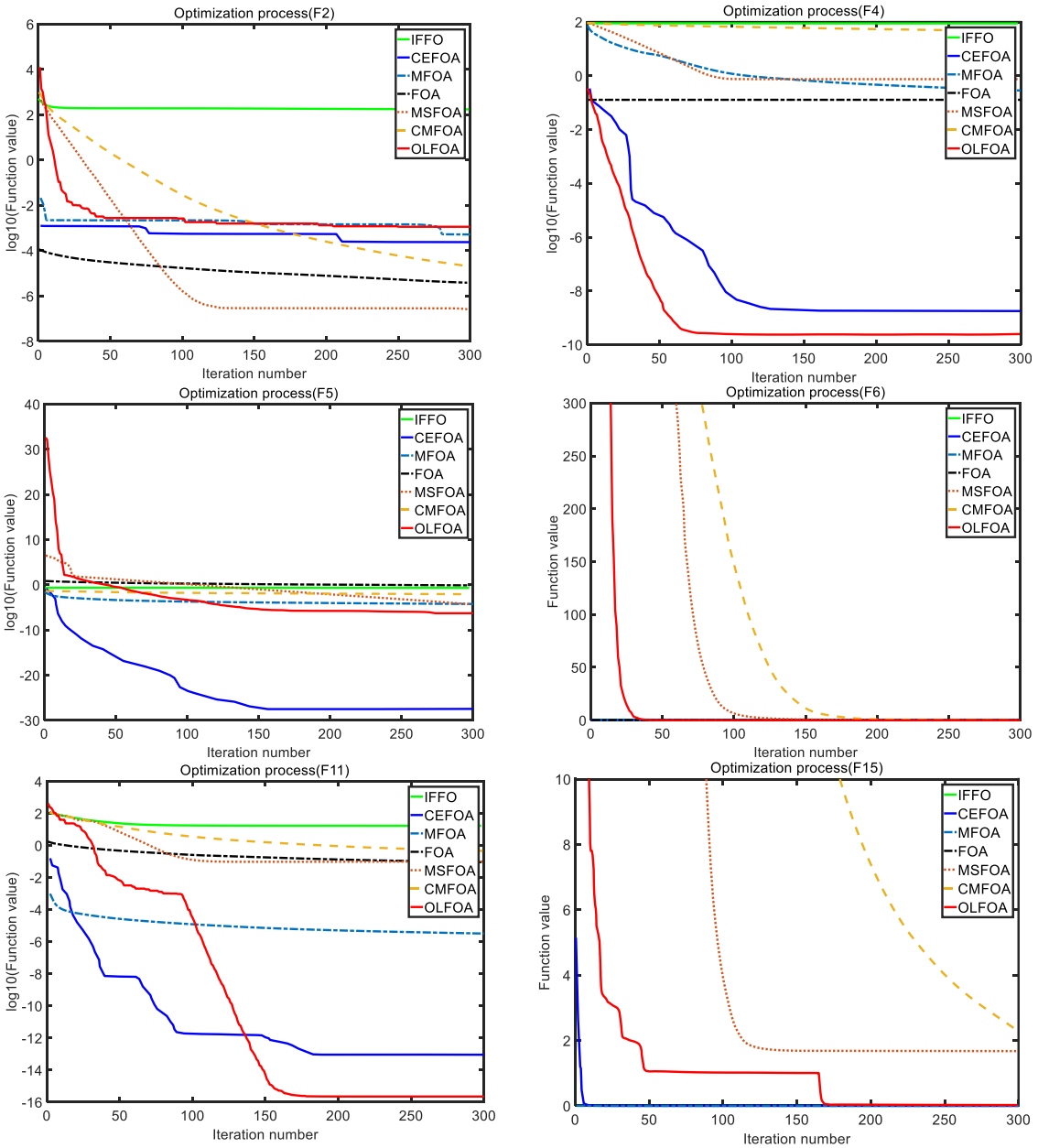


FIGURE 5. Convergence curve for multimodal functions with dimension = 50.

C. EXTRA EXPERIMENTAL RESULTS

Considering the array synthesis application of the proposed algorithm has a specific range of [0, 1], it is necessary to carry out tests on benchmark functions in the specific range for further performance verification. In this subsection, OLFOA is compared with CEFOA, and the recently proposed improved FOAs in Table 5. The search space of fruit fly individuals are set as [0,1].

Table 6 shows the mean value and the standard deviation of those FOAs for different benchmark functions in the range of [0, 1] with a dimension of 30. Conclusion can be drawn that the performance of CEFOA deteriorates severely when applied in such test functions, whereas the proposed OLFOA obtains better and more stable results when compared with other improved FOA algorithms.

TABLE 5. Comparison algorithms in extra tests.

Algorithm	Method	Authors and reference
AE-LGMS-FOA	Averager engine linear generation mechanism of candidate solution of FOA	A. Darvish et al. [21]
IFOA	Improved fruit fly algorithm	Xu et al. [31]

D. COMPARISON WITH OTHER ALGORITHMS

In this subsection, OLFOA is compared with other different types of improved algorithms (RMPSO, BLPSO, JADE, and

TABLE 6. Results of extra basic benchmark functions with dimension = 30.

	Dim	F1		F2		F3		F4	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std
AE-LGMS-FOA	30	−1.04E−1	3.57E−2	3.31E−2	7.76E−2	4.49	3.36	7.95E-4	1.54E−4
IFOA	30	−2.19E−1	4.10E−2	3.54E−2	1.82E−2	3.18	2.68	6.68E−6	2.97E−6
CEFOA	30	−0.51	3.27E−1	232.5	6.77E-1	7.54E5	8.82E−5	0.7	5.99E−4
OLFOA	30	**-1**	**0.00**	**2.9E-3**	**1.78E-3**	**9.37E-1**	**3.34E-1**	**5.27E-9**	**3.75E-8**
	Dim	F5		F6		F7		F8	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std
AE-LGMS-FOA	30	5.96E−7	8.58E−6	5.46E-23	4.14E−23	3.33E−25	4.90E−25	−4.49E2	9.12E−14
IFOA	30	6.68E−6	2.85E−6	8.82E−26	2.65E−26	6.47E−27	3.43E−26	−4.49E2	2.05E−17
CEFOA	30	20	2.79E−2	15	1.02E−15	4.2E3	5.87E−3	−4.35E2	6.65E−24
OLFOA	30	**1.87E-15**	**2.5E-14**	**0.00**	**0.00**	**0.00**	**0.00**	**−450**	**0.00**
	Dim	F9		F10		F11		F12	
		Mean	Std	Mean	Std	Mean	Std	Mean	Std
AE-LGMS-FOA	30	−4.49E2	7.14E−9	3.98E−7	9.66E−6	5.68E−11	4.89E−10	2.69E−23	6.77E−23
IFOA	30	−4.48E2	5.31E−12	7.29E−9	4.47E−7	2.74E−12	2.78E−10	7.32E−21	5.02E−21
CEFOA	30	−4.35E2	8.83E−30	3.57	5.14E−1	4.04E−4	1.91E−4	5.45E−1	1.46E−2
OLFOA	30	**−450**	**0.00**	**3.64E-14**	**1.12E-14**	**5.74E-20**	**1.32E-18**	**0.00**	**0.00**
	Dim	F13		F15		F16			
		Mean	Std	Mean	Std	Mean	Std		
AE-LGMS-FOA	30	−28.57	2.12E-3	4.98E-4	5.79E-3	4,63E−1	4.24E−1		
IFOA	30	−27.71	1.78E-5	1.89E-4	2.18E-4	5.17E−2	2.98E−2		
CEFOA	30	−23.84	7.87E-9	4.06E2	8.00E1	6.28E−1	3.05E−1		
OLFOA	30	**-29**	**0**	**1.95E-8**	**1.50E-7**	**3.13E-3**	**7.71E-3**		

NGHS) to justify its performance. The parameter settings are the same as those in Part A of Section IV.

Table 7 shows the mean value and standard deviation of those algorithms for different benchmark functions with dimensions of 30 and 50. Obviously, OLFOA exhibits its superior advantage in global optimum and convergence stability when compared with RMPSO, BLPSO, and JADE. When compared with NGHS, OLFOA performs better for most of the benchmark functions (F1, F2, F4, F5, F6, F7, F8, F9, F10, F11, F12, F13, F15, and F16) with fast and stable convergence speed, whereas OLFOA does not show competitive performance advantage compared to NGHS for F3 and F14.

From all these results and analysis above, conclusions can be drawn below:

1) OLFOA obtains much more superior values among all the algorithms in most cases. OLFOA can achieve excellent results with high stability in different dimensional and population problems.

2) OLFOA shows a strong local search capability and stable convergence rate among most of the algorithms.

3) OLFOA can demonstrate its competitive advantage in the benchmark tests in the range of [0, 1], which exactly corresponds to the application considered later in this paper.

V. OLFOA APPLICATION

In this section, OLFOA is applied into several different subarray phased array synthesis problems to prove its performance. In the simulation, the excitation of each subarray element is set as one dimension of the individual fruit fly, and the excitation of the entire subarray antenna element constitutes the location of individual fruit fly, the number of the swarm is set as 80, T is set as 900, $Tfin$ is set as 0.001×10^{-10}, the weighting factor wf is set as 0.4, and the maximum iteration number is set as 100.

A. ANTENNA ARRAY NOTATION

Considering a planar array antenna with N rows and M columns of elements arranged in a rectangular grid in the x- and y-$axis$, then the radiation pattern of planar array can be expressed as:

$$FF(\theta,\varphi) = \sum_{n=1}^{N}\sum_{m=1}^{M} I_{mn} f_{mn}(\theta,\varphi) \times e^{-j(kd_{xmn}\sin\theta\cos\varphi + kd_{ymn}\sin\theta\sin\varphi + \phi_{mn})} \quad (36)$$

where I_{mn} is the element excitation current amplitude, ϕ_{mn} is the element excitation current phase, $f_{mn}(\theta,\varphi)$ is the array element pattern, d_{xmn} and d_{ymn} are the distance from array element to axis origin in x-and y-$axis$ respectively.

Put this array antenna into the subarray tiling and each subarray consists of C elements. Then the maximum number of subarrays (L) that can be used to tile the entire aperture is $L = N \cdot M/C$[2]. If the shape of the subarray is the same, after rotating and folding, eight different shapes exist on the array aperture. Furthermore, if the subarray is symmetrical, the subarray shapes are reduced to four or even two [3].

In this study, the subarray tiling method based on X algorithm [7] is adopted to achieve the exact partition of subarrays on the array aperture. After the exact partition, the radiation

TABLE 7. Results of five algorithms on basic benchmark functions with dimension = 30, 50.

	Dim	F1		F2		F3		F4	
		Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
RMPSO	30	−1.04E-3	2.15E−2	2.18	1.59	5.93	1.46	1.709E-1	4.51E−1
	50	−7.15E−2	3.733E−2	1.87E1	2.13E1	8.19E1	4.51E1	1.93	4.74
BLPSO	30	−6.47E−2	1.04E−2	3.54E−2	2.80E−2	2.10E−1	3.08E−2	8.13E−1	2.65E−1
	50	−5.72E−2	3.54E−2	2.99E−2	1.96E−2	2.04E−1	2.87E−2	5.24E−1	2.85E−1
JADE	30	−3.57E−1	3.12E−1	4.27E−2	5.19E−2	8.54E−1	2.97E−2	6.07E−4	2.48E−4
	50	−3.14E−1	2.92E−1	3.86E−2	4.24E−2	7.11E−1	2.49E−2	5.67E−4	1.98E−4
NGHS	30	−1.03E−4	4.47E−5	7.48E−3	2.04E−3	**2.80E−6**	**3.16E−6**	1.44E−4	1.29E−3
	50	−1.01E−4	2.15E−5	**5.22E−3**	**1.09E−3**	**3.02E−6**	**2.92E−6**	1.04E−4	7.20E−4
OLFOA	30	**-1.00**	**0.00**	**3.26E-3**	**3.68E-3**	6.37E-1	5.13E-1	**7.35E-10**	**2.84E-8**
	50	**-1.00**	**0.00**	7.78E-3	5.59E-3	5.35	7.56	**3.26E-7**	**4.53E-6**
	Dim	**F5**		**F6**		**F7**		**F8**	
		Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
RMPSO	30	3.71E−1	5.48E−1	6.67E-1	1.44	5.78E−1	2.26E−1	−4.41E2	3.72E−4
	50	6.35	4.42	1.92	1.77	3.56	1.85	−4.23E2	1.36E−4
BLPSO	30	1.12E−1	5.29E−2	1.78E−4	4.22E−4	2.88E−4	5.60E−3	−4.40E2	1.41E−3
	50	1.08E−1	6.19E−2	1.25E−4	3.59E−4	1.93E−4	5.11E−3	−4.40E2	1.05E−3
JADE	30	1.12E−1	5.29E−2	1.78E−4	4.22E−4	2.88E−4	5.60E−3	−4.43E2	3.89E−1
	50	1.17E−5	3.51E−5	1.19E−5	3.08E−6	5.70E−5	2.03E−5	−4.33E2	2.94E−1
NGHS	30	4.55E−7	3.24E−6	6.99E−8	3.18E−8	1.50E−6	8.62E−8	−4.42E2	1.06E−1
	50	**2.44E−7**	**2.28E−6**	6.14E−8	1.27E−8	9.12E−7	7.29E−8	−4.40E2	1.01E−1
OLFOA	30	1.50E-7	2.12E-7	**0.00**	**0.00**	**0.00**	**0.00**	**−450**	**0.00**
	50	3.25E-6	8.19E-6	**0.00**	**0.00**	**0.00**	**0.00**	**−450**	**0.00**
	Dim	**F9**		**F10**		**F11**		**F12**	
		Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
RMPSO	30	−4.32E2	3.13E−1	4.47E−2	6.69E−2	8.23E−1	8.46E−1	9.82	6.77
	50	−4.25E2	2.35E−1	5.89	4.61	6.03E−1	2.73E−1	1.35E1	1.22E1
BLPSO	30	−4.41E2	1.35E−2	9.27E−3	2.27E−3	4.06E−2	1.41E−1	7.32E−2	5.02E−2
	50	−4.40E2	1.24E−08	7.19E−3	5.98E−01	3.41E−2	5.42E−02	5.32E−2	9.67E−01
JADE	30	−4.33E2	5.70E−1	8.06E−1	5.14E−1	3.78E−2	1.79E−2	5.45E−1	1.46E−2
	50	−4.36E2	5.10E−1	7.43E−1	4.58E−1	3.20E−2	1.51E−2	3.75E−1	1.11E−2
NGHS	30	−4.45E2	1.00E−2	2.56E−3	3.80E−3	7.01E−4	4.11E−4	3.22E−2	1.12E−2
	50	−4.44E2	1.04E−2	**2.77E−3**	**2.63E−3**	8.54E−4	2.71E−4	3.02E−2	1.02E−2
OLFOA	30	**−450**	**0.00**	**1.32E-4**	**5.33E-4**	**2.87E-20**	**5.91E-18**	**6.44E-4**	**2.82E-4**
	50	**−450**	**0.00**	3.72E-2	7.28E-2	**2.21E-16**	**4.39E-15**	**7.17E-3**	**8.33E-3**
	Dim	**F13**		**F14**		**F15**		**F16**	
		Mean	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
RMPSO	30	−2.33	4.66	2.54E5	1.151E4	4.77E1	7.82E1	3.54E−1	2.44E−1
	50	−1.51	6.57	2.76E5	2.51E4	5.12E2	4.43E2	8.98E−1	4.89E−1
BLPSO	30	−1.24	2.45	8.75E4	3.67E4	3.27	3.07E-01	2.71E−2	1.89E−2
	50	−1.15	1.59	7.05E4	3.13E4	2.18	1.53E−1	2.0E−2	1.80E−2
JADE	30	−5.05	1.23	4.22E3	7.23E3	4.06E2	8.00E1	6.28E−1	3.05E−1
	50	−4.84	1.2	3.11E3	5.03E3	3.54E2	4.12E1	3.14E−1	3.51E−1
NGHS	30	−6.00	1.07E−1	**1.05E3**	**3.91E3**	4.80E−2	1.46E−2	2.48E−2	9.22E−3
	50	−6.22	8.25E−2	**8.12E2**	**4.58E3**	4.87E−2	7.23E−3	**2.08E−2**	**1.29E−2**
OLFOA	30	**-27.7434**	**1.56E-3**	1.08E5	2.26E5	**6.60E-8**	**8.85E-7**	**5.13E-3**	**4.45E-3**
	50	**-46.9264**	**5.56E-3**	1.97E5	5.50E6	**7.87E-5**	**3.67E-4**	3.32E-2	1.32E-2

pattern of the subarray antenna can be expressed as

$$FF(\theta,\varphi)=\sum_{l=0}^{L} I_l^{sub} e^{-j\phi_l} \sum_{\{m,n|l_{mn}=l\}} I_{mn}^{ele} f_{mn}(\theta,\varphi) e^{-j\phi_{mn}} \times e^{-j(kd_{xmn}\sin\theta\cos\varphi + kd_{ymn}\sin\theta\sin\varphi)} \tag{37}$$

where I_l^{sub} is the normalized subarray excitation current amplitude, I_{mn}^{ele} is the normalized element excitation current amplitude, ϕ_l is the subarray excitation current phase, $\{m, n|l_{mn} = l\}$ is the set of antenna elements longing to the lth subarray, ϕ_i is the element excitation current phase.

For engineering practicality, the optimization object of subarray antenna array synthesis problems is set as the subarray excitation current amplitude (I_l^{sub}), which ranges from 0 to 1. The optimization target is the maximum sidelobe level of the antenna pattern. The fitness function can be expressed as:

$$\text{Fitness} = \underset{(\theta,\varphi)\in FF_S}{\text{abs}}\left(\frac{FF}{\max(FF)}\right) \tag{38}$$

where FF_S represents the area of sidelobe.

B. RESULTS OF ARRAY SYNTHESIS

The first synthesis case is a 40-element array (5 × 8) with half-wavelength spacing, and a rectangular subarray consisting of two connected elements is desired [6], as shown in Fig. 6. Fig. 7 presents the far-field radiation pattern of the arrays optimized after 100 iterations, and the corresponding *PSL* is −23.46 dB, which is 4.57 dB lower than that in [6].

Arrays of 128 (8 × 16) and 432 (18 × 24), with inter-element spacing of half-wavelength, are considered in the second and the third synthesis cases, respectively.

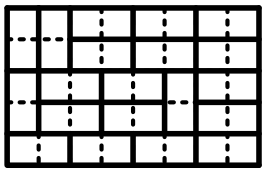


FIGURE 6. Structures of 5 × 8 array tiling optimized with *X* algorithm.

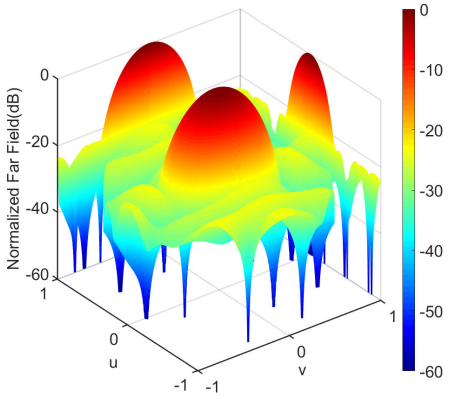


FIGURE 7. Radiation pattern for the arrays optimized with OLFOA.

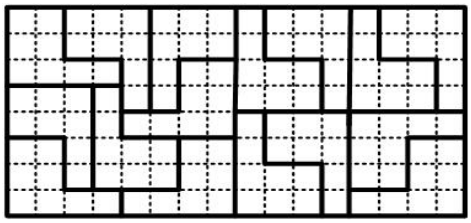


FIGURE 8. Structures of 8 × 16 array tiling optimized with *X* algorithm.

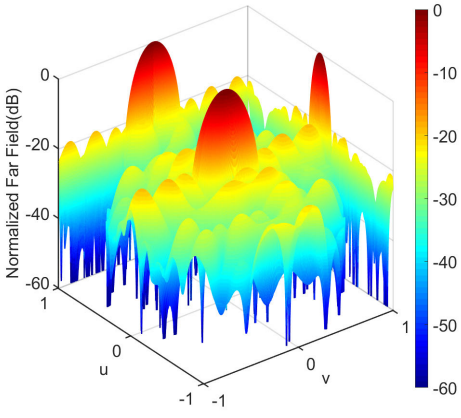


FIGURE 9. Radiation pattern for the arrays optimized with OLFOA.

A subarray with an L-octomino shape consisting eight connected elements is desired. Fig. 8 and 10 show the acceptable exact partition of the aperture by using the *X* algorithm. Fig. 9 and 11 present the far-field radiation patterns of the arrays optimized by OLFOA after 100 iterations. The corresponding *PSL* for these two arrays are −19.98 and −30.64 dB, respectively.

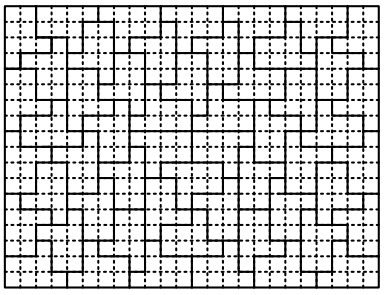


FIGURE 10. Structures of 18 × 24 array tiling optimized with *X* algorithm.

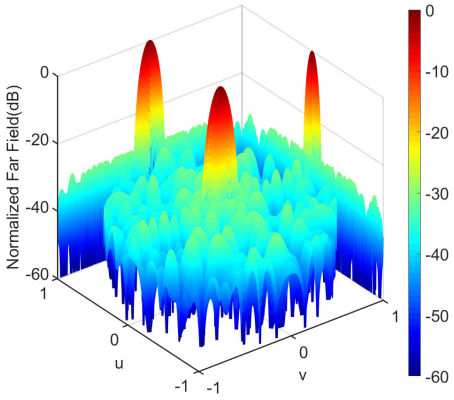


FIGURE 11. Radiation pattern for the arrays optimized with OLFOA.

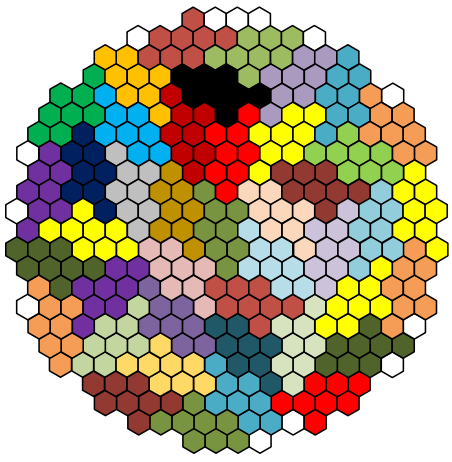


FIGURE 12. Structures of circular aperture array tiling optimized with the *X* algorithm.

The fourth case is a 349-element circular aperture array with half-wavelength spacing. The subarray consists of eight connected polyhex-shaped elements. As shown in Fig. 12, the array is divided into 55 subarrays by the *X* algorithm (including 42 eight-element polyhex-shaped subarrays and 13 independent elements) [7]. The subarray partition and the beam pointing of the array are set to be the same as in [7]. Fig. 13 displays the far-field radiation pattern of the arrays optimized after 100 iterations, and the corresponding *PSL* is −30.76dB. This optimization result is 3.56dB lower than the maximum sidelobe achieved in [7], which indicates the

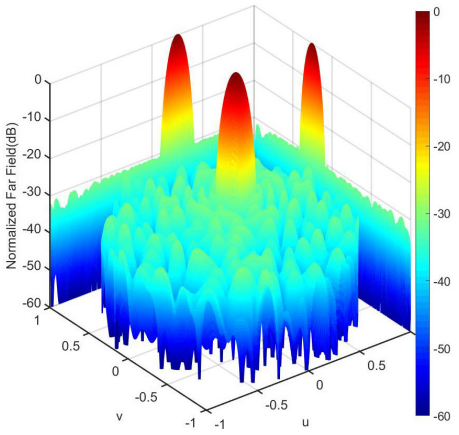


FIGURE 13. Radiation pattern for the arrays optimized with OLFOA.

TABLE 8. Performance comparison of different subarray antennas synthesized by different methods of three algorithms.

Synthesis case	PSL (dB)			
	Case 1	Case 2	Case 3	Case 4
Uniform excitation	−12.05	−13.17	−13.29	−17.34
LGMS-FOA [16]	−21.97	−17.19	−26.45	−25.47
AE-LGMS-FOA [21]	−22.63	−18.68	−28.28	−27.89
GA	−19.26	−15.74	−24.5	−24.14
DE	−18.13	−16.61	−25.1	−23.43
QPSO [17]	−18.64	−16.79	−24.4	−23.85
OLFOA	**−23.46**	**−19.98**	**−30.64**	**−30.76**

superiority of the proposed algorithm when applied in the design of irregular subarray antennas.

Table 8 presents the comparison results in terms of *PSL* optimized by those different algorithms under four irregular subarray cases. For fair comparison, the parameters of the other algorithms are set the same as those of the proposed algorithm. From Table 8, conclusion can be drawn that the proposed OLFOA shows the superior *PSL* performance when compared with the other algorithms for all these cases.

VI. CONCLUSION

In this paper, an advanced fruit fly optimization algorithm, namely, OLFOA has been proposed. Individuals are selected in a highly balanced way by adding the OQSM and SM mechanisms. The population diversity, global searching capability, and ability to escape from the local optimum of the algorithm are significantly improved. The OLFOA has been tested via numerous classical benchmark functions in comparison with other improved FOAs and advanced algorithms to demonstrate its capability. Then, OLFOA is applied to synthesize four different cases of subarray antenna array of planar and circular aperture. Simulation results show that the proposed algorithm presents a superior performance both in benchmark function test and the subarray antenna array synthesis problems.

REFERENCES

[1] R. J. Mailloux, S. G. Santarelli, and T. M. Roberts, "Array aperture design using irregular polyomino subarrays," in *Proc. IEEE Int. Symp. Phased Array Syst. Technol.*, Oct. 2010, pp. 740–744.

[2] P. Rocca, R. Chirikov, and R. J. Mailloux, "Polyomino subarraying through genetic algorithms," in *Proc. IEEE Int. Symp. Antennas Propag.*, Jul. 2012, pp. 1–2.

[3] R. Chirikov, P. Rocca, L. Manica, S. Santarelli, R. J. Mailloux, and A. Massa, "Innovative GA-based strategy for polyomino tiling in phased array design," in *Proc. 7th Eur. Conf. Antennas Propag. (EuCAP)*, Apr. 2013, pp. 2216–2219.

[4] B. H. Gwee and M. H. Lim, "Polyominoes tiling by a genetic algorithm," *Comput. Optim. Appl.*, vol. 6, no. 3, pp. 273–291, Nov. 1996.

[5] P. Rocca, R. J. Mailloux, and G. Toso, "GA-based optimization of irregular subarray layouts for wideband phased arrays design," *IEEE Antennas Wireless Propag. Lett.*, vol. 14, pp. 131–134, 2015.

[6] N. Anselmi, P. Rocca, and A. Massa, "Irregular phased array tiling by means of analytic schemata-driven optimization," *IEEE Trans. Antennas Propag.*, vol. 65, no. 9, pp. 4495–4510, Sep. 2017.

[7] Z.-Y. Xiong, Z.-H. Xu, S.-W. Chen, and S.-P. Xiao, "Subarray partition in array antenna based on the algorithm X," *IEEE Antennas Wireless Propag. Lett.*, vol. 12, pp. 906–909, 2013.

[8] Z. Amir and B. A. Arand, "GA-based approach to phase compensation of large phased array antennas," *J. Syst. Eng. Electron.*, vol. 29, no. 4, pp. 723–730, Aug. 2018.

[9] H. Li, Y. Jiang, Y. Ding, J. Tan, and J. Zhou, "Low-sidelobe pattern synthesis for sparse conformal arrays based on PSO-SOCP optimization," *IEEE Access*, vol. 6, pp. 77429–77439, 2018.

[10] B. Gonzalez-Valdes, G. Allan, Y. Rodriguez-Vaqueiro, Y. Álvarez, S. Mantzavinos, M. Nickerson, B. Berkowitz, J. A. Martínez-Lorenzo, F. Las-Heras, and C. M. Rappaport, "Sparse array optimization using simulated annealing and compressed sensing for near-field millimeter wave imaging," *IEEE Trans. Antennas Propag.*, vol. 62, no. 4, pp. 1716–1722, Apr. 2014.

[11] D. Z. Zhu, P. L. Werner, and D. H. Werner, "Design and optimization of 3-D frequency-selective surfaces based on a multiobjective lazy ant colony optimization algorithm," *IEEE Trans. Antennas Propag.*, vol. 65, no. 12, pp. 7137–7149, Dec. 2017.

[12] R.-Q. Wang, Y.-C. Jiao, H. Zhang, and Z. Zhou, "Synthesis of unequally spaced linear arrays using modified differential evolution algorithm," *IET Microw., Antennas Propag.*, vol. 12, no. 12, pp. 1908–1912, Oct. 2018.

[13] S. K. Mahto and A. Choubey, "A novel hybrid IWO/WDO algorithm for interference minimization of uniformly excited linear sparse array by position-only control," *IEEE Antennas Wireless Propag. Lett.*, vol. 15, pp. 250–254, 2016.

[14] X. Zhang, X. Zhang, and L. Wang, "Antenna design by an adaptive variable differential artificial bee colony algorithm," *IEEE Trans. Magn.*, vol. 54, no. 3, Mar. 2018, Art. no. 7201704.

[15] W.-T. Pan, "A new fruit fly optimization algorithm: Taking the financial distress model as an example," *Knowl.-Based Syst.*, vol. 26, no. 2, pp. 69–74, Feb. 2012.

[16] D. Shan, G. Cao, and H. Dong, "LGMS-FOA: An improved fruit fly optimization algorithm for solving optimization problems," *Math. Problems Eng.*, vol. 2013, Aug. 2013, Art. no. 108768.

[17] J. Sun, W. Xu, and B. Feng, "A global search strategy of quantum-behaved particle swarm optimization," in *Proc. IEEE Conf. Cybern. Intell. Syst., Dec.*, Dec. 2004, pp. 111–116.

[18] F. van den Bergh and A. P. Engelbrecht, "A new locally convergent particle swarm optimiser," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, vol. 3, Oct. 2002, p. 6.

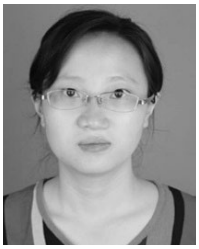
[19] Q. Zhang and Y.-W. Leung, "An orthogonal genetic algorithm for multimedia multicast routing," *IEEE Trans. Evol. Comput.*, vol. 3, no. 1, pp. 53–62, Apr. 1999.

[20] E. H. L. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. New York, NY, USA: Wiley, 1989.

[21] A. Darvish and A. Ebrahimzadeh, "Improved fruit-fly optimization algorithm and its applications in antenna arrays synthesis," *IEEE Trans. Antennas Propag.*, vol. 66, no. 4, pp. 1756–1766, Apr. 2018.

[22] Q.-K. Pan, H.-Y. Sang, J.-H. Duan, and L. Gao, "An improved fruit fly optimization algorithm for continuous function optimization problems," *Knowl.-Based Syst.*, vol. 62, pp. 69–83, May 2014.

[23] M. Mitić, N. Vuković, M. Petrović, and Z. Miljković, "Chaotic fruit fly optimization algorithm," *Knowl.-Based Syst.*, vol. 89, pp. 446–458, Nov. 2015.
[24] Y. Zhang, G. Cui, J. Wu, W.-T. Pan, and Q. He, "A novel multi-scale cooperative mutation fruit fly optimization algorithm," *Knowl.-Based Syst.*, vol. 14, pp. 24–35, Dec. 2016.
[25] X. Han, Q. Liu, H. Wang, and L. Wang, "Novel fruit fly optimization algorithm with trend search and co-evolution," *Knowl.-Based Syst.*, vol. 141, pp. 1–17, Feb. 2018.
[26] W.-T. Pan, "Mixed modified fruit fly optimization algorithm with general regression neural network to build oil and gold prices forecasting model," *Kybernetes*, vol. 43, no. 7, pp. 1053–1063, 2014.
[27] B. Jana, S. Mitra, and S. Acharyya, "Repository and mutation based particle swarm optimization (RMPSO): A new PSO variant applied to reconstruction of gene regulatory network," *Appl. Soft Comput.*, vol. 74, pp. 330–355, Jan. 2019.
[28] X. Chen, H. Tianfield, C. Mei, W. Du, and G. Liu, "Biogeography-based learning particle swarm optimization," *Soft Comput.*, vol. 21, no. 24, pp. 7519–7541, Dec. 2017.
[29] J. Zhang and A. C. Sanderson, "JADE: Adaptive differential evolution with optional external archive," *IEEE Trans. Evol. Comput.*, vol. 13, no. 5, pp. 945–958, Oct. 2009.
[30] D. Zou, L. Gao, J. Wu, and S. Li, "Novel global harmony search algorithm for unconstrained problems," *Neurocomputing*, vol. 73, no. 3, pp. 3308–3318, Aug. 2010.
[31] X. Tian and J. Li, "A novel improved fruit fly optimization algorithm for aerodynamic shape design optimization," *Knowl.-Based Syst.*, vol. 179, pp. 77–91, Sep. 2019.



WENTAO LI was born in Shaanxi, China. She received the B.E. degree in electromagnetic engineering and the Ph.D. degree in electromagnetic fields and microwave technology from Xidian University, Xi'an, China, in 2006 and 2010, respectively. She was a Visiting Scholar with The University of Texas at Austin, from August 2015 to August 2016. She is currently an Associate Professor with the School of Electronic Engineering, Xidian University. Her research interests include evolutionary optimization techniques, antenna arrays, and MIMO antennas.

YUDONG ZHANG was born in Shaanxi, China, in 1995. He received the B.E. degree in electronic information engineering from Xidian University, Xi'an, China, in 2017. He is currently pursuing the master's degree in electronics and communication engineering with the Science and Technology on Antenna and Microwave Laboratory, Xidian University. His research interests include array synthesis, evolution optimization techniques, and antenna designs.

XIAOWEI SHI was born in Guangdong, China, in 1963. He received the B.S. degree in radio physics, the M.Eng. degree in electrical engineering, and the Ph.D. degree in electromagnetic field and microwave technology from Xidian University, Xi'an, China, in 1982, 1990, and 1995, respectively. From 1996 to 1997, he was a Cooperator with the Electronics and Telecommunications Research Institute of Korea, for his postdoctoral research work. He has been a Professor and a Ph.D. Student Advisor with Xidian University. His research interests include the theory of microwave networks, microwave measurement, electromagnetic inverse scattering, the theory of electromagnetic variation, electromagnetic compatibility, and smart antenna. In recent years, he mainly studies the smart antenna.