

Optimal Synthesis of Unequally Spaced Linear Arrays under Multiple Constraints

Zhong-Hui Zhao

College of Electronic and Information Engineering
Shandong University of Science and Technology, Qingdao, Shandong, 266590 China
zhaozh@sdu.edu.cn

Abstract – This paper proposes a differential evolution modified with adaptive ε -constraint handling and whale optimization algorithm (ε -WOA-DE) for the synthesis of unequally spaced linear arrays under array layout constraints and array pattern characteristics constraints. In particular, the success history based adaptive differential evolution with linear population reduction (LSHADE) serves as the basic search engine in this study. To ensure the searching ability of LSHADE under multiple constraints, an adaptive ε -constraint handling technique is implemented in LSHADE, in which the epsilon level is adjusted dynamically to make the solution scalable to the feasible region when it is in the infeasible region. In addition, the WOA mutation is implemented in the LSHADE to enhance the local search capability. Two array synthesis examples with multiple constraints are chosen to demonstrate the effectiveness of the proposed algorithm. The simulation results comparison and the convergence analysis of the ε -WOA-DE illustrate the superior capability of the proposed method.

Index Terms – adaptive ε -constraint handling, array synthesis, differential evolution, unequally spaced linear arrays, whale optimization algorithm.

I. INTRODUCTION

Synthesis of unequally spaced arrays has been widely explored in the last decades [1]. Compared with uniform spaced arrays, the element positions of the unequally spaced arrays can be exploited to achieve better pattern radiation characteristics. In particular, when array design is further combined with array amplitude and phase optimization, arrays with unequal spacing can achieve better array pattern performance [2].

Generally, the layouts of the unequally spaced arrays involve several constraints, such as element number, the total array length and the spacing between two adjacent elements [3–6]. Despite the array layout constraints, several array pattern characteristics, such as the maximum sidelobe level (SLL), the required mainlobe beamwidth

(BW) and the maximum null depth (ND) in some specified directions, are required in the array synthesis. These multiple constraints and requirements in the antenna array design lead the synthesis of unequally spaced arrays to complicated nonlinear constrained optimization problems, which increase the difficulty of the antenna synthesis.

Various evolution algorithms, such as the genetic algorithm, differential evolution (DE), seagull optimization algorithm, comprehensive learning particle swarm optimization (CLPSO) and whale optimization algorithm (WOA), have been concerned about solving this complicated nonlinear constraint problem [7–11]. For the geometry constraints, [7] has successfully transformed the geometry constraints to the unconstrained model by employing the vector mapping method and has been widely used in array synthesis. However, there is difficulty in satisfying the array pattern characteristic constraints. To deal with the array pattern characteristic constraints, [8–12] incorporate the constraints into the fitness function, by which the fitness values of the infeasible vectors are large and will be discarded in the optimization process. To satisfy the array pattern characteristic constraints, [12] proposed the modified DE with constrained vector projection (MDE-CVP) algorithm, the CVP method is used to exclude the infeasible solutions which unsatisfy the desired null depth. Although the lower null depth was realized, the SLL could not reach the desired value.

When constrained array synthesis evolutionary algorithms give priority to the satisfaction of constraints [13], it is likely to cause the following two problems. On the one hand, it is likely to make the population fall into a local infeasible region, so that the algorithm cannot find a feasible solution, which may result in failing to satisfy some constraints. On the other hand, it is likely to make the population converge to a locally feasible region, but far away from the location of the constraint in which the complete pareto optimal solution set in the target space cannot be found. Thus, in the constraint array synthesis, instead of excluding the infeasible solutions directly,

exploring infeasible regions around the feasible region is very effective in searching for the global optimal solutions in the constraint optimization [14].

In this study, in order to find better layouts of the antennas under multiple constraints, a modified DE algorithm with the adaptive ε -constraint control method is proposed to deal with the constraints, in which the epsilon level dynamically is adjusted to enhance the exploration of the infeasible regions in the optimization process. To ensure the convergence speed of the optimization, the adaptive ε -constraint control method is incorporated with the success-history based differential evolution with linear population reduction (LSHADE), which has shown superiority in the single objective optimization [15]. In addition, WOA mutation is introduced in the mutation process to enhance exploitation [16, 17]. The optimization of the array geometry is initialized to generate optimal radiation pattern under geometry constraints and the SLL, the BW and the ND constraints. Compared with other synthesis techniques [9, 10, 12], the proposed method performs well in the constrained array synthesis.

II. PROBLEM FORMULATION

As shown in Fig. 1, consider a $2N$ element linear array symmetrically placed along the x -axis with the aperture of $2L$, the array factor can be written as follows:

$$AF(\theta, X) = \sum_{n=1}^N \cos\left(\frac{2\pi}{\lambda} x_n \cos \theta\right), \quad (1)$$

where θ is the steering angle, λ is the wavelength, $X = [x_1, x_2, \dots, x_N]$ denote the element positions. Since the array aperture is $2L$, x_N is L . The SLL of the radiation pattern can be expressed as:

$$SLL(X) = \max_{\theta \in \text{Sidelobe}} \left| \frac{AF(\theta, X)}{AF(\theta_s, X)} \right|, \quad (2)$$

where *Sidelobe* is the sidelobe region corresponding to X , θ_s is the mainbeam direction. The maximum ND is denoted as:

$$ND(X) = \max_{m=1,2,\dots,M} \left| \frac{AF(\theta_m, X)}{AF(\theta_s, X)} \right|, \quad (3)$$

where $ND(X)$ is the maximum nulldepth corresponded with X , and θ_m , ($m = 1, 2, \dots, M$) are the specific directions of the steering nulls.

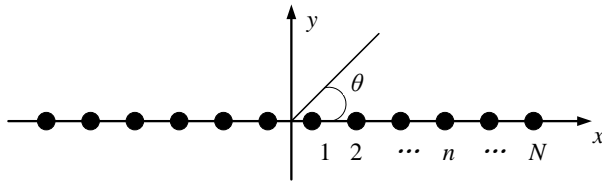


Fig. 1. Geometry of the $2N$ -element symmetric linear array.

In this study, we aim to design the layout of the linear array with the desired array aperture, the minimum element spacing constraints and limited SLL, BW and ND in some specific directions. Based on the vector mapping method in [7], the array layout constraints can be transformed in the element positions optimization:

$$\begin{cases} x_n = \sum_{j=1}^n \Delta x_j \\ \Delta x_n = \Delta d x_n + (L - (N - 0.5)d_c) \times a_n / \sum_{j=1}^N a_j \end{cases}, \quad (4)$$

where $\Delta x_j = x_j - x_{j-1}$ is the element spacing, $A = [a_1, a_2, \dots, a_N]$ is randomly generated among the range of $[0, 1]$, $\Delta d x = [0.5d_c, d_c, \dots, d_c]$ is a N -dimensional vector, d_c is the required minimum element spacing and $\Delta d x_n$ is the n th element of $\Delta d x$.

Inspired by the objective function implemented in [18], the new objective function, which not only aims to meet the BW, the SLL and the ND requirements, but also optimizes these array pattern characteristics is developed as follows:

$$\begin{aligned} \min_A & |BW(A) - BW_d| + SLL(A) + ND(A) \\ \text{s.t.} & SLL(A) \leq SLL_d \\ & ND(A) \leq ND_d \\ & |BW(A) - BW_d| \leq \tau BW_d \end{aligned}, \quad (5)$$

where $BW(A)$ is the beamwidth corresponded with A , BW_d is the desired beamwidth, SLL_d is the constraint SLL, ND_d is the desired maximum ND, and τ is the desired tolerance percentage of the beamwidth.

III. PROPOSAL OF THE ARRAY SYNTHESIZER

The array synthesis model in (5) is a nonconvex and highly nonlinear problem. With good global search capability, the success history based adaptive differential evolution with linear population reduction (LSHADE) is used in this study. Additionally, to handle the constraints and enhance the local search capability, the LSHADE is modified with the adaptive ε -constraint handling and WOA mutation, respectively.

A. Basic LSHADE

LSHADE is an improved DE algorithm, which adapts the parameters based on the success-history and employs the population size reduction (LPSR) mechanism [19]. The population of LSHADE is initialized as follows:

$$y_{i,j}^0 = lb_j + rand \cdot (ub_j - lb_j), \quad (6)$$

where *rand* represents a random number which distributed uniformly in $[0, 1]$. $y_{i,j}^0$ is the j th component ($j = 1, 2, \dots, D$) of the i th individual ($i = 1, 2, \dots, N$) in the initial population, ub_j and lb_j are the upper and lower bounds of the j th variable. And then, these individuals are evolved by the mutation, crossover and selection operators with the successful history based parameter adaption and linear population reduction.

a) Mutation: in this operator, the mutant vector v_i^G is created according to current-to-pbest/1 mutation strategy:

$$v_i^G = y_i^G + F_i^G (y_{pbest}^G - y_i^G) + F_i^G (y_{r1}^G - y_{r2}^G), \quad (7)$$

where y_i^G represents the i th target vector of the G th generation. y_{pbest}^G is randomly selected from the best $NP \times p$, ($p \in [0, 1]$) vectors of current population. y_{r1} and y_{r2} is randomly chosen from the union of the current population and the external archive. F_i^G is the scaling factor and is updated according to its historical successful experience.

b) Crossover: in this operator, the trial vector $u_i^G = [u_{i,1}^G, u_{i,2}^G, \dots, u_{i,D}^G]$ is generated according to the crossover rate, which can be expressed by:

$$u_{i,j}^G = \begin{cases} v_{i,j}^G & \text{if } (rand_{i,j} \leq Cr_i^G \text{ or } j = j_{rand}) \\ y_{i,j}^G & \text{otherwise} \end{cases}, \quad (8)$$

where j_{rand} is an integer randomly selected from $[1, D]$, Cr_i^G is the crossover rate and is updated according to its historical successful experience.

c) Selection: in this operator, not only the population is generated, but also the external archive is updated. After the selection of the vectors with better fitness function value, only NP_{G+1} best vectors will survive into the next generation, which is updated by linear population size reduction mechanism:

$$NP_{G+1} = \text{round} \left[\left(\frac{NP_{min} - NP_{ini}}{MFES} \right) \times FES + NP_{ini} \right], \quad (9)$$

where NP_{ini} is the initial population size, NP_{min} is the population size of the last generation, $MFES$ is the maximum number of fitness function evaluations, and FES is the current number of fitness function evaluations. The other $NP_G - NP_{G+1}$ vectors are removed to the external archive.

B. Adaptive ε -constraint handling

To ensure the searching ability under multiple constraints, this paper incorporates an adaptive ε -constraint handling technique in SHADE. In constraint problems optimization, the constraint violation is an important factor in measuring the constraints. In this array synthesis problem (5), the constraint violation is:

$$\begin{aligned} \varphi(A) = & \max(0, SLL(A) - SLL_d) \\ & + \max(0, ND(A) - ND_d) \\ & + \max(0, |BW(A) - BW_d| - \tau BW_d) \end{aligned}, \quad (10)$$

For two solutions A_1 and A_2 , their constraint violations are φ_1 and φ_2 . Then, for ε satisfies $\varepsilon \geq 0$, the ε -level comparison selects the better solutions as follows:

$$(A_1, \varphi_1) \preceq_{\varepsilon} (A_2, \varphi_2) \Leftrightarrow \begin{cases} A_1 \preceq A_2, & \text{if } \varphi_1, \varphi_2 \leq \varepsilon \\ A_1 \preceq A_2, & \text{if } \varphi_1 = \varphi_2 \\ \varphi_1 < \varphi_2, & \text{otherwise} \end{cases}. \quad (11)$$

Through the ε -level comparison, LSHADE algorithm can be used for constrained optimization directly [15]. Moreover, the ε -level comparison can extend the exploration of the infeasible regions around the feasible regions by setting appropriate ε value. Thus, to maintain the balance of searching between infeasible and feasible regions, an improved adaptive ε level control based on the exponential decline scheme is formulated in this study as:

$$\varepsilon(t) = \begin{cases} (1 - \frac{t}{T_c})^{cp} \varphi_{max} & \text{if } \varphi_{max} \leq Th \text{ or } r_t \leq ap_1 \\ & \text{and } t \leq T_c \\ ap \times \varphi_{max} & \text{if } \varphi_{max} > Th \text{ or } r_t > ap_1 \\ & \text{and } t \leq T_c \\ 0 & \text{if } t > T_c \end{cases}, \quad (12)$$

where φ_{max} is the maximum constraint violation in the current generation, cp controls the speed of declining constraints, r_t is the ratio of feasible vectors to total vectors in the t th generation, Th and ap_1 is to control the preference rule of setting ε value, ap is a small value.

Considering that in the exponential decline scheme, the ε may maintain a big value over a long period, which will degrade the search efficiency. When φ is larger than Th or there are enough feasible vectors, the ε value is set as a relatively small value to make the search focus on the feasible region and the infeasible region around the feasible region. In the final-stage, when the iteration number is larger than T_c , the ε value is set as 0 to enable the final exploitation in the feasible region.

C. WOA mutation

To enhance the exploitation around the best vectors, the spiral movement operator of WOA is incorporated into the mutation process[16]. The mutant vector v_i^G has a chance to make further updates using the spiral movement of WOA:

$$v_i^G = \begin{cases} w D_i e^{l} \cos(2\pi l) + y_{pbest}^G & \text{if } rand < 0.5 \\ v_i^G & \text{otherwise} \end{cases}, \quad (13)$$

where $w = \cos(\frac{0.5\pi G}{G_m})$ is the weight coefficient, $D_i = |y_{pbest}^G - v_i^G|$ and l is the uniform random number in the intervals $[-1, 1]$.

D. Synthesis procedure

The pseudo-code for the ε -WOA-DE is summarized in Algorithm 1.

Algorithm 1: ε -WOA-DE based array synthesis procedure

- 1: Set the number of elements N , the required array characters, the initial population size NP_{ini} , maximum generation number G_{max} .
- 2: Randomly generate NP_{ini} individuals
- 3: Calculate the fitness value by (5) and the constraint violation by (7)
- 4: Set $G = 1$
- 5: **while** $G < G_{max}$, **do**
- 6: **for** $i = 1 : NP_G$, **do**
- 7: Calculate F_i^G and CR_i^G
- 8: Make mutant vector v_i^G by (10) and (12)
- 9: Make trail vector u_i^G using (11)
- 10: Calculate the fitness function by (5) and the constraint violation by (7)
- 11: Select the next generation and update the external archive
- 12: **end for**
- 13: $G = G + 1$
- 14: **end while**
- 15: Output the best vector and the corresponding array element positions

IV. NUMERICAL RESULTS

To verify the effectiveness and efficiency of the ε -WOA-DE algorithm, two linear array synthesis examples are presented and compared with CLPSO [9], WOA [10] and MDE-CVP [12].

The initial population size NP_{ini} , the minimum population size NP_{min} and the maximum generation number G_{max} are set as 50, 10, 500, respectively. The other DE control parameters are the same as those in [19]. For the ε level handling, the decline speed cp is set as 2, ap is set as 0.2, the threshold parameters Th , Tc and ap_1 are set as 0.25, 150 and 0.2, respectively. For all examples, ten independent trails are performed and the best simulation results are evaluated.

The first example is a 28-element unequally spaced linear array synthesis. For comparison, we set $2L = 15.8 \lambda$, $d_c = 0.25 \lambda$, $\theta_s = 90^\circ$, $\theta_m = [120^\circ \ 122.5^\circ \ 125^\circ]$, $BW_d = 8.35^\circ$, $\tau = 0.05$, $SLL_d = -23 \text{ dB}$ and $ND_d = -90 \text{ dB}$, respectively. The corresponding optimal array geometry is shown in Table 1. Figure 2 compares the radiation pattern obtained by the ε -WOA-DE algorithm

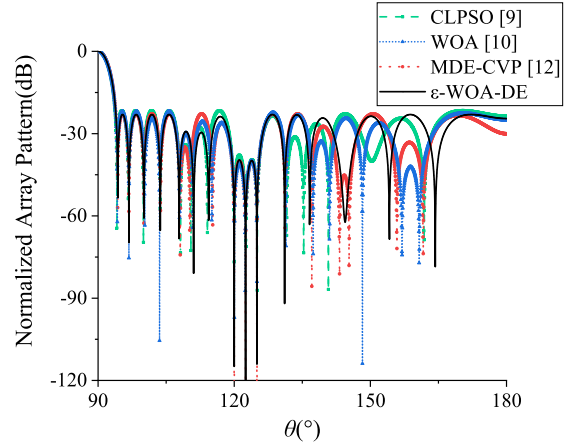


Fig. 2. Best radiation pattern of the 28-element array obtained by different algorithms.

and the other algorithm. The corresponding array pattern factors comparison of different algorithms are listed in Table 2. The ε -WOA-DE algorithm has achieved the lowest SLL and satisfied all the required array pattern characters in the two examples. The convergence plots of the objective function value, constraint violation, and the value of ε are shown in Fig. 3. The ε value is adjusted adaptively during the optimization process while the constraint violation declined gradually to 0. The algorithm converges in generations, respectively.

The second example is a 32-element linear array. The desired array factors are set as $2L = 16.8 \lambda$, $d_c = 0.25 \lambda$, $\theta_s = 90^\circ$, $\theta_m = 99^\circ$, $BW_d = 8.3^\circ$, $SLL_d = -23.5 \text{ dB}$ and $ND_d = -110 \text{ dB}$, respectively. The optimal array geometry is shown in Table 1. The performance comparisons are presented in Fig. 4 and Table 3. The ε -WOA-DE algorithm has met all the required

Table 2: Comparison of the MDE-CVP algorithm with other algorithms for the 28-element array

Algorithm	SLL, dB	ND, dB	BW, deg
CLPSO [9]	-21.60	-60	8.35
WOA [10]	-21.86	-106.27	8.49
MDE-CVP [12]	-22.80	≤ -150	8.6
ε -WOA-DE	-23.03	-114.02	8.6

Table 1: Optimal geometries of the antenna arrays obtained by the the ε -WOA-DE algorithm

Example	$[x_1/\lambda, x_2/\lambda, \dots, x_N/\lambda]$
Example 1	[0.3380 0.6104 1.3092 1.8184 2.1648 2.8049 3.1209 3.4471 4.1020 4.9288 5.8134 6.5189 7.1720 7.9]
Example 2	[0.2042 0.6589 1.0253 1.3998 1.8792 2.2190 2.7378 3.0639 3.5908 4.0176 4.5907 5.0038 5.7295 6.6059 7.5015 8.4]

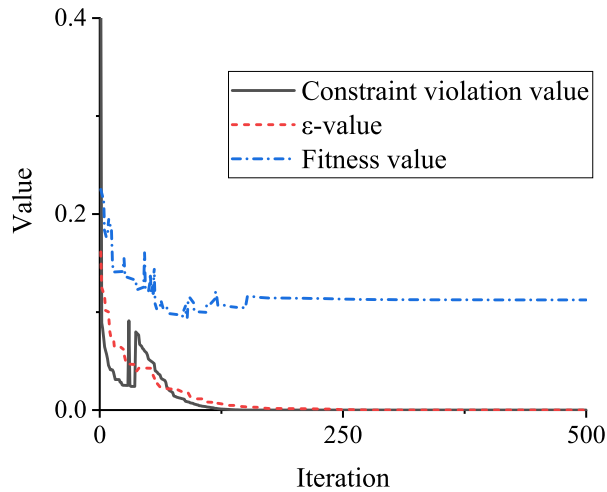


Fig. 3. Convergence curve plots of the 28-element array.

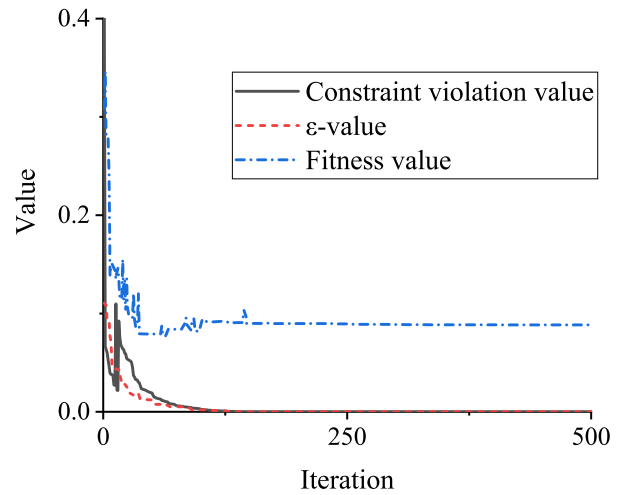


Fig. 5. Convergence curve plots of the 32-element array.

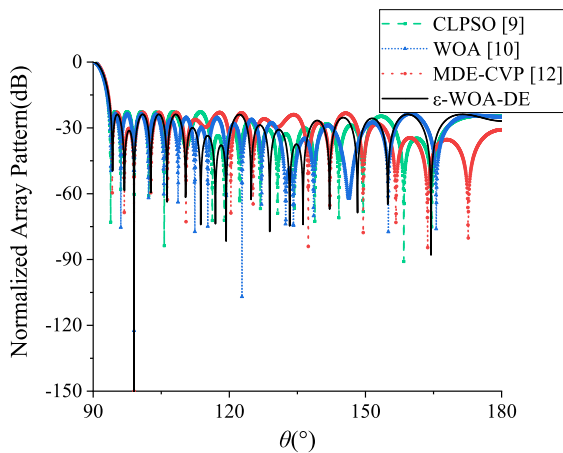


Fig. 4. Best radiation pattern of the 32-element array obtained by different algorithms.

Table 3: Comparison of the MDE-CVP algorithm with other algorithms for the 32-element array

Algorithm	SLL, dB	ND, dB	BW, deg
CLPSO [9]	-22.73	-60.45	8.35
WOA [10]	-23.62	-122.41	7.86
MDE-CVP [12]	-22.98	≤ -150	8.4
ϵ -WOA-DE	-23.83	-151.17	8.5

array pattern characters with lower ND and lowest SLL. The convergence curves of the objective function, constraint violation, and the ϵ value can be seen in Fig. 5.

In order to investigate the convergence performance and computational costs of the proposed ϵ -WOA-DE, Table 4 compares the required number of fitness function evaluations for convergence (NEC) in all examples. In

Table 4: Convergence analysis of different methods

Algorithm	NECs	
	Example1	Example2
CLPSO [9]	47560	27360
WOA [10]	21460	7440
ϵ -WOA-DE	7711	6220

addition, the algorithm converges at around 24000 NECs for all the examples in MDE-CVP [12]. In comparison, the ϵ -WOA-DE algorithm not only has powerful search capability but also performs quick convergence rate.

V. CONCLUSION

In this study, in order to optimize the positions of the unequally spaced array under multiple constraints, we propose a modified DE algorithm. The algorithm is based on the LSHADE and modified by implementing the adaptive constraint handling technique and integrating the spiral movement of the DE mutation process. Simulation results show that the proposed ϵ -WOA-DE algorithm has an improved performance in the array pattern characteristics control and the efficient computation time. Although ϵ -WOA-DE is only used for linear array synthesis in this study, it is worth noting that the constraint handling implemented technique is suitable for the other geometry array synthesis such as the circular and planar array. In future, in order to more accurately simulate and optimize the performance of unequally spaced arrays, we will investigate the incorporation of mutual coupling into the synthesis of unequally spaced arrays.

ACKNOWLEDGMENT

This work is supported by the Shandong Province Science Foundation for Youths: ZR202111100132.

REFERENCES

- [1] P. Rocca, G. Oliveri, R. J. Mailloux, and A. Massa, "Unconventional phased array architectures and design methodologies—a review," *Proceedings of the IEEE*, vol. 104, no. 3, pp. 544-560, 2016.
- [2] J. R. Mohammed, "Element selection for optimized multiwide nulls in almost uniformly excited arrays," *IEEE Antennas and Wireless Propagation Letters*, vol. 17, no. 4, pp. 629-632, 2018.
- [3] K. Chen, 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, pp. 2169-2173, 2006.
- [4] Y. Pan and J. Zhang, "Synthesis of linear symmetric antenna arrays using improved bat algorithm," *Microwave and Optical Technology Letters*, vol. 62, no. 6, pp. 2383-2389, 2020.
- [5] J. R. Mohammed and K. M. Younus, "Null steering implementation by controlling side elements positions," *International Journal of Microwave and Optical Technology*, vol. 16, no. 6, pp. 568-575, 2021.
- [6] J. R. Mohammed, "Obtaining wide steered nulls in linear array patterns by optimizing the locations of two edge elements," *AEU International Journal of Electronics and Communications*, vol. 101, pp. 145-151, 2019.
- [7] S. K. Goudos, K. Siakavara, T. Samaras, E. E. Vafiadis, and J. N. Sahalos, "Sparse linear array synthesis with multiple constraints using differential evolution with strategy adaptation," *IEEE Antennas and Wireless Propagation Letters*, vol. 10, pp. 670-673, 2011.
- [8] E. Kurt, S. Basbug, and K. Guney, "Linear antenna array synthesis by modified seagull optimization algorithm," *The Applied Computational Electromagnetics Society Journal (ACES)*, vol. 36, no. 12, pp. 1552-1562, 2022.
- [9] S. K. Goudos, V. Moysiadou, T. Samaras, K. Siakavara, and J. N. Sahalos, "Application of a comprehensive learning particle swarm optimizer to unequally spaced linear array synthesis with sidelobe level suppression and null control," *IEEE Antennas and Wireless Propagation Letters*, vol. 9, pp. 125-129, 2010.
- [10] C. Zhang, X. Fu, L. P. Ligthart, S. Peng, and M. Xie, "Synthesis of broadside linear aperiodic arrays with sidelobe suppression and null steering using whale optimization algorithm," *IEEE Antennas and Wireless Propagation Letters*, vol. 17, no. 2, pp. 347-350, 2018.
- [11] G. X. Liu, Q. Qin, and Q. H. Zhang, "Linear array synthesis for wireless power transmission based on brain storm optimization algorithm," *International Journal of Antennas and Propagation*, vol. 2021, pp. 1-8, 2021.
- [12] R. Q. Wang, Y. C. Jiao, H. Zhang, and Z. Zhou, "Synthesis of unequally spaced linear arrays using modified differential evolution algorithm," *IET Microwaves, Antennas & Propagation*, vol. 12, no. 12, pp. 1908-1912, 2018.
- [13] T. Takahama and S. Sakai, "Constrained optimization by the ϵ constrained differential evolution with gradient-based mutation and feasible elites," in *2006 IEEE International Conference on Evolutionary Computation*, pp. 1-8, 2006.
- [14] C. Zhang, A. K. Qin, W. Shen, L. Gao, K. C. Tan, and X. Li, " ϵ -Constrained differential evolution using an adaptive ϵ -level control method," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1-17, 2020.
- [15] R. Tanabe and A. S. Fukunaga, "Improving the search performance of SHADE using linear population size reduction," in *2014 IEEE Congress on Evolutionary Computation (CEC)*, pp. 1658-1665, 2014.
- [16] S. Mirjalili and A. Lewis, "The whale optimization algorithm," *Advances in Engineering Software*, vol. 95, pp. 51-67, 2016.
- [17] D. Prabhakar, K. Srinivas, R. Spandana, D. Anusha, M. Srikanth, and Y. R. Krishna, "The synthesis of elliptical antenna array using hybrid SSWOA algorithm," *Applied Computational Electromagnetics Society Journal*, vol. 38, no. 5, p. 309, 2023.
- [18] Q. Xu, S. Zeng, F. Zhao, R. Jiao, and C. Li, "On formulating and designing antenna arrays by evolutionary algorithms," *IEEE Transactions on Antennas and Propagation*, vol. 69, no. 2, pp. 1118-1129, 2021.
- [19] S. M. Islam, S. Das, S. Ghosh, S. Roy, and P. N. Suganthan, "An adaptive differential evolution algorithm with novel mutation and crossover strategies for global numerical optimization," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 482-500, 2012.



Zhonghui Zhao received M.S. degree in electromagnetic field and microwave technology and Ph.D. degree in electrical engineering from Northwestern Polytechnical University, Xi'an, Shaanxi in 2016 and 2020, respectively. She is currently employed as a lecturer in College of Electronic and Information Engineering, Shandong University of Science and Technology. Her current research interests include antenna arrays, evolutionary algorithms and machine learning.