

Sidelobe Reduction in Array-Pattern Synthesis Using Genetic Algorithm

Keen-Keong Yan and Yilong Lu, *Member, IEEE*

Abstract—A simple and flexible genetic algorithm (GA) for pattern synthesis of antenna array with arbitrary geometric configuration is presented. Unlike conventional GA using binary coding and binary crossover, this approach directly represents the array excitation weighting vectors as complex number chromosomes and uses decimal linear crossover without crossover site. Compared with conventional GA's, this approach has a few advantages: giving a clearer and simpler representation of the problem, simplifying chromosome construction, and totally avoiding binary encoding and decoding so as to simplify software programming and to reduce CPU time. This method also allows us to impose constraints on phases and magnitudes of complex excitation coefficients for preferable implementation in practice using digital phase shifters and digital attenuators. Successful applications show that the approach can be used as a general tool for pattern synthesis of arbitrary arrays.

Index Terms—Antenna arrays, genetic algorithms.

I. INTRODUCTION

IN array-pattern synthesis, the main concern is to find an appropriate weighting vector to yield the desired radiation pattern. Various analytical and numerical techniques have been developed to meet the challenge. Examples of analytical techniques include the well-known Taylor method and Chebyshev method [1]. In recent years, numerical approaches have become more popular as they are applicable not only to regular arrays (such as linear arrays and circular arrays) but also to arrays with complicated geometry layout and radiation pattern requirement. Examples of numerical techniques include the linear or nonlinear optimization methods [2], [3] and adaptive methods [4], [5]. However, all these methods are limited to relatively simple and ideal arrays without mutual coupling. The proposed genetic algorithm (GA) may be used as a simple and flexible alternative to achieve the same objectives which other methods can do and, more importantly, it has unique features to treat some complicated problems (such as arbitrary geometric layout, including mutual coupling) which cannot be done by other methods.

GA's [6] are search and optimization algorithms which have very wide applications. Recently, GA's have been applied to the field of array-pattern synthesis. Reference [7] uses a GA to optimize a thinned array and a planar array to produce patterns with the lowest sidelobe level, while [8] demonstrates its use in null steering in phased and adaptive array. Reference [9] applies a GA to determine the excitation coefficients for

a linear array connected with four-bit infinity digital phase shifters. All these applications use binary coding and binary genetic operation. Conventional GA's with binary coding and binary genetic operation are inconvenient and inefficient for array pattern synthesis problems to optimize real or complex numbers.

We propose here a simple and flexible GA for pattern synthesis of arbitrary arrays. This approach avoids coding and directly deals with real or complex weighting vectors. Using this approach, constraints on the phases and magnitudes of the complex coefficients can be easily imposed for practical implementation of digital phase shifters and digital attenuators.

II. THE GENETIC ALGORITHM

The flow of this approach is similar to that of a standard GA [10], [11]. But, unlike a conventional GA using binary coding and binary genetic operations, the proposed approach avoids encoding/decoding and uses decimal genetic operations treating directly real/complex array weighting vectors.

A. Construction of Chromosomes

Using GA's for array-pattern synthesis, radiation patterns correspond to living beings and array-weighting vectors correspond to chromosomes. GA's were invented to manipulate a string of binary coding. Conventional GA's encode the parameters in binary chromosomes and perform binary genetic operations. In this approach, chromosomes are represented directly by real/complex weighting vectors

$$\mathbf{C} = [c_1 c_2 \cdots c_n \cdots c_N], \quad c_n \in C_n \quad (1)$$

where c_n (known as a genetic material in a GA) represents the excitation of the n th radiator and C_n is the set or a subset of all complex numbers. In other words, C_n can be a set of some integers or real numbers or complex numbers or their mixed combination. N is the length of the weighting vector. This simple representation explicitly shows the relation between chromosomes in GA and array-excitation weighting vectors and, therefore, it is easier to understand and to implement in computer programs.

B. Initial Population

For fast convergence of GA iteration, the initial population can include approximate excitations by other simple techniques (such as minimization of mean square errors (MMSE) method [12], etc.) and excitations by guess based on experience and/or at random.

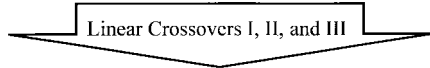
Manuscript received June 12, 1995; revised October 22, 1996.

The authors are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798.

Publisher Item Identifier S 0018-926X(97)04892-8.

Parent 1: $P1 = (0.45+0.20i, 0.30, 0.79)$

Parent 2: $P2 = (0.55+0.40i, -0.10, 0.79)$



Child 1: $C1 = (0.50+0.30i, 0.10, 0.79) \Leftarrow I. (P1+P2)/2$

Child 2: $C2 = (0.40+0.10i, 0.50, 0.79) \Leftarrow II. (3 \times P1 - P2)/2$

Child 3: $C3 = (0.60+0.50i, -0.30, 0.79) \Leftarrow III. (3 \times P2 - P1)/2$

Fig. 1. Example of the decimal linear crossover.

The MMSE method finds a weighting vector yielding a pattern which is as close as a reference pattern by minimizing the errors between the actual and reference response. The reference pattern is usually a rectangular pulse whose unity gain is located in the direction of the main beam of the desired pattern. Different sets of weights can be obtained by simply varying the pulse width of the reference pattern.

C. Reproduction

The reproduction processing consists of three basic genetic operations: *mating*, *crossover*, and *mutation*. There are many mating techniques available to pick two *parent* chromosomes to produce *child* chromosomes. Proportionate reproduction, ranking selection, tournament selection and genitor selection are some of the examples of the techniques used [10], [11]. In general, the more highly fit chromosomes should have higher chances to be selected and mated for producing children for the succeeding generation so we can use ranking selection to mate the couples.

Crossover is another process that involves exchange of genetic materials between two parent chromosomes to make child chromosomes. Unlike conventional binary crossover with one or more random crossover sites, we use more logical decimal linear crossover. For example, from two parent chromosomes U and V , three new child chromosomes can be produced from crossover I: $(U + V)/2$; crossover II: $(3U - V)/2$; and crossover III: $(3V - U)/2$. Crossover I gives the (average) "midpoint" of U and V , and crossover II and III give two extrapolation "points" from the midpoint. If the two corresponding genetic materials u_m and v_m in U and V , respectively, are identical, the corresponding materials in the three new child chromosomes will remain same as u_m or v_m . This is an important function for children to keep good features of their parent. Furthermore, we find that there is no need here to choose crossover site to perform partial crossover of genetic materials, adopted in conventional binary cross over. In this approach, we choose to perform crossover on all genetic materials and this will reduce the number of children so as to reduce the CPU time. Fig. 1 illustrates how the three child chromosomes $C1$, $C2$, and $C3$ are produced after the linear crossovers from two parent chromosomes $P1$ and $P2$.

Mutation plays a secondary role in GA. Mating and crossover effectively search and recombine extant useful genetic materials and occasionally they may become overzealous and lose some of these potential materials, so mutation is needed to protect against such an irrecoverable loss. Mutation is carried out by intentionally altering one or more genetic

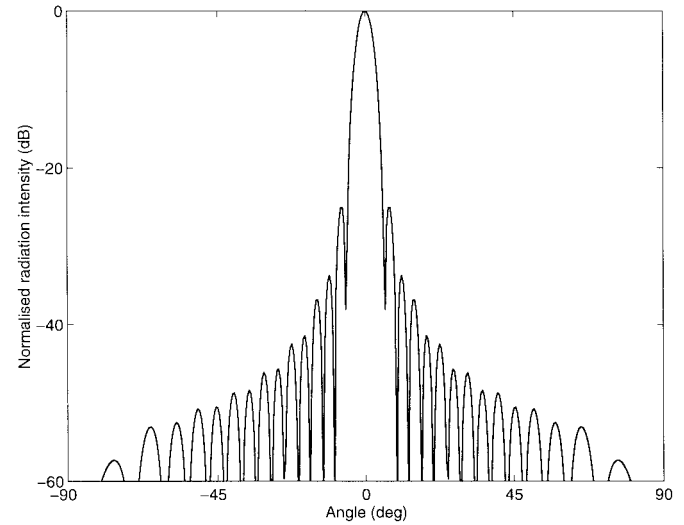


Fig. 2. Radiation pattern of the broadside linear array using the best set of initial coefficients by the MMSE method.

materials in a chromosome. In optimization, it corresponds to prevention of the algorithm of being stuck in a local minimum. Mutation is only carried out when a chromosome passes a probability check. When it does, one or a few randomly selected genetic materials of the chromosome are replaced with other randomly generated genetic materials.

D. Survival Selection

Once all the new child chromosomes are produced and their fitnesses corresponding to sidelobe levels are evaluated for selection of suitable chromosomes to compete for the next generation. The evaluation can be done by an ideal pattern function or by an accurate and realistic electromagnetic field computational simulator that can take into account the mutual coupling and environment effects.

There are a few selection techniques available. In this approach, both the child and parent populations are ranked together in the ascending order (for example, from the lowest sidelobe level to the highest sidelobe level). Then, based on the principle of survival of the fittest, those producing superior output survive, while those producing inferior output die off. Please note that the competitors for survival selection include both parents and their children so that the members of next generation may include members of the previous generation. This guarantees that the newer generation performs no worse than old ones. In other words, the error versus generation curve decreases monotonically.

E. Stopping Criteria

The progress of reproduction and survival selection continues until a satisfied result is obtained or preset maximum number of iteration is reached.

F. Comments on Convergence

We have found that it is much faster to achieve the desired result by having several shorter trials than a single longer

TABLE I
RELATIVE SIDELobe LEVELS (RSLl's) OF THE INITIAL POPULATION AND TEN GA RUNS FOR THE LINEAR ARRAYS

Broadside linear array		Linear array with $\theta_0 = 60^\circ$	
RSLl's (dB) with the initial 6 sets of coefficients by MMSE method	Lowest RSLl's (dB) for the 10 GA runs	RSLl's (dB) with the initial 6 sets of coefficients by MMSE method	Lowest RSLl's (dB) for the 10 GA runs
-13.25	-32.73	-12.25	-22.66
-14.22	-33.05	-12.79	-20.65
-16.83	-31.05	-13.15	-21.65
-19.08	-33.57	-13.63	-22.15
-22.03	-33.73	-14.27	-21.74
-25.07	-35.51	-15.14	-23.82
	-33.34		-21.84
	-32.52		-23.67
	-36.02		-22.09
	-32.67		-23.48

TABLE II
RELATIVE SIDELobe LEVELS (RSLl's) OF THE INITIAL POPULATION AND TEN GA RUNS FOR THE CIRCULAR ARRAYS

RSLl's (dB) with the initial 10 sets of coefficients by MMSE method	Lowest RSLl's (dB) for the 10 GA runs without phase constraint	Lowest RSLl's (dB) for the 10 GA runs with phase constraint to multiples of 11.25°
-13.87	-27.54	-24.88
-14.00	-27.09	-22.15
-14.26	-28.12	-23.81
-14.68	-26.83	-24.30
-15.21	-27.28	-23.48
-15.88	-26.32	-22.75
-16.83	-28.15	-22.58
-18.02	-26.28	-23.18
-19.50	-29.16	-22.37
-21.25	-26.98	-23.44

run. This is because when the progress sticks in somewhere (for example, a local minimum) it may take a long time to jump out. In general, the results from different runs are not identical due to different search routes caused by randomness of mutation. Therefore, performing another run increases the possibility of finding a better search route for a faster solution. In our implementation, if the satisfactory result is not achieved after a certain number of iterations, the program will restart the optimization. In our experience, 100 generations (iterations) are sufficient for each trial and more iteration will not help much to converge. Of course, the maximum number of trials is also required to stop the computing in case there is no solution.

III. EXAMPLES

To illustrate the effectiveness of the proposed approach, two examples are presented here. In the examples, the GA are carried out in ten runs of 100 iterations each. For these two examples, the CPU time for 100 iterations is about 120 s using MATLAB software on a 486/33MHz PC. All the initial weighting vectors of the following examples, except randomly generated ones, are produced by the MMSE method [12].

A. Linear Arrays

A broadside linear array of 30 isotropic elements, equally spaced at half wavelength, is used. The initial parent population consists of six sets of positive excitation amplitudes. Symmetric excitation is assumed, the crossover is done on half of the chromosome followed by copying it to the other half in the mirror image format.

As shown in the first column of Table I, the lowest sidelobe level achieved from the initial population is about -25 dB. After going through ten runs, the relative sidelobe levels obtained are between -31.05 and -36.02 dB. Comparing the solid line curves in Figs. 2 and 3, it can be seen that the sidelobes close to the main beam are lowered. At the same time, the leveled sidelobes in Fig. 3 indicates that the result is close to optimum solution for that particular beamwidth. This can be confirmed by comparing the solid line pattern by the GA and the dotted line pattern by Chebyshev method [1] in Fig. 3. Although Chebyshev method is able to generate perfectly leveled sidelobes, it is only applicable to uniform spaced linear arrays with isotropic elements. As for the GA, it is more flexible and versatile and it can be applied to arbitrarily spaced arrays.

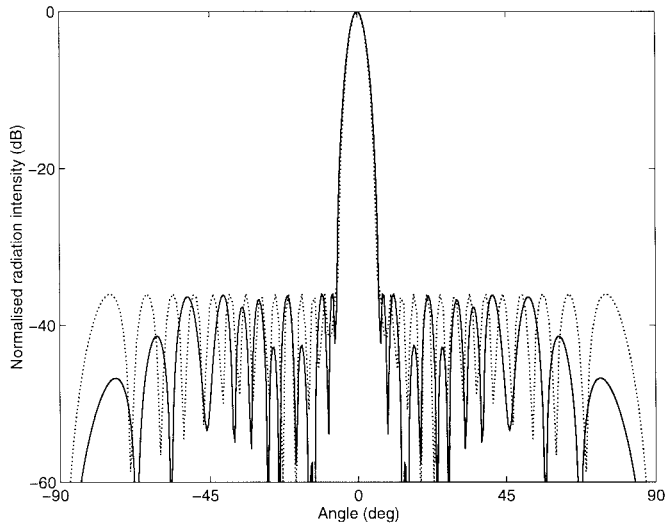


Fig. 3. Comparison of the radiation patterns of the broadside linear array with excitation coefficients by the GA (solid line) and by Chebyshev method (dotted line).

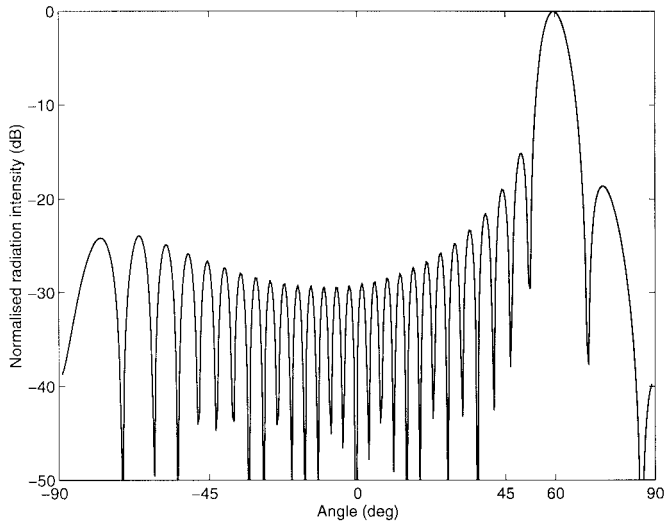


Fig. 4. Radiation pattern of the linear array with main beam steered to 60° using the best set of initial coefficients by the MMSE method.

The second column of Table III lists the normalized weighting vectors for the lowest relative sidelobe level obtained from the ten trials. For the broadside linear array, the coefficients are symmetrical and positive real values. To demonstrate that the algorithm can also be used for complex numbers, the main beam of the linear array is steered to $\theta_0 = 60^\circ$.

Another six weighting vectors are prepared as the initial population. The solid line curves in Figs. 4 and 5 show the radiation patterns before and after the GA operation, respectively. Again the leveled sidelobes in Fig. 5 indicate that the result obtained is closed to the optimum shown by the dotted line curve by Chebyshev method in Fig. 5.

B. Circular Array

In the second example, a circular array of 30 isotropic element with half-wave spacing is considered. In this example,

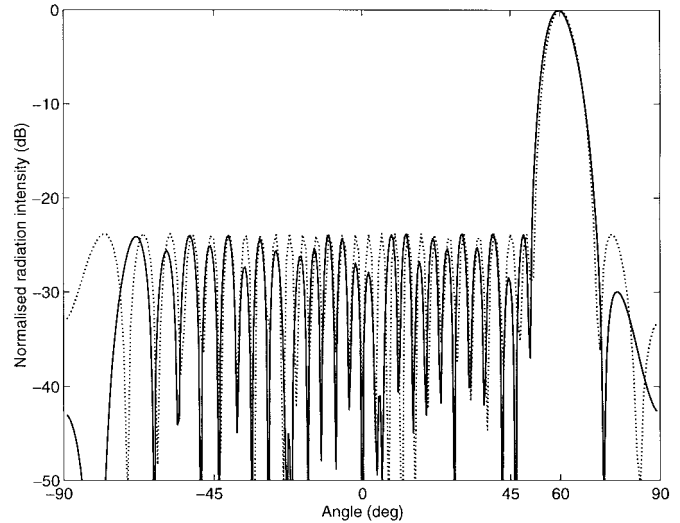


Fig. 5. Radiation pattern of the linear array with main beam steered to 60°. Solid line: by the GA, Dotted line: Chebyshev method.

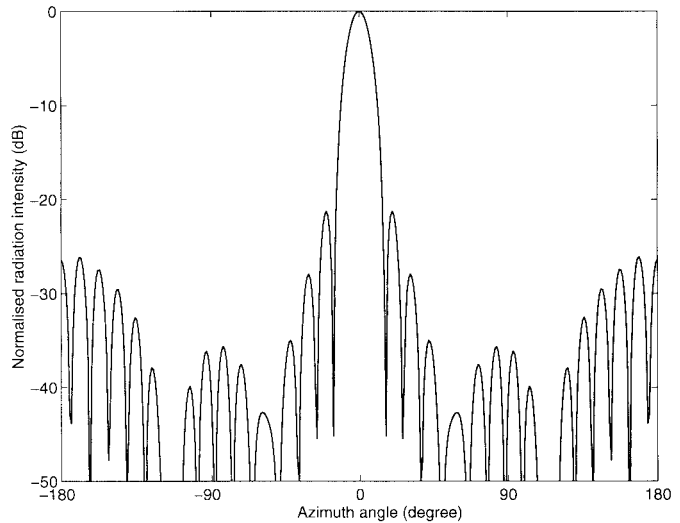


Fig. 6. Radiation pattern of the circular array using the best set of initial coefficients by the MMSE method.

symmetric excitation is assumed. The radiation pattern of a circular array can be evaluated by (2) assuming that $\phi_0 = 0^\circ$ and $\theta = \theta_0 = 90^\circ$ [1]:

$$F(\phi) = \sum_{n=1}^N w_n \exp\{jka[\sin\theta \cos(\phi - \phi_n) - \sin\theta_0 \cos(\phi_0 - \phi_n)]\}. \quad (2)$$

The resulting sidelobe levels are shown in Table II. There is a sidelobe level reduction from -21.25 dB (the best pattern of initial population) to -29.16 dB (the best pattern among the ten trials of the GA). The two patterns are also shown as the solid line curves in Figs. 6 and 7, respectively. The weighting vector of the solid line pattern in Fig. 7 is listed in the fourth column of Table III, showing that the excitation coefficients are with irregular phases.

To facilitate the practical implementation by n -bit digital phase shifters, the GA is modified to restrict the phase of

TABLE III
NORMALIZED EXCITATION COEFFICIENTS OF THE LOWEST RSLL FOR THE FOUR ARRAYS BY THE GA

Element	Broadside linear array (RSLL=-36.02 dB)	Linear array with $\theta_0 = 60^\circ$ (RSLL=-23.82 dB) (bottom half is the conjugate of the top half)	Circular array : without phase constraint (RSLL=-29.16 dB)	Circular array : with phase constraint to multiples of 11.25° (RSLL=-24.88 dB)
1 & 30	0.08	0.63 / -20.53°	1.00 / 47.77°	0.97 / 56.25°
2 & 29	0.17	0.32 / -14.19°	0.93 / 59.45°	0.85 / 56.25°
3 & 28	0.15	0.30 / -9.05°	0.54 / 134.90°	0.46 / 123.75°
4 & 27	0.25	0.31 / -5.51°	0.50 / -71.28°	0.49 / -67.50°
5 & 26	0.33	0.53 / -3.37°	0.52 / 63.71°	0.52 / 67.50°
6 & 25	0.42	0.47 / -4.38°	0.21 / 174.39°	0.22 / 168.75°
7 & 24	0.49	0.68 / -3.55°	0.47 / -58.20°	0.43 / -56.25°
8 & 23	0.59	0.72 / -3.06°	0.33 / -180.00°	0.31 / 180.00°
9 & 22	0.68	0.73 / -2.75°	0.44 / 57.65°	0.47 / 56.25°
10 & 21	0.71	0.90 / -3.00°	0.22 / -174.75°	0.19 / -168.75°
11 & 20	0.82	0.88 / -2.89°	0.52 / -63.44°	0.40 / -67.25°
12 & 19	0.88	0.87 / -2.34°	0.45 / 70.76°	0.48 / 67.25°
13 & 18	0.92	1.00 / -1.75°	0.53 / -133.11°	0.49 / -123.75°
14 & 17	1.00	0.90 / -1.13°	0.95 / -58.54°	0.88 / -56.25°
15 & 16	0.97	0.91 / -0.51°	1.00 / -47.11°	1.00 / -45.00°

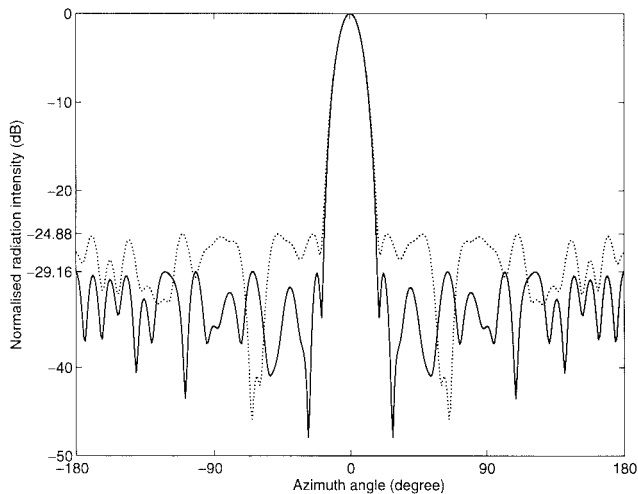


Fig. 7. Radiation patterns of the circular array by the GA. Dotted line: without phase constraint; solid line: with phase constraint to multiples of 11.25° .

coefficients to multiples of $2\pi/2^n$. This can be achieved by rounding up the phase of the newly generated chromosomes to a specified value after the crossover. A test is conducted to restrict the phase of the coefficients to multiple of 11.25° corresponding to the five-bit digital phase shifter. Same initial population as for the previous example is used. As shown in the third column of Table II, the best obtainable solution is about -24.88 dB. The corresponding radiation pattern and its weighting vector are shown by the dotted line curve in Fig. 7 and the fifth column of Table III, respectively. As expected, the overall results are not as good as the previous example due to the phase constraint on the excitation weighting vectors. However, the capability of imposing phase or magnitude constraints is very useful in designing practical

arrays to use preferable digital phase shifters and digital attenuators.

IV. CONCLUSION

A simple and flexible GA is proposed as a general purpose tool for array-pattern synthesis of arbitrary arrays. This approach avoids coding and directly works with real/complex numbers so as to simplify computing programming and to speed up computation. Although only linear and circular arrays are used to demonstrate the effectiveness of the method, the GA can be applied to arbitrary arrays.

REFERENCES

- [1] C. Balanis, *Antenna Theory—Analysis and Design*. New York: Wiley, 1982.
- [2] Y. C. Jiao, W. Y. Wei, L. W. Huang, and H. S. Wu, "A new low-side-lobe pattern synthesis technique for conformal arrays," *IEEE Trans. Antennas Propagat.*, vol. 41, pp. 824–831, June 1993.
- [3] M. H. Er, S. L. Sim, and S. N. Koh, "Application of constrained optimization techniques to array pattern synthesis," *Signal Processing*, vol. 34, pp. 323–334, Nov. 1993.
- [4] E. C. Dufort, "Pattern synthesis based on adaptive array theory," *IEEE Trans. Antennas Propagat.*, vol. 37, pp. 1011–1018, Aug. 1989.
- [5] C. A. Olen and R. T. Compton, Jr., "A numerical pattern synthesis algorithm for arrays," *IEEE Trans. Antennas Propagat.*, vol. 38, pp. 1666–1676, Oct. 1990.
- [6] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [7] R. L. Haupt, "Thinned arrays using genetic algorithms," *IEEE Trans. Antennas Propagat.*, vol. 42, pp. 993–999, July 1994.
- [8] A. Tennant, M. M. Dawoud, and A. P. Anderson, "Array pattern nulling by element position perturbations using a genetic algorithm," *Electron. Lett.*, vol. 30, no. 3, pp. 174–176, Feb. 1994.
- [9] M. Shimizu, "Determining the excitation coefficients of an array using genetic algorithm," in *IEEE Antenna Propagat. Soc. Int. Symp.*, Seattle, WA, June 1994, vol. 1, pp. 530–533.
- [10] L. Davis, Ed., *Handbook of Genetic Algorithms*. New York: Van Nostrand Reinhold, 1991.
- [11] G. J. E. Rawlins, Ed., *Foundations of Genetic Algorithm*. San Mateo, CA: Morgan Kaufmann, 1991.
- [12] S. U. Pillai, *Array Signal Processing*. New York: Springer-Verlag, 1989, pp. 184–185.



Keen-Keong Yan received the B.Eng. degree in electrical engineering (with honors) and the M.Eng. degree, both from Nanyang Technological University, Singapore, in 1992 and 1996, respectively.

Currently, he is working in Defence Science Organization, Singapore, as a Project Engineer. His main research interests are in the area of radar signal processing and antenna pattern synthesis.



Yilong Lu (S'90-M'92) was born in Chengdu, China. He received the B.Eng. degree from Harbin Institute of Technology, Harbin, China, in 1982, the M.Eng. degree from Tsinghua University, Beijing, China, in 1984, and the Ph.D. degree from University College London, U.K., in 1991, all in electronic engineering.

From 1984 to 1988, he was with the University of Electronic Science and Technology of China, Chengdu, China. Since 1992 he has been with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, where he is currently a Senior Lecturer. His research interests include computational electromagnetics, antennas, array beamforming, and application of genetic algorithms to optimize engineering parameters.

Dr. Lu is a member of the Applied Computational Electromagnetics Society (ACES).