Optimization of Sparse Linear Arrays Using Harmony Search Algorithms

Song-Han Yang and Jean-Fu Kiang

***Abstract*—A sparse linear array, composed of a uniformly spaced core subarray and an extended sparse subarray, is synthesized using a harmony search (HS) and an exploratory harmony search (EHS) algorithms. The optimal solution is searched by changing the amplitudes of all the elements and the positions of the extended elements, under a set of practical constraints. Performance of the EHS, the HS, a genetic algorithm (GA) and a particle swarm optimization (PSO) algorithm, has also been compared in synthesizing these sparse linear arrays.**

***Index Terms*—Antenna radiation patterns, evolutionary computation, phased arrays.**

I. Introduction

SYNTHESIS of a linear antenna array by tuning the amplitudes, phases, and/or positions of its elements can be formulated as a nonconvex optimization problem. Different types of evolutionary algorithms (EAs) have been successfully applied to achieve one design goal or another. In general, the performance of a specific EA is related to the array it is to be applied. An EA works better than the others for certain types of electromagnetic problems. Sometimes, just changing the number of control parameters in the same problem may favor different algorithms. In many applications, the arrays are required to be able to generate a specified beamwidth while maintaining an allowable peak sidelobe level (PSL). More complicated tasks like beamforming or pattern shaping have also been accomplished.

For example, Murino *et al.* applied a simulated annealing (SA) algorithm to reduce the sidelobe level by adjusting the position and amplitude of elements [1]. Chen *et al.* proposed a modified genetic algorithm (GA) to reduce the sidelobe level by adjusting the element positions [2]. Lommi *et al.* applied a GA to reduce the PSL of a linear sparse array [3]. Khodier *et al.* applied a particle swarm optimization (PSO) algorithm to adjust the spacings between elements, aiming to minimize the sidelobe level, as well as control the beamwidth and the null locations [4]. Bevelacqua *et al.* proposed a two-step PSO approach to achieve the minimum sidelobe level of an array [5]. The amplitudes are first optimized under given element positions, leading to a convex problem. Then, the element positions are optimized, with the amplitudes adjusted together, using a PSO algorithm. Guney *et al.* applied a harmony search (HS) algorithm to adjust the amplitude, phase, and position, respectively, of a linear array; in order to obtain single or multiple nulls at specific directions while maintaining low sidelobe level [6].

Rocca *et al.* presented a systematic review of differential evolution (DE) technique to synthesize array patterns [7]. Kurup *et al.* shaped the radiation pattern using a differential evolution strategy (DES) to determine the positions and phases of elements [8]. Chen *et al.* modified the conventional DES to improve its balance between convergence rate and exploration ability in searching for the global optimum [9]. The sum and difference patterns of various linear arrays have been optimized by varying element phases, positions, or both. Lin *et al.* applied several variants of DES to optimize linear arrays by adjusting element positions, phases, or both [10]. Goudos *et al.* applied a self-adaptive DE algorithm to optimize a linear array, among other electromagnetic problems [11]. It was reported that the comparison among different EAs and the choice of control parameters in a given algorithm are dependent on the problem to be solved.

Karimkashi and Kishk [12] applied an invasive weed optimization (IWO) algorithm to optimize the element positions of a planar array in [2] to achieve a lower sidelobe level with fewer elements. Quevedo-Teruel and Rajo-Iglesias applied an ant colony optimization (ACO) algorithm to control the sidelobe level of a thinned array [13]. Weng *et al.* used the Taguchi's method (TM) to control the null locations and to generate a sector-beam pattern [14]. Datta and Misra applied an adaptive bacteria-foraging algorithm (BFA) to steer the null locations and suppress the sidelobe level of an array [15]. Ho and Yang used the tabu search algorithm (TSA) to synthesize a nonuniformly spaced linear array, involving a tradeoff between the desired pattern and the PSL [16].

Oliveri *et al.* proposed a technique based on an almost difference set (ADS), which is a binary sequence bearing certain autocorrelation property, to remove a specific set of elements from an otherwise uniform planar array [17]. The PSL can be specified a priori. A similar technique has been proposed to design a very sparse array of a rectangular geometry [18]. These techniques can be applied to generate the optimal layout of elements efficiently out of a specific type of uniform array geometry.

Oliveri *et al.* proposed a Bayesian compressive sampling technique to design maximally sparse linear arrays, by casting the pattern matching problem into a probabilistic formulation [19], [20]. Tradeoff analysis on several key parameters have also been presented.

Manuscript received May 24, 2013; revised January 23, 2015; accepted August 30, 2015. Date of publication September 03, 2015; date of current version October 28, 2015. This work was supported in part by the Ministry of Science and Technology, Taiwan, ROC, under Contract NSC 101-2221-E-002-129 and in part by the Ministry of Education, Taiwan, ROC, under Aim for Top University Project 103R3401-1.

The authors are with the Department of Electrical Engineering and the Graduate Institute of Communication Engineering, National Taiwan University, Taipei 106, Taiwan (e-mail: jfkiang@ntu.edu.tw).

Digital Object Identifier 10.1109/TAP.2015.2476518

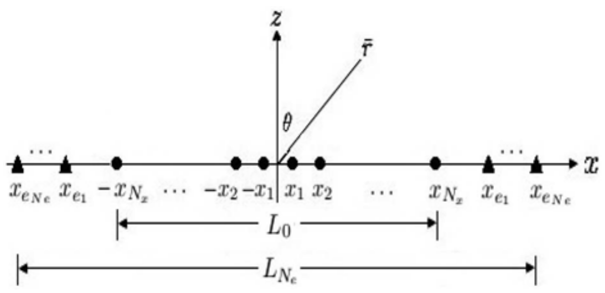

Fig. 1. Configuration of a linear subarray with uniform spacing (circle), extended by a linear spare subarray with at unequal spacings (triangle).

From the hardware perspective, phase shifters are needed to implement phase differences among elements of equal amplitudes, and variable-gain amplifiers are needed to implement amplitude differences among elements.

In this work, we choose the HS algorithm [21] to synthesize a linear array, which is composed of a uniform core subarray and an extended sparse subarray. Another exploratory harmony search (EHS) [22] is also chosen for comparison. The HS algorithm is briefly reviewed in Section II, and the EHS algorithm is a straightforward extension of the former. The linear arrays and the constraints upon them are described in Section III; simulation results of several optimized linear arrays are presented in Section IV; the comparison with GA and PSO algorithms in synthesizing these arrays is discussed in Section V; and some conclusion is drawn in Section VI.

II. Harmony Search Algorithms

The HS algorithm mimics the improvisation process of musicians, each playing a note, to find the best harmony for all [21]. The HS algorithm has been successfully applied to structure problems [23], communications [24], power systems [25], course scheduling [26], and array synthesis [6]; and it will be applied to synthesize sparse arrays in this work.

There are four major parameters in the HS algorithm, the harmony memory size (HMS), the harmony memory considering rate (HMCR), the pitch adjustment rate (PAR) and the fret width (FW), also called the adjusting bandwidth. The conventional HS algorithm used in this work is described below [21].

Initialization
for ($q = 1 : N_i$)
{
Improvisation starts
for ($n = 1 : N$)
{
if rand(1) < HMCR
$\bar{h}^{(q)}[n] = \bar{\bar{H}}^{(q)}[m^{(q)}, n]$
if rand(1) < PAR
$\bar{h}^{(q)}[n] = \bar{h}^{(q)}[n] + (-1 + 2 \times \text{rand}(1))\text{FW}$
else
$\bar{h}^{(q)}[n] = -X_{\min} + \text{rand}(1)\,(X_{\max} - X_{\min})$
}
} end of loop q

where rand(1) generates a random number between 0 and 1; N_i is the number of iterations; $m^{(q)}$ is an integer picked from $1, 2, \ldots, \text{HMS}$ with equal probability; $X_{\min}$ and $X_{\max}$ are the lower and upper bounds, respectively, of the note.

A. Initialization

The values of HMS, HMCR, PAR, and FW are assigned at the initialization stage, where HMS is the number of rows in the harmony memory (HM) matrix, having the explicit form of

$$\bar{\bar{H}} = \begin{bmatrix} \bar{h}_1 \\ \bar{h}_2 \\ \vdots \\ \bar{h}_M \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & \cdots & h_{1N} \\ h_{21} & h_{22} & \cdots & h_{2N} \\ \vdots & \vdots & \cdots & \vdots \\ h_{M1} & h_{M2} & \cdots & h_{MN} \end{bmatrix}$$

where M is equal to HMS and N is the number of musicians. Each row of $\bar{\bar{H}}$ represents a harmony, and each column records the notes played by a musician. In the first improvisation, take

$$\bar{\bar{H}}^{(1)} = X_{\min} + (X_{\max} - X_{\min})\,\text{rand}(M, N)$$

where rand(M, N) returns an $M \times N$ matrix of random numbers, all picked from the interval $[0, 1]$.

TABLE I
Parameters Used in Conventional HS and EHS Algorithms [22]

Case	HMS	N	HMCR	FWR	PAR	N_i	K
$L_{5,3}$(1O)	30	10	0.99	0.05	0.33	10^6	10^3
$L_{5,3}$(2O)	30	10	0.99	0.05	0.33	10^6	10^3
$L_{5,4}$(1O)	30	12	0.99	0.05	0.33	10^6	10^3
$L_{50,50}$(1O)	30	149	0.99	0.05	0.33	10^6	10^3

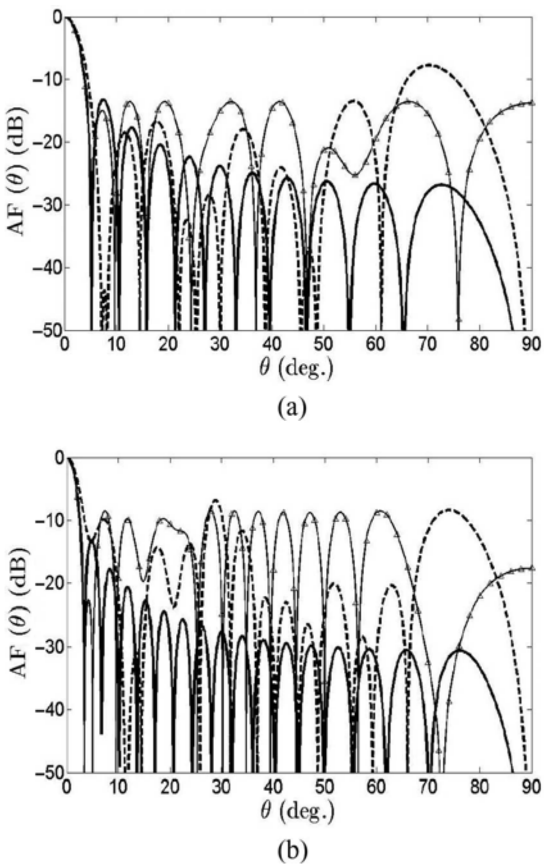

Fig. 2. Radiation patterns. (a) $-\triangle-$: $L_{5,3}$(1O),——: $L_{5,6}$(R), $- - -$: $L_{5,3}$(1S); $d_x = 0.5\lambda$, BW$_d = 10.40°$, $\kappa = 1$. (b) $-\triangle-$: $L_{5,3}$(2O), ——: $L_{5,12}$(R), $- - -$: $L_{5,3}$(2S); $d_x = 0.5\lambda$, BW$_d = 6.80°$, $\kappa = 1$.

TABLE II
BEAMWIDTH AND PSL OF LINEAR ARRAYS WITH $N_e = 3, 4$

	Core	$L_{5,3}$(1S)	$L_{5,6}$(R)	$L_{5,3}$(1O), $\kappa = 1$	$L_{5,3}$(1O), $\kappa = 1.2$	$L_{5,3}$(1O), $\kappa = 1.5$	$L_{5,3}$(1O), $\kappa = 2$
BW (°)	23.00	14.40	10.40	10.40	12.48	15.60	20.80
PSL (dB)	−12.96	−7.78	−13.20	−13.59	−14.57	−15.67	−27.76
	core	$L_{5,3}$(2S)	$L_{5,12}$(R)	$L_{5,3}$(2O), $\kappa = 1$	$L_{5,3}$(2O), $\kappa = 1.2$	$L_{5,3}$(2O), $\kappa = 1.5$	$L_{5,3}$(2O), $\kappa = 2$
BW (°)	23.00	9.40	6.80	6.80	8.16	10.20	13.60
PSL (dB)	−12.96	−6.84	−13.23	−8.39	−9.02	−11.60	−12.56
	core	$L_{5,4}$(1S)	$L_{5,8}$(R)	$L_{5,4}$(1O), $\kappa = 1$	$L_{5,4}$(1O), $\kappa = 1.2$	$L_{5,4}$(1O), $\kappa = 1.5$	$L_{5,4}$(1O), $\kappa = 2$
BW (°)	23.00	12.80	8.80	8.80	10.56	13.20	17.60
PSL (dB)	−12.96	−6.82	−13.21	−14.60	−15.69	−16.19	−29.17

TABLE III
POSITIONS AND AMPLITUDES OF ELEMENTS IN LINEAR ARRAYS WITH $N_e = 3$

Positions (λ) in $L_{5,3}$(1S)	± 0.25	± 0.75	± 1.25	± 1.75	± 2.25	± 3.25	± 4.25	± 5.25
Positions (λ) in $L_{5,3}$(1O)	± 0.25	± 0.75	± 1.25	± 1.75	± 2.25	± 3.66	± 4.36	± 5.25
Amplitudes in $L_{5,3}$(1S)	1	1	1	1	1	1	1	1
Amplitudes in $L_{5,3}$(1O)	1.06	0.77	1.05	0.91	1.36	1.73	1.65	1.20
positions (λ) in $L_{5,3}$(2S)	± 0.25	± 0.75	± 1.25	± 1.75	± 2.25	± 4.25	± 6.25	± 8.25
positions (λ) in $L_{5,3}$(2O)	± 0.25	± 0.75	± 1.25	± 1.75	± 2.25	± 4.43	± 6.73	± 8.25
Amplitudes in $L_{5,3}$(2S)	1	1	1	1	1	1	1	1
Amplitudes in $L_{5,3}$(2O)	0.25	0.53	1.27	1.07	1.08	0.91	1.90	1.79

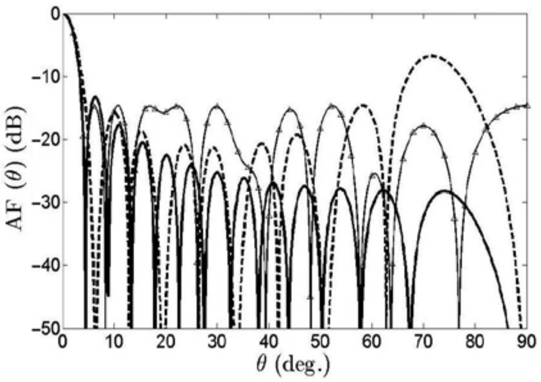

Fig. 3. Radiation patterns: $-\triangle-$: $L_{5,4}$(1O), ——: $L_{5,8}$(R), − − −: $L_{5,4}$(1S); $d_x = 0.5\lambda$, $\mathrm{BW}_d = 8.80°$, $\kappa = 1$.

B. Improvisation

In the qth iteration, a new harmony $\bar{h}^{(q)}$ is created via a recombination and a mutation process. The nth component of the new harmony is selected, with probability HMCR, from the set $\{h_{1n}, h_{2n}, \ldots, h_{Mn}\}$; or randomly selected, with probability 1− HMCR, from $[X_{\min}, X_{\max}]$. A pitch adjustment is applied to the nth component, with probability PAR, if it is selected from the existing harmonies.

The FW is derived as

$$\mathrm{FW} = \mathrm{FWR}(X_{\max} - X_{\min})$$

where FWR stands for FW ratio. An adaptive FW can also be adopted as [22]

$$\mathrm{FW}^{(q)} = \zeta\sqrt{\mathrm{var}\{\bar{\bar{H}}^{(q)}\}}$$

where $\mathrm{var}\{\bar{\bar{H}}^{(q)}\}$ is the variance over all the elements in $\bar{\bar{H}}^{(q)}$, and ζ is an empirical number, suggested to be 1.17.

The EHS algorithm is the same as the conventional HS algorithm, with the FW in each iteration updated according to the variance of the HM matrix $\bar{\bar{H}}$ during that iteration.

C. HM Update

During the improvisation process, each musician (decision variable) generates a note (value) to find the best harmony (global optimum). If a set of values, representing a specific geometrical configuration with specific amplitudes, do not satisfy the constraints, they will be discarded. If the new harmony is better than the worst harmony in matrix $\bar{\bar{H}}^{(q)}$, based on the fitness value (aesthetic evaluation), this new harmony will replace the worst harmony.

III. LINEAR ARRAY WITH EXTENDED SPARSE ELEMENTS

In a typical sparse array or thinned array, the elements are allowed to move farther away from adjacent elements to reduce the number of elements, while maintaining similar radiation pattern as that of a counterpart with uniform spacing [1], [2], [4], [8], [12], [13], [27]–[29]. In this work, we attach a sparse subarray to a core subarray with uniform spacing. The core subarray determines the key characteristics of the radiation pattern, while the sparse subarray is used to improve the radiation pattern using fewer elements than another conventional subarray of the same length.

Fig. 1 shows a linear subarray of $2N_x$ elements, at a uniform spacing of d_x; extended by a sparse subarray of $2N_e$ elements. All the elements are aligned along the x axis. The length of the core subarray is L_0, and the total length of the whole array is L_{N_e}.

In this work, we will focus on the comparison of different EAs in optimizing a sparse linear array to achieve a narrow beamwidth and low sidelobe levels. The main beam is steered toward the broadside direction without loss of generality. More complicated patterns as in [19] and [20] can also be achieved.

To further simplify the problem, the positions and amplitudes of the elements are required to be symmetrical about the origin. Hence, the array factor can be reduced to

$$\mathrm{AF}(\theta) = \sum_{n=1}^{N_x} a_n \cos(kx_n \cos\theta) + \sum_{m=1}^{N_e} a_{e_m} \cos(kx_{e_m} \cos\theta)$$

TABLE IV
POSITIONS AND AMPLITUDES OF ELEMENTS IN LINEAR ARRAYS WITH $N_e = 4$

Positions (λ) in $L_{5,4}$(1S)	± 0.25	± 0.75	± 1.25	± 1.75	± 2.25	± 3.25	± 4.25	± 5.25	± 6.25
Positions (λ) in $L_{5,4}$(1O)	± 0.25	± 0.75	± 1.25	± 1.75	± 2.25	± 3.73	± 4.49	± 5.29	± 6.25
Amplitudes in $L_{5,4}$(1S)	1	1	1	1	1	1	1	1	1
Amplitudes in $L_{5,4}$(1O)	0.60	0.60	1.04	1.12	0.92	1.39	1.22	1.26	1.05

TABLE V
BEAMWIDTH AND PSL OF LINEAR ARRAYS WITH $N_x = 50$ AND $N_e = 50$

	Core	$L_{50,50}$(1S)	$L_{50,100}$(R)	$L_{50,50}$(1O), $\kappa = 1$
BW (°)	2.30	1.10	0.80	0.80
PSL (dB)	−13.26	−6.78	−13.26	−18.05

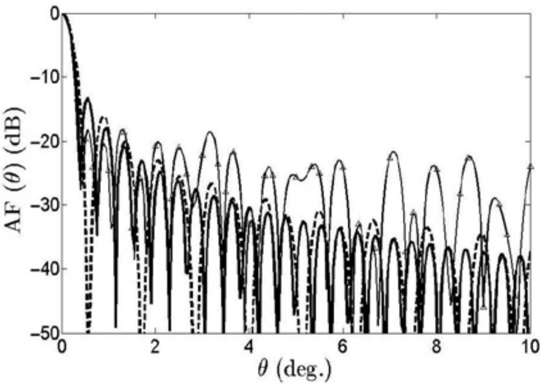

Fig. 4. Radiation patterns: $-\triangle-$: $L_{50,50}$(1O), ——: $L_{50,100}$(R), − − −: $L_{50,50}$(1S); $d_x = 0.5\lambda$, $\mathrm{BW}_d = 0.80°$, $\kappa = 1$.

where $x_n = (n - 1/2)d_x$ is the position of the nth core element; a_n is the amplitude of the nth core element, with $1 \leq n \leq N_x$; a_{e_m} is the amplitude of the mth extended element, with $1 \leq m \leq N_e$. The amplitudes of all the elements and the positions of all the extended elements are recorded in vector form as

$$\left[a_1, a_2, \ldots, a_{N_x}, a_{e_1}, a_{e_2}, \ldots, a_{e_{N_e}}\right]$$
$$\left[x_{e_1}, x_{e_2}, \ldots, x_{e_{N_e}}\right]$$

The amplitudes of all the elements are initially set to one, and limited to $(0, 2]$ all the time to provide more freedom for optimization yet are practical to realize. Tuning the positions of the extended elements will add more degrees of freedom to optimize the radiation pattern. A similar discussion on how to optimize the amplitudes and positions of elements in a linear array can be found in [5].

A few more practical constraints will be imposed on the positions of the extended elements: the spacing between two adjacent extended elements must be greater than $\lambda/2$ and smaller than 2λ; and the outermost extended elements must be located at $x = \pm L_{N_e}/2$ to make the array length equal to L_{N_e}.

IV. RESULTS OF OPTIMIZATION

Firstly, consider a linear array with $N_x = 5$, $d_x = 0.5\lambda$, and $N_e = 3$. The initial amplitudes are chosen as $a_1 = a_2 = \cdots = a_5 = a_{e_1} = a_{e_2} = a_{e_3} = 1$, and the initial positions of the extended elements are chosen as $[x_{e_1}, x_{e_2}, x_{e_3}] = [3.25, 4.25, 5.25]\,\lambda$. For the convenience of discussion, this initial setting is labeled as $L_{5,3}$(1S), where the number 1 in the parenthesis indicates the original spacing of the extended elements. For comparison, another linear array, labeled as $L_{5,6}$(R), is simulated, in which the extended elements are located at the spacing of 0.5λ to reach the total length of $L_{N_e} = 10.5\lambda$, and the amplitudes of all the elements are equal to one.

We will try to minimize the PSL of $L_{5,3}$(1S), while maintaining the same beamwidth as that of $L_{5,6}$(R), by optimizing the positions of its extended elements and the amplitudes of all the elements. The optimized array is labeled as $L_{5,3}$(1O).

The fitness function for optimization is chosen as

$$f\left(\bar{w}, \bar{\psi}\right) = P(\Omega \left|\bar{w}, \bar{\psi}\right) + K\left|\mathrm{BW}(\bar{w}, \bar{\psi}) - \kappa \times \mathrm{BW}_d\right| \quad (1)$$

where $\bar{w} = \left[a_1, a_2, \ldots, a_{N_x}, a_{e_1}, a_{e_2}, \ldots, a_{e_{N_e}}\right]$ contains the amplitudes of all the elements; $\bar{\psi} = \left[x_{e_1}, x_{e_2}, \ldots, x_{e_{N_e - 1}}\right]$ contains the positions of the extended elements except the outermost one, which is fixed to $L_{N_e}/2$; and K is an empirical weighting factor. The first term on the right-hand side of (1) is used to minimize the PSL, which appears at the angular direction Ω. The second term on the right-hand side of (1) is used to achieve the desired beamwidth $\kappa \times \mathrm{BW}_d$ (in degrees), where BW_d is the beamwidth of $L_{5,6}$(R), which is measured from null to null.

Table I lists the parameters used in the HS and the EHS algorithms. More details can be found in [22]. We also observed that the optimum solution is not very sensitive to the parameters as long as they fall within a reasonable range. A tradeoff analysis in [19] shows that the object functions usually appear smooth near the optimal parameters, leading to a mild tolerance in picking these parameters.

Fig. 2 shows the field patterns of various linear arrays with $N_x = 5$ and $N_e = 3$, which are derived from the best solution in 10 trials of the EHS algorithm. The associated beamwidths and PSLs are listed in Table II.

The sparse array $L_{5,3}$(1S) has the beamwidth of $14.40°$ and the PSL of -7.78 dB. The optimized array, $L_{5,3}$(1O), under $\mathrm{BW}_d = 10.40$, and $\kappa = 1$, achieves a similar beamwidth as that of $L_{5,6}$(R); its sidelobe level is reduced to -13.59 dB, better than -13.20 dB of the latter.

Table III lists the element positions and amplitudes in these arrays. Compared with a shorter array consisting of the core elements only, $L_{5,3}$(1O) provides a narrower beamwidth and lower PSL. Compared with the uniformly spaced array $L_{5,6}$(R), $L_{5,3}$(1O) takes fewer elements. If the beamwidth requirement is relaxed by setting $\kappa = 2$, the PSL can be further reduced to -27.76 dB.

Next, the spacing between the extended elements is increased to 2λ, to check if more elements can be spared without degrading the radiation pattern too much. A linear array $L_{5,3}$(2S) is thus proposed, with its performance listed in Table II and its pattern shown in Fig. 2(b). The tradeoff between beamwidth

TABLE VI
PERFORMANCE OF EHS, HS, PSO, AND GA ON $L_{5,3}$(1O)

Algorithm	Best fitness (dB)	Average fitness (dB)	Fitness SD (dB)	Average FI	SD of FI
EHS	-13.59	-13.47	0.07	76,796	18,076
HS	-13.53	-13.48	0.10	84,450	17,818
PSO	-13.61	-13.07	0.57	41,269	25,037
GA	-11.88	-11.32	0.67	57,347	37,063

FI: fitness iteration, SD: standard deviation.

and PSL is similar to the previous case. However, reducing the PSL of $L_{5,3}$(2O) becomes more difficult, only -8.39 dB with $\kappa = 1$ and -12.56 dB with $\kappa = 2$, as listed in Table II. Obviously, a wider initial spacing between extended elements raises the PSL. The optimal positions and amplitudes in $L_{5,3}$(2O) are also listed in Table III.

A longer sparse array $L_{5,4}$(1S) can be derived by adding one more extended element to either side of $L_{5,3}$(1S). The performance of $L_{5,4}$(1S) and $L_{5,4}$(1O) is listed in Table II, and their patterns are shown in Fig. 3. The PSL is suppressed to -14.60 dB with $\kappa = 1$ and -29.17 dB with $\kappa = 2$. The positions and amplitudes of the extended elements in $L_{5,4}$(1O) are listed in Table IV.

Finally, consider large linear arrays with $N_x = 50$, $d_x = 0.5\lambda$, and $N_e = 50$. In $L_{50,50}$(1S), the amplitudes are $a_1 = a_2 = \cdots = a_{50} = a_{e_1} = a_{e_2} = \cdots = a_{e_{50}} = 1$, and the spacing between extended elements is 1λ. The $L_{50,50}$(1O) array is derived from $L_{50,50}$(1S) by optimizing the amplitudes of all the elements and the positions of the extended elements. In $L_{50,100}$(R), more extended elements are located at a uniform spacing of 0.5λ, making its total length equal to that of $L_{50,50}$(1S).

Table V lists the performance of these three arrays plus a core array, which is consisted of 50 core elements only. These results are derived from the best solution in 10 trials using the EHS algorithm. Fig. 4 shows the radiation patterns of these three arrays. The PSL of $L_{50,50}$(1S) is -6.78 dB. As a comparison, $L_{50,50}$(1O) has the PSL of -18.05 dB, while its beamwidth is $0.80°$, the same as that of $L_{50,100}$(R).

V. COMPARISON OF EHS, HS, PSO, AND GA

In this section, performance of the EHS and the HS algorithms will be compared, in optimizing two sparse arrays $L_{5,3}$(1O) and $L_{50,50}$(1O), respectively. Two commonly used EAs, PSO and GA, will also be used for reference [30].

These algorithms are assessed for their convergence rate, in terms of the required number of fitness iterations (FI), and the success ratio of converging to the global optimum. The deviation among individual realizations is characterized by the standard deviation of FI, labeled as SD.

A general view of matching different EAs to different electromagnetic problems has been presented in [31]; in which a probability density function is propossed to assess the matching between algorithm and specific problem. The success ratio defined in our work plays a similar role. A higher success ratio implies less probability of being trapped in a local minimum, which is one of the motives to develop EAs.

TABLE VII
PERFORMANCE OF EHS, HS, PSO, AND GA ON $L_{5,3}$(1O)

Algorithm	Requirement on PSL	-11 (dB)	-12 (dB)	-13 (dB)
EHS	Success ratio (%)	100	100	100
	Average FI (SD)	1311 (1300)	3237 (2227)	9513 (3842)
HS	Success ratio (%)	100	100	100
	Average FI (SD)	2423 (2621)	3998 (1580)	14 597 (22 372)
PSO	Success ratio (%)	100	90	70
	Average FI (SD)	7639 (2629)	11 952 (5212)	13 216 (2806)
GA	Success ratio (%)	70	20	0
	Average FI (SD)	35 707 (34 406)	36 402 (39 259)	–

FI: fitness iteration, SD: standard deviation.

A. Particle Swarm Optimization

The PSO algorithm is inspired by food-searching swarms [32], [33]. Particles fly in their search space based on the instructions of habit, self knowledge, and social knowledge. In the subsequent discussions, we choose the population size of the swarm to be 30, and the empirical factors C_1 and C_2 in the updating equation to be 2. The habit weight w is decreased from 0.9 to 0.4 linearly with iterations. The invisible boundary condition (IBC) is adopted, and the maximum velocity is set to 20% that of the maximum coordinate in the search space.

B. Genetic Algorithm

The GA was motivated by Darwins theories of evolution, which is consisted of the processes of genetic selection, recombination (also named reproduction or crossover), mutation, and propagation (also named creation) [34], [35]. The version suggested in [36] is used in the subsequent discussions. The population size is assumed 30; the tournament selection is adopted; the crossover and the mutation probabilities are set to $p_r = 0.9$ and $p_m = 0.3$, respectively. The maximum numbers of recombined and mutated individuals are functions of p_r, p_m and the population size. The recombined and the mutated individuals are always involved in the propagation process. If the population size is not full, the best individual and some randomly chosen individuals from the old generation will be recruited.

C. Comparison on $L_{5,3}$(1O)

The computational time depends on the computers used for simulations and how the computer codes are architected. In general, the computational time is roughly in proportional to

TABLE VIII
PERFORMANCE OF EHS, PSO, AND GA ON $L_{50,50}(1O)$

Algorithm	Constraint	Success ratio (%)	Best fitness (dB)	Average fitness (dB)	Fitness SD (dB)	Average FI	SD of FI
EHS	$\kappa = 1$	100	-18.05	-17.79	0.13	104 207	46 807
	$\kappa = 1.2$	100	-19.70	-19.51	0.20	87 923	45 966
	$\kappa = 1.5$	100	-20.30	-20.01	0.10	44 750	27 350
PSO	$\kappa = 1$	0	–	–	–	–	–
	$\kappa = 1.2$	50	-20.54	-19.40	0.51	69 398	45 775
	$\kappa = 1.5$	80	-21.05	-11.70	5.58	27 054	10 996
GA	$\kappa = 1$	0	–	–	–	–	–
	$\kappa = 1.2$	0	–	–	–	–	–
	$\kappa = 1.5$	0	–	–	–	–	–

the number of iterations required, assuming that each iteration takes about the same amount of arithmetic operations. Different algorithms usually take different amounts of arithmetic operations in each iteration. In this work, we focus more on the convergence behavior and the exploration ability (success ratio) to reach the global optimum.

In any trial of EHS, HS, PSO, or GA, different number of FIs may be taken [14]. In order to reach a fair comparison, each algorithm is given 10 trials, with a maximum of 100 000 FIs in each trial. The number of FIs is recorded when the algorithm ceases to improve over the previous solution. Table VI lists the performance of these algorithms on $L_{5,3}(1O)$. Table VII lists the performance statistics of these algorithms, under different requirements on the PSL.

The EHS and the HS algorithms take more FIs to reach the optimal solution than the other two algorithms, but the global optimum can always be reached in 10 trials. On the other hand, the PSO and the GA algorithms may fail to find the solution, especially when the requirement on PSL is more stringent.

If we take a closer look at Table VI, the EHS and the HS algorithms seem to be more capable of avoiding the local optima than the other two algorithms, at the cost of taking more iterations. As listed in Table VII, the success ratios of the EHS and the HS algorithms are 100%, under three different PSL requirements. The PSO algorithm seems prone to stuck in local optimum, and its success ratio gets lower under more stringent requirement, e.g., 70% under -13 dB of PSL. The GA algorithm has poor success ratios of 20% and 0% under PSL of -12 and -13 dB, respectively.

As listed in Table VII, when the PSL is used as the stopping criterion of iteration, the EHS and the HS algorithms take fewer FIs than the PSO and the GA algorithms. When the PSL is set on -11 dB, the EHS algorithm takes about 17% iterations that of the PSO algorithm. Compared with the HS algorithm, the EHS algorithm takes fewer iterations due to the adaptive FW. When the PSL is set to -13 dB, the EHS algorithm takes about 65% iterations that of the HS algorithm.

D. Comparison on $L_{50,50}(1O)$

Table VIII lists the performance of EHS, PSO, and GA algorithms on a larger array of $L_{50,50}(1O)$. Each algorithm is given 10 trials, with a maximum of 200 000 FIs in each trial. A trial is claimed successful if the beamwidth satisfies the requirement.

The GA fails in all three cases. The PSO algorithm fails to fulfill the narrowest beamwidth constraint ($\kappa = 1$), and achieve 50% and 80% of success rates under the constraints of $\kappa = 1.2$ and $\kappa = 1.5$, respectively. The EHS algorithm reaches 100% of success ratio in all three cases. However, the best fitness of the EHS algorithm is worse than that of the PSO algorithm.

The factor κ can be viewed as a relaxation parameter to achieve the ideal beamwidth BW_d of a uniformly spaced array. The simulation results associated with $1 \leq \kappa \leq 2$ are presented in this work. It is observed by simulations that choosing $\kappa < 1$ is accompanied by high sidelobe levels. When $\kappa > 2$, the required beamwidth is very difficult to achieve. It appears that BW_d sets a reasonable range of beamwidth that is achievable.

E. Further Considerations

In practical implementation, especially when the hardware is involved, the allowable amplitudes can be predefined in a set of discrete levels. The step size between adjacent levels may affect the optimization results of amplitudes and radiation pattern, which are expected to converge to those obtained by assuming analog amplitudes as the step size is reduced to a very small number. The extended elements may also be allocated to a finite number of allowable positions, which will also affect the optimal choice of amplitudes.

One may also consider optimizing the amplitudes of the S arrays, which is equivalent to optimizing the fitness function in (1) over possible amplitudes $\bar{w}$, with fixed positions $\bar{\psi}$.

A specific optimization algorithm may work better than the others for one specific application, and vice versa for another application. The results presented in this work are used to validate the proposed method in optimizing sparse linear arrays specified in the context.

VI. CONCLUSION

A linear array composed of a uniform core subarray and an extended sparse subarray is optimized using the HS and the EHS algorithms. The amplitudes of all elements and the positions of the extended elements are optimized to generate a field pattern with the desired beamwidth and allowable PSL. Arrays of different sizes and sparsities have been simulated. Under the specific geometrical configuration and the assumption of analog amplitudes, the performance of synthesized arrays is compared with that using the GA and the PSO algorithm. The EHS and the Hs algorithms take fewer FIs than the PSO and the GA algorithms to reach the optimal solution, which satisfies the requirement on the PSL and the beamwidth.

ACKNOWLEDGMENTS

The authors would like to thank the reviewers for their valuable comments and references.

REFERENCES

[1] V. Murino, A. Trucco, and C. S. Regazzoni, "Synthesis of unequally spaced arrays by simulated annealing," *IEEE Trans. Signal Process.*, vol. 44, no. 1, pp. 119–123, Jan. 1996.
[2] K. Chen, X. Yun, Z. He, and C. Han, "Synthesis of sparse planar arrays using modified real genetic algorithm," *IEEE Trans. Antennas Propag.*, vol. 55, no. 4, pp. 1067–1073, Apr. 2007.
[3] A. Lommi, A. Massa, E. Storti, and A. Trucco, "Sidelobe reduction in sparse linear arrays by genetic algorithms," *Microw. Opt. Technol. Lett.*, vol. 32, no. 3, pp. 194–196, Feb. 5, 2002.
[4] M. M. Khodier and C. G. Christodoulou, "Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization," *IEEE Trans. Antennas Propag.*, vol. 53, no. 8, pp. 2674–2679, Aug. 2005.
[5] P. J. Bevelacqua and C. A. Balanis, "Minimum sidelobe levels for linear arrays," *IEEE Trans. Antennas Propag.*, vol. 55, no. 12, pp. 3442–3449, Dec. 2007.
[6] K. Guney and M. Onay, "Optimal synthesis of linear antenna arrays using a harmony search algorithm," *Expert Syst. Appl.*, vol. 38, pp. 15455–15462, 2011.
[7] P. Rocca, G. Oliveri, and A. Massa, "Differential evolution as applied to electromagnetics," *IEEE Antennas Propag. Mag.*, vol. 53, no. 1, pp. 38–49, Feb. 2011.
[8] D. G. Kurup, M. Himdi, and A. Rydberg, "Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm," *IEEE Trans. Antennas Propag.*, vol. 51, no. 9, pp. 2210–2217, Sep. 2003.
[9] Y. K. Chen, S. W. Yang, and Z. P. Nie, "The application of a modified differential evolution strategy to some array pattern synthesis problems," *IEEE Trans. Antennas Propag.*, vol. 56, no. 7, pp. 1919–1927, Jul. 2008.
[10] C. Lin, A. Y. Qing, and Q. Y. Feng, "Synthesis of unequally spaced antenna arrays by using differential evolution," *IEEE Trans. Antennas Propag.*, vol. 58, no. 8, pp. 2553–2561, Aug. 2010.
[11] S. K. Goudos, K. Siakavara, T. Samaras, E. E. Vafiadis, and J. N. Sahalos, "Self-adaptive differential evolution applied to real-valued antenna and microwave design problems," *IEEE Trans. Antennas Propag.*, vol. 59, no. 4, pp. 1286–1298, Apr. 2011.
[12] S. Karimkashi and A. A. Kishk, "Invasive weed optimization and its features in electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 58, no. 4, pp. 1269–1278, Apr. 2010.
[13] O. Quevedo-Teruel and E. Rajo-Iglesias, "Ant colony optimization in thinned array synthesis with minimum sidelobe level," *IEEE Antennas Wireless Propag. Lett.*, vol. 5, no. 1, pp. 349–352, Dec. 2006.
[14] W.-C. Weng, F. Yang, and A. Z. Elsherbeni, "Linear antenna array synthesis using Taguchi's method: A novel optimization technique in electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 723–730, Mar. 2007.
[15] T. Datta and I. S. Misra, "A comparative study of optimization techniques in adaptive antenna array processing: The bacteria-foraging algorithm and particle-swarm optimization," *IEEE Antennas Propag. Mag.*, vol. 51, no. 6, pp. 69–81, Dec. 2009.
[16] S. Ho and S. Yang, "Multiobjective synthesis of antenna arrays using a vector tabu search algorithm," *IEEE Antennas Wireless Propag. Lett.*, vol. 8, pp. 947–950, Aug. 2009.
[17] G. Oliveri, L. Manica, and A. Massa, "ADS-based guidelines for thinned planar arrays," *IEEE Trans. Antennas Propag.*, vol. 58, no. 6, pp. 1935–1948, Jun. 2010.
[18] G. Oliveri, F. Caramanica, C. Fontanari, and A. Massa, "Rectangular thinned arrays based on McFarland difference sets," *IEEE Trans. Antennas Propag.*, vol. 59, no. 5, pp. 1546–1552, May 2011.
[19] G. Oliveri, M. Carlin, and A. Massa, "Complex-weight sparse linear array synthesis by Bayesian compressive sampling," *IEEE Trans. Antennas Propag.*, vol. 60, no. 5, pp. 2309–2326, May 2012.
[20] G. Oliveri and A. Massa, "Bayesian compressive sampling for pattern synthesis with maximally sparse non-uniform linear arrays," *IEEE Trans. Antennas Propag.*, vol. 59, no. 2, pp. 467–481, Feb. 2011.
[21] K. S. Lee and Z. W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: Harmony search theory and practice," *Comput. Methods Appl. Mech. Eng.*, vol. 194, no. 36–38, pp. 3902–3933, Sep. 2004.
[22] S. Das, A. Mukhopadhyay, A. Roy, A. Abraham, and B. K. Panigrahi, "Exploratory power of the harmony search algorithm: Analysis and improvements for global numerical optimization," *IEEE Trans. Syst. Man Cybern., Part B*, vol. 41, no. 1, pp. 89–106, Feb. 2011.
[23] M. Saka, "Optimum geometry design of geodesic domes using harmony search algorithm," *Adv. Struct. Eng.*, vol. 10, no. 6, pp. 595–606, Dec. 2007.
[24] R. Forsati, A. Haghighat, and M. Mahdavi, "Harmony search based algorithms for bandwidth-delay-constrained least-cost multicast routing," *Comput. Commun.*, vol. 31, no. 10, pp. 2505–2519, Jun. 2008.
[25] A. Verma, B. Panigrahi, and P. Bijwe, "Harmony search algorithm for transmission network expansion planning," *IET Gener. Transmiss. Distrib.*, vol. 4, no. 6, pp. 663–673, Jun. 2010.
[26] M. A. Al-Betar, A. T. Khader, and M. Zaman, "University course timetabling using a hybrid harmony search metaheuristic algorithm," *IEEE Trans. Syst. Man Cybern., Part C*, vol. 42, no. 5, pp. 664–681, Sep. 2012.
[27] 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 Wireless Propag. Lett.*, vol. 9, pp. 125–129, Jun. 2010.
[28] R. Bhattacharya, T. K. Bhattacharyya, and R. Garg, "Position mutated hierarchical particle swarm optimization and its application in synthesis of unequally spaced antenna arrays," *IEEE Trans. Antennas Propag.*, vol. 60, no. 7, pp. 3174–3181, Jul. 2012.
[29] N. Jin and Y. Rahmat-Samii, "Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multiobjective implementations," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 556–567, Mar. 2007.
[30] D. W. Boeringer and D. H. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," *IEEE Trans. Antennas Propag.*, vol. 52, no. 3, pp. 771–779, Mar. 2004.
[31] O. Quevedo-Teruel, E. Rajo-Iglesias, and A. Oropesa-Garcia, "Hybrid algorithms for electromagnetic problems and the no-free-lunch framework," *IEEE Trans. Antennas Propag.*, vol. 55, no. 3, pp. 742–749, Mar. 2007.
[32] J. Robinson and Y. Rahmat-Samii, "Particle swarm optimization in electromagnetics," *IEEE Trans. Antennas Propag.*, vol. 52, no. 2, pp. 397–407, Feb. 2004.
[33] S. Xu and Y. Rahmat-Samii, "Boundary conditions in particle swarm optimization revisited," *IEEE Trans. Antennas Propag.*, vol. 52, no. 3, pp. 760–765, Mar. 2007.
[34] D. Golberg, "Genetic and evolutionary algorithms come of age," *ACM Commun.*, vol. 37, pp. 113–119, 1994.
[35] J. M. Johnson and Y. Rahmat-Samii, "Genetic algorithms in engineering electromagnetics," *IEEE Antennas Propag. Mag.*, vol. 39, no. 4, pp. 7–25, Apr. 1997.
[36] S. Caorsi, A. Massa, M. Pastorino, M. Raffetto, and A. Randazzo, "Detection of buried inhomogeneous elliptic cylinders by a memetic algorithm," *IEEE Trans. Antennas Propag.*, vol. 51, no. 10, pp. 2878–2884, Oct. 2003.

Song-Han Yang was born in Tainan, Taiwan, on May 4, 1988. He received the B.S. degree in electrical engineering from the National Sun Yat-Sen University, Kaohsiung, Taiwan, in June 2011, and the M.S. degree in electrical engineering from the Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan, in July 2013.

Jean-Fu Kiang received the Ph.D. degree in electrical engineering from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 1989.

Since 1999, he has been a Professor of the Department of Electrical Engineering and the Graduate Institute of Communication Engineering, National Taiwan University, Taipei, Taiwan. His research interests include electromagnetic applications, including antennas and arrays, wave propagation in ionosphere and atmosphere, satellite navigation, remote sensing, and microwave systems.