(Received 21 February 2024, Accepted 12 April 2024, Scheduled 5 May 2024)

Suppression of Peak Sidelobe Level in Linear Symmetric Antenna Arrays Using Hybrid Grey Wolf and Improved Bat Algorithm

Jiao Zhang*, Jiajun Chu, Yufeng Liu, and Wenmei Zhang

School of Physics and Electronic Engineering, Shanxi University, Taiyuan, China

ABSTRACT: In this paper, the Hybrid Grey Wolf and Improved Bat Optimization Algorithm (HWIBO) is proposed to reduce the peak sidelobe level (PSLL) of linear symmetric array synthesis with aperture and element spacing constraints. The HWIBO utilizes both the Grey Wolf Optimization (GWO) and Improved Bat Algorithms (IBA) simultaneously to optimize PSLL. Each iteration generates two sets of results, and the optimal result is chosen for the next loop. Compared to other algorithms used in simulation of antenna sidelobe suppression, the HWIBO not only inherits the fast convergence advantage of the IBA which enhances population diversity but also possesses the strong global search capability of the GWO. This helps the IBA escape local optima and strengthens the global search capability during the later stages of algorithm iterations. Finally, the simulation results demonstrate the successful reduction of PSLL under various constraints, confirming the effectiveness of the hybrid algorithm.

1. INTRODUCTION

The research on linear array antennas with randomly distributed elements started in the 1960s. They have demonstrated their significance in various engineering applications, particularly in ground radar and satellite communications [1]. A nonuniform linear array is generally divided into two types: thinned arrays and sparse arrays. Compared with thinned arrays that achieve thinned layout by zeroing or not using part of the array elements, sparse arrays have higher degrees of freedom and can more effectively reduce the PSLL, so it has attracted wide attention [2]. Optimizing the positions of complex array elements in a random distributed array has always been a challenge. In the past decades, researchers have put forth various algorithms to achieve a smaller PSLL, such as differential evolution algorithm and its improvement [2, 9], dynamic programming [3], ant colony optimization [4], binary particle swarm algorithm [5], genetic algorithm and its improvement [6–8].

The Bat Algorithm (BA) is a heuristic search algorithm proposed in 2010, which is based on swarm intelligence [10]. BA can rapidly discover the optimal solution and enhance local search by generating novel solutions through random flights around the optimal solution. However, it also has some defects, such as excessively rapid convergence and a tendency to get trapped in local minima. To overcome these shortcomings, [11] uses interval type-2 fuzzy logic to change the parameters of BA and help the algorithm to jump out of local minimum cycles. Ref. [12] improved the bat algorithm by incorporating a dynamic local search strategy, which enhanced the population diversity of BA and effectively reduced the PSLL.

The GWO was first proposed in 2014 [13]. The algorithm exhibits remarkable proficiency in global optimization and rapid convergence, but it also has weak local search ability and poor algorithm stability.

Each algorithm comes with its own set of pros and cons. Hybrid algorithm can combine the advantages of algorithms and compensate for the shortcomings. In this paper, we combine GWO [13] and IBA [12] to optimize the element spacing of linear sparse arrays. The hybrid algorithm is based on the coevolution behavior of IBA and GWO [14]. GWO is used to help IBA escape local minimum and enhance global search capability, while IBA is employed to improve the GWO's capability in conducting local searches, and the two algorithms cooperate to find the optimal result.

The specific arrangement is as follows. Section 2 briefly introduces the synthesis formula of symmetric linear arrays. Section 3 describes the HWIBO and the specific details of the HWIBO hybrid process. In Section 4, simulation experiments are conducted to analyze the data. In Section 5, the simulation results are summarized.

2. LINER ARRAYS SYNTHESIS

The odd-numbered symmetric linear array antenna structure is shown in Figure 1. We can see that the array of $2N + 1$ elements is placed symmetrically on both sides of the X-axis, and its x-half axis array length is $L/2$. The origin serves as the position for the 0th element, while the remaining elements exhibit symmetry with respect to the origin. Each element satisfies the isotropic condition, and the array elements are randomly spaced. The following equation represents the array pat-

* Corresponding author: Jiao Zhang (zhangjiao@sxu.edu.cn).

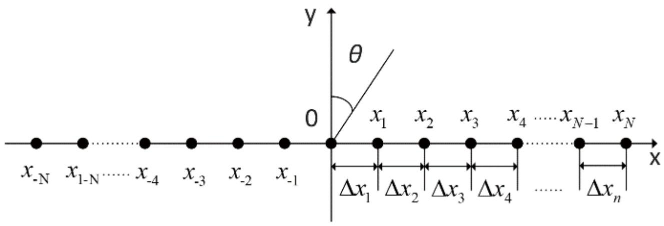

FIGURE 1. $(2N+1)$-element linear symmetric array structure.

tern function.

$$\begin{aligned}\mathrm{AF}(u) &= \sum_{-N}^{N} A_n e^{jkux_n+\varphi_n} \\ &= A_0 + 2\sum_{n=1}^{N} A_n \cos(kux_n + \varphi_n) \qquad (1)\end{aligned}$$

here A_n, φ_n, and x_n represent the amplitude, phase, and position of the nth element in the half axis, respectively. $k = 2\pi/\lambda$, λ is the wavelength, and $u = \sin\phi$, $\phi \in [-90°, 90°]$. In this paper, we assume that all elements have identical amplitude and phase, then we set $A_n = A_0 = 1$, $\varphi_n = 0$, $n = 0, 1, 2...N$. The AF(u) can be simplified as

$$\mathrm{AF}(u) = 2\sum_{n=1}^{N} \cos(kux_n) + 1 \qquad (2)$$

To achieve a lower PSLL for the symmetric linear array, the fitness function is expressed by the following formula.

$$F(A) = \mathrm{PSLL}_{\min} = \max\left|\frac{\mathrm{AF}(u_s)}{\mathrm{AF}_{\max}}\right| \qquad (3)$$

The main lobe reaches its maximum value at $\mathrm{AF}_{\max}$, while AF(u_s) indicates the presence of side lobe regions.

The optimization conditions mainly include the constraint of the spacing between adjacent elements and limitations on the aperture. The specific constraints are as follows

$$\begin{cases} \min_{\mathrm{PSLL}}(d_1, d_2, ..., d_n) \\ s.t\ d_{\min} \le d_i - d_j \le d_{\max} \\ 1 \le j \le i \le N \\ d_1 = 0;\ \ d_N = L \end{cases} \qquad (4)$$

where $d_{\min}$ represents the minimum value of the range of adjacent element spacing, and $d_{\max}$ represents the maximum value and $d_{\min} = 0.5\lambda$.

The purpose of the optimization is to reduce the PSLL. In the case of satisfying the constraints, the distance Δx_n between two adjacent elements is optimized to achieve the purpose of reducing PSLL.

3. HYBRID GREY WOLF AND IMPROVED BAT OPTIMIZATION ALGORITHM

Currently, there are two common mixing methods of hybrid algorithms. One approach is to use the output of an algorithm as the input for another algorithm to optimize the objective, which is called serial hybridization. Another approach is to simultaneously iterate two algorithms, comparing the results outputted by each iteration and selecting the optimal value as the initial value for the next iteration loop, which is called parallel hybridization.

3.1. Parallel Hybridization

Given the same initial values, each of IBA and GWO generates a current best solution in each iteration. Then, these two solutions are compared, and the optimal one is retained and carried into the next iteration loop. This process is repeated until all loops are iterated. The algorithm can be divided into several stages as follows.

In the first phase, the parameters are initialized, and both algorithms set the same initial population size, maximum number of iterations $\mathrm{iter}_{\mathrm{MAX}}$, and initial positions. $\mathrm{iter}_{\mathrm{MAX}} = 3000$, and population size is 200. The initial position is obtained from formula (5), where x_i represents the position of the ith array element along the X-axis. The variables $x_{\min}$ and $x_{\max}$ represent the upper and lower limits, respectively, of the search space, and *rand*$(0, 1)$ represents generate a random number from 0 and 1.

$$x_i = x_{\min} + (x_{\max} - x_{\min}) \cdot \mathit{rand}(0, 1) \qquad (5)$$

The positions of elements are judged and adjusted to ensure that they satisfy the constraints (subsequent changes in element position must also adjust the spacing to within the constraint conditions). Then the adjusted array element positions into formula (3); take the optimal value as the initial optimal solution of GWO and IBA; and enter the iterative loop.

In the second stage, the iteration begins, and two solutions are generated after each iteration.

The IBA process is as follows.

Step 1: Update the velocity and position of all bats, using formulas (6)–(9).

$$\begin{aligned} f_i &= \mathit{rand}(0, 1) \cdot (f_{\max} - f_{\min}) + f_{\min} & (6) \\ f_i' &= \frac{c + v_i^t}{c + v_g^t} f_i \left(1 + C_i \frac{g^t - x_i^t}{|g^t - x_i^t| + \varepsilon}\right) & (7) \\ v_i^{t+1} &= wv_i^t + \left(x_i^t - g^t\right) f_i' & (8) \\ x_i^{t+1} &= x_i^t + v_i^{t+1} & (9) \end{aligned}$$

If the current best location g is found by a bat, other bats will move closer to this location to find the best foraging location, where the minimum frequency $f_{\min} = 0$ and maximum frequency $f_{\max} = 1$. $C \in [0, 1]$ represents the compensation rate for the Doppler effect. c, v, and w respectively represent the speed of sound, the flight speed of bats, and the coefficient of inertia weight, where $v \in (0, 1)$ and $w \in [0.5, 1]$.

Step 2: Search for a better solution near the optimal solution. If *rand*$(0, 1) > r_i$, a local search is performed according to formula (10) [17]. Here, $r_i = 0.9$ represents the pulse emissivity, and A^t represents the average loudness of all bats. Otherwise, proceed to Step 3.

$$x_{\text{new}} = A^t \cdot N[0,1] + x_{\text{old}} \tag{10}$$

Step 3: If $rand(0,1) < A_i$ and $F(x_i) < F(g^t)$, the optimal solution is updated to the new solution. According to formulas (11) and (12), we can increase the loudness value and reduce the pulse emission frequency.

$$A_i^{t+1} = \alpha A_i^t \tag{11}$$

$$r_i^{t+1} = r_i^o\,[1 - \exp(-\gamma t)] \tag{12}$$

here, $\alpha = 0.9$ and $\gamma = 0.9$. $A_i^t \in (1,2)$ and $r_i^o \in (0,1)$.

$$A_i^t \to 0,\ r_i^t \to r_i^o, \quad \text{as } t \to \infty \forall 0 < \alpha,\ \gamma < 1 \tag{13}$$

Step 4: Update the current best solution.

The GWO process is given by following steps.

Step 1: The fitness function is calculated according to the initial position of the grey Wolf, and the wolves are ranked according to their fitness values. We define the top three optimal solutions as alpha (α), beta (β), and delta (δ) in turn. The other remaining wolves are defined as omega (ω). The optimization process is dominated by α, β and δ, and the ω wolves follow the hunting path.

Step 2: Update the positions of the gray wolves.

The process of encircling prey can be represented by formulas (14) to (17) [18].

$$\mathbf{D} = \left|-\mathbf{X}(t) + \mathbf{C} \cdot \mathbf{X}_P(t)\right| \tag{14}$$

$$\mathbf{X}(t+1) = -\mathbf{A} \cdot \mathbf{D} + \mathbf{X}_P(t) \tag{15}$$

$$\mathbf{A} = -\mathbf{a} + 2\mathbf{r}_1 \cdot \mathbf{a} \tag{16}$$

$$\mathbf{C} = 2 \cdot \mathbf{r}_2 \tag{17}$$

$$\mathbf{a} = 2 \cdot \left(1 - \frac{t}{\text{iterMAX}}\right) \tag{18}$$

where t denotes the current number of iterations, and $\mathbf{X}_P$, $\mathbf{X}$, and $\mathbf{D}$ denote the position of the prey, position of the Wolf, and distance of the Wolf from the prey, respectively. $\mathbf{A}$ and $\mathbf{C}$ are coefficient vectors, and $\mathbf{a}$ is a convergence factor that linearly decreases from 2 to 0. $\mathbf{r}_1$, $\mathbf{r}_2$ are randomly generated within [0, 1].

The hunting process can be represented by the following formula [18]

$$\begin{aligned} \mathbf{D}_\alpha &= \left|-\mathbf{X} + \mathbf{C}_1 \cdot \mathbf{X}_\alpha\right| \\ \mathbf{D}_\beta &= \left|-\mathbf{X} + \mathbf{C}_2 \cdot \mathbf{X}_\beta\right| \\ \mathbf{D}_\delta &= \left|-\mathbf{X} + \mathbf{C}_3 \cdot \mathbf{X}_\delta\right| \end{aligned} \tag{19}$$

$$\begin{aligned} \mathbf{X}_1 &= -\mathbf{A}_1 \cdot \mathbf{D}_\alpha + \mathbf{X}_\alpha \\ \mathbf{X}_2 &= -\mathbf{A}_2 \cdot \mathbf{D}_\beta + \mathbf{X}_\beta \\ \mathbf{X}_3 &= -\mathbf{A}_3 \cdot \mathbf{D}_\delta + \mathbf{X}_\delta \end{aligned} \tag{20}$$

$$\mathbf{X}(t+1) = \left(\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3\right)/3 \tag{21}$$

α leads β and δ surrounding the prey. In the tth iteration, the positions of the α, β, and δ wolves are represented by $\mathbf{X}_\alpha$, $\mathbf{X}_\beta$, and $\mathbf{X}_\delta$. Under the guidance of α, β, and δ, the new positions resulting from the $(t+1)$th iteration are calculated using formula (21).

With a decrease from 2 to 0, A decreases accordingly. When $|\mathbf{A}| < 1$, the wolves move closer to the prey in a converging manner, and when $|\mathbf{A}| > 1$, the gray wolves spread apart to search for better prey. The value of $\mathbf{C}$ is a random number from 0 to 2. When $\mathbf{C} > 1$, ω wolves will move quickly towards the prey, and when $\mathbf{C} < 1$, it will discourage wolves from approaching their prey.

Step 3: Evaluate the fitness function for all gray wolves and update the values and positions of α, β, and δ. More detailed information about GWO can be found in [16].

In the third stage, the best solutions from each iteration of IBA ($\text{IBA}_i^{\text{best}}$) and GWO ($\text{GWO}_i^{\text{best}}$) are compared. If all the loops are iterated, the optimal solution and its element positions are output. Otherwise, use the best solution and its element

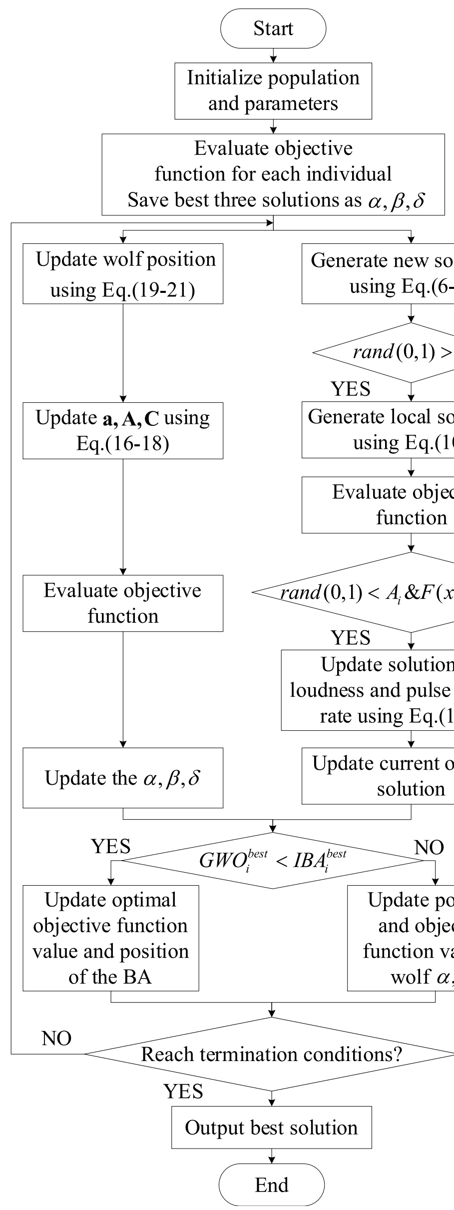

FIGURE 2. Flowchart of the hybrid grey wolf and improved bat optimization algorithm.

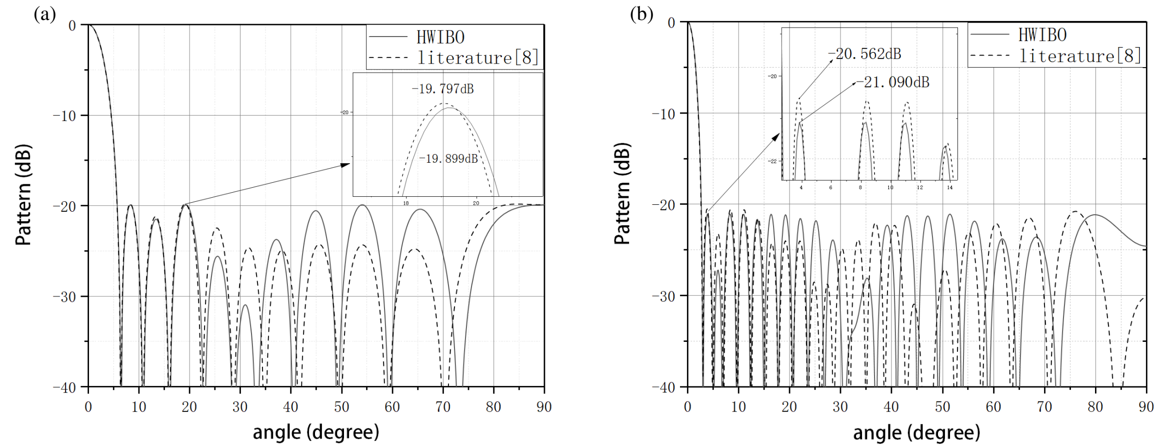

FIGURE 3. (a) Array synthesis diagram of 17-element, (b) array synthesis diagram of 37-elements.

positions from both methods as the initial values for the next iteration. Apply step 1 of IBA and step 2 of GWO in a loop until all the loops have completed iteration.

Figure 2 shows the combination process of the hybrid algorithm.

When the IBA becomes stuck in a local optimum, the GWO can help it escape the local optimum, and when the GWO gets stuck, the IBA will push it towards the global optimum. This algorithm takes the optimal value between the two as the result to help a single algorithm break out of the local infinite loop in the loop process. HWIBO can further improve the global and local search ability to enhance the accuracy. The two complement each other to achieve overall balance.

3.2. Serial Hybridization

First, initialize the population parameters of GWO, and the initial grey wolf position is obtained from formula (5). Then, the results are brought into step 1–step 3 of the GWO to generate preliminary solutions. Use the preliminary solutions as the initial values for IBA and input them into formula (8) of IBA. Start the iterative process of step 1–step 4 of IBA. Finally, when all loops have been iterated, the optimal value and corresponding position of IBA are output. Otherwise, return to step 1 of GWO and continue the iteration.

The simulation results show that under the same conditions, the parallel mixing method is preferable. Therefore, the serial hybrid algorithm will not be introduced in detail here.

4. ANALYSIS OF SIMULATION RESULTS

In this paper, we simulate an array with a symmetric linear structure consisting of an odd number of elements. Because of the symmetry of the array, we only need to show the data for half of the X-axis. Due to the different constraints in previous literature, this paper conducts a comparative analysis from the following two different constraints. In case A, the array aperture size is specified, and the minimum spacing between the elements is limited, that is $\Delta x \geq 0.5\lambda$. In case B, the array element spacing is limited within $0.5\lambda \leq d \leq \lambda$, and the array aperture size is also limited. We verify the effectiveness and robustness of the proposed method by optimizing the PSLL of the array antenna under two distinct conditions.

4.1. Symmetric Linear Array Synthesis under Array Aperture and Minimum Element Spacing Constraints

Considering the characteristics of the symmetrical array, the number of elements distributed on the X-half axis is 8 for an array of 17-elements and 18 for an array of 37-elements. The aperture sizes of the arrays are 9.744λ and 21.996λ, respectively. The range of the array element spacing is $\Delta x \geq 0.5\lambda$. The HWIBO was applied to synthesize arrays of 17 and 37 elements for case A, and Figures 3(a) and 3(b) show the comparison between the results of HWIBO and [8].

From Figure 3(a), we can see that among the 17 elements, the PSLL of HWIBO decreases to -19.899 dB. In Figure 3(b), among the 37 elements, HWIBO can reach 21.090 dB. In both cases, the optimal PSLL obtained by our proposed HWIBO is lower than [8].

In Table 1(a), under the same parameter configurations, the PSLL obtained by HWIBO is compared with the data in [8, 2, 15]. The boundary arrays are fixed at $x_n = 4.872\lambda$ and 10.998λ, with units dB.

To validate the effectiveness of HWIBO, we compare its convergence with the BA and GWO under 17-elements, as illustrated in Figure 4(a).

Figure 4(b) shows the variation of the independent and random simulated 17-elements best and worst convergence curves for HWIBO.

4.2. Symmetric Linear Array Synthesis under Array Aperture and Minimum and Maximum Element Spacing Constraints

The size of the array aperture is set to 9.744λ for 17-elements and 21.996λ for 37-elements under the condition that the element spacing is limited to $0.5\lambda \leq d \leq \lambda$. The HWIBO is used to optimize the array synthesis of 17 and 37 elements. The obtained optimal array model is shown in Figure 5(a).

TABLE 1. (a) Comparison of array PSLL for different algorithms in case A, (b) the element spacing for the 17-element and 37- element arrays obtained by HWIBO in case A and case B.

(a)

Method	HWIBO	MGA[8]	SADE[2]	TSA[15]
17-element	-19.899	-19.797	-19.898	-19.898
37-element	-21.090	-20.562	-20.942	-21.025

(b)

17-element			37-element					
n	Case A	Case B	n	Case A	Case B	n	Case A	Case B
1	0.5000	0.5000	1	0.5000	0.5000	10	0.5066	0.5393
2	0.5000	0.5000	2	0.5000	0.5000	11	0.6046	0.5404
3	0.5000	0.5000	3	0.5000	0.5000	12	0.5671	0.5846
4	0.5000	0.5000	4	0.5000	0.5000	13	0.6192	0.7362
5	0.5751	0.5744	5	0.5000	0.5000	14	0.8543	0.7711
6	0.7443	0.7449	6	0.5000	0.5000	15	0.6917	0.8225
7	0.7128	0.7123	7	0.5000	0.5000	16	1.3212	0.9921
8	0.8398	0.8404	8	0.5000	0.5006	17	0.7918	0.9940
			9	0.5066	0.5101	18	0.5349	0.5071

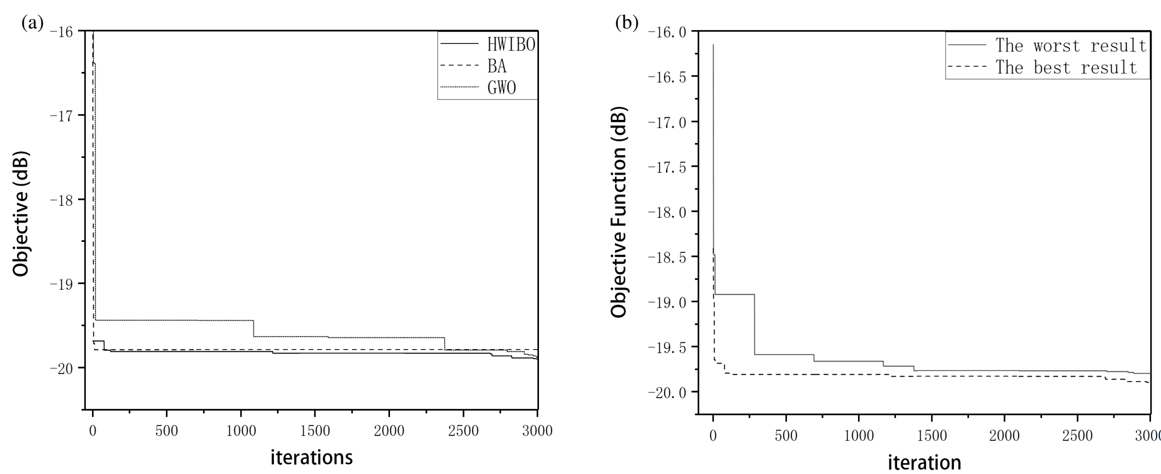

FIGURE 4. (a) Convergence curves comparisons of different algorithms, (b) convergence trend curve of the 17-element array.

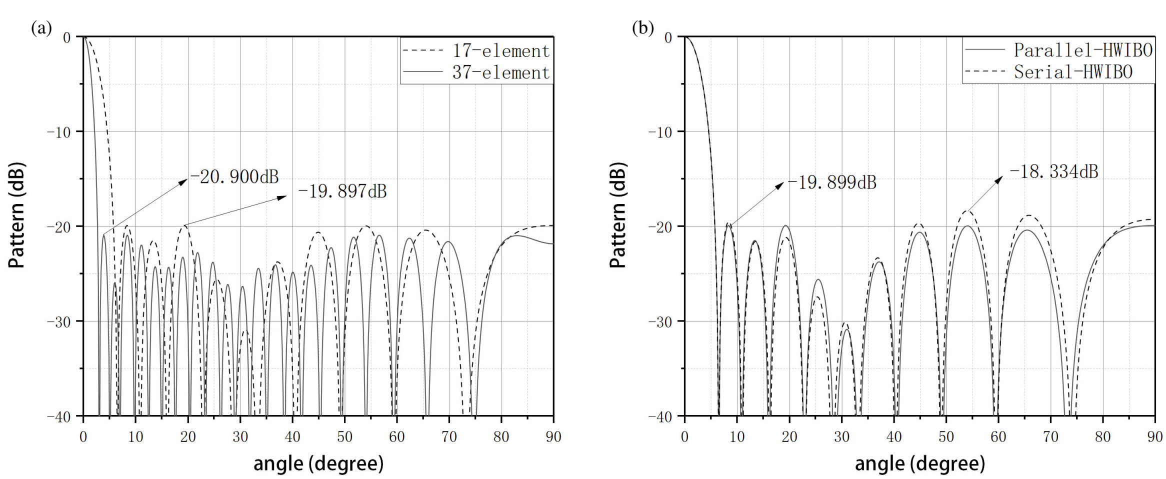

FIGURE 5. (a) Array synthesis diagram of 17-element and 37-element, (b) radiation patterns of a 17-element array under different hybrid configurations.

From Figure 5(a), we can see that among the 17-elements, the PSLL for the best solutions of HWIBO decreases to -19.899 dB, and among the 37-elements, the PSLL of HWIBO decreases to -20.900 dB.

The optimal element spacings obtained using HWIBO are shown in Table 1(b), Case B.

4.3. Comparison of Results between Serial and Parallel Hybrid Algorithms Considering the Limitations of Minimum Spacing and Array Aperture

Under the conditions of 4.1 invariance, the array synthesis of a 17-elements array is optimized using both serial hybrid algorithm and parallel hybrid algorithm. The array factors under the same conditions are shown in Figure 5(b).

From Figure 5(b), the PSLL of Parallel-HWIBO is reduced to -19.899 dB, whereas the Serial-HWIBO only reaches -18.334 dB. The results show that the parallel hybrid approach adopted in this paper is superior to the serial hybrid approach.

5. CONCLUSION

In this paper, we present a novel approach for nonuniform symmetric linear array synthesis based on HWIBO to effectively address the problem of optimizing element spacing in multi-constrained array synthesis. The proposed hybrid algorithm is more efficient in finding linear arrays with low PSLL. By optimizing the element positions with the hybrid algorithm, the better peak sidelobe levels are achieved. In future work, we will utilize the algorithm to optimize planar array antennas to validate its effectiveness in higher dimensions.

ACKNOWLEDGEMENT

This work was supported in part by the Natural Science Foundation of China under Grant 62071282, 61775126, in part by Open Fund of the Guangdong Provincial Key Laboratory of Optical Fiber Sensing and Communications (Jinan University).

REFERENCES

[1] Unz, H., "Linear arrays with arbitrarily distributed elements," *IRE Transactions on Antennas and Propagation*, Vol. 8, No. 2, 222–223, 1960.

[2] Zhang, F., W. Jia, and M. Yao, "Linear aperiodic array synthesis using differential evolution algorithm," *IEEE Antennas and Wireless Propagation Letters*, Vol. 12, 797–800, 2013.

[3] Skolnik, M., G. Nemhauser, and J. Sherman, "Dynamic programming applied to unequally spaced arrays," *IEEE Transactions on Antennas and Propagation*, Vol. 12, No. 1, 35–43, 1964.

[4] Rajo-Iglesias, E. and O. Quevedo-Teruel, "Linear array synthesis using an ant-colony-optimization-based algorithm," *IEEE Antennas and Propagation Magazine*, Vol. 49, No. 2, 70–79, 2007.

[5] Jin, N. and Y. Rahmat-Samii, "Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multiobjective implementations," *IEEE Transactions on Antennas and Propagation*, Vol. 55, No. 3, 556–567, 2007.

[6] Tian, Y.-B. and J. Qian, "Improve the performance of a linear array by changing the spaces among array elements in terms of genetic algorithm," *IEEE Transactions on Antennas and Propagation*, Vol. 53, No. 7, 2226–2230, 2005.

[7] Cen, L., Z. L. Yu, W. Ser, and W. Cen, "Linear aperiodic array synthesis using an improved genetic algorithm," *IEEE Transactions on Antennas and Propagation*, Vol. 60, No. 2, 895–902, 2012.

[8] Chen, K., Z. He, and C. Han, "A modified real GA for the sparse linear array synthesis with multiple constraints," *IEEE Transactions on Antennas and Propagation*, Vol. 54, No. 7, 2169–2173, 2006.

[9] Kurup, D. G., M. Himdi, and A. Rydberg, "Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm," *IEEE Transactions on Antennas and Propagation*, Vol. 51, No. 9, 2210–2217, 2003.

[10] Yang, X. S., "A new metaheuristic bat-inspired algorithm," *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*, 65–74, 2010.

[11] Perez, J., F. Valdez, O. Castillo, and O. Roeva, "Bat algorithm with parameter adaptation using interval type-2 fuzzy logic for benchmark mathematical functions," in *2016 IEEE 8th International Conference on Intelligent Systems (IS)*, 120–127, IEEE, 2016.

[12] Pan, Y. and J. Zhang, "Synthesis of linear symmetric antenna arrays using improved bat algorithm," *Microwave and Optical Technology Letters*, Vol. 62, No. 6, 2383–2389, 2020.

[13] Mirjalili, S., S. M. Mirjalili, and A. Lewis, "Grey wolf optimizer," *Advances in Engineering Software*, Vol. 69, 46–61, 2014.

[14] Sharma, S. K., N. Mittal, R. Salgotra, and U. Singh, "Linear antenna array synthesis using bat flower pollinator," in *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, 1–4, IEEE, 2017.

[15] Jiang, Y., S. Zhang, and Q. Guo, "An effective two-step approach to the synthesis of uniform amplitude linear arrays," *IEEE Antennas and Wireless Propagation Letters*, Vol. 16, 437–440, 2016.

[16] Li, X. and K. M. Luk, "The grey wolf optimizer and its applications in electromagnetics," *IEEE Transactions on Antennas and Propagation*, Vol. 68, No. 3, 2186–2197, 2019.

[17] Zhang, J., H. Zhang, X. Chen, F. Wu, Y. Liu, and W. Zhang, "A scalable arrangement method for aperiodic array antennas to reduce peak sidelobe level," *IEEE Access*, Vol. 11, 59 899–59 908, 2023.

[18] Chen, Y., S. Yang, Y. Suo, and M. Zheng, "Ship track prediction based on DLGWO-SVR," *Scientific Programming*, Vol. 2021, 1–14, 2021.