# AN IMPROVED GENETIC ALGORITHM FOR APERIODIC ARRAY SYNTHESIS

*Ling Cen*[1], *Wee Ser*[1], *Zhu Liang Yu*[1], *Susanto Rahardja*[2]

[1] Centre for Signal Processing, Nanyang Technological University,
[2] Institute for Infocomm Research, A* Star

## ABSTRACT

In this paper, a novel algorithm on beam pattern synthesis for linear aperiodic arrays with arbitrary geometrical configuration is proposed. The algorithm is based on an Improved Genetic Algorithm (IGA) that simultaneously adjusts the weight coefficients and inter-sensor spacings of a linear aperiodic array. A novel section-based crossover and a self-supervised mutation process are developed to improve the convergence performance. The results from simulation illustrate that with the IGA, the Peak Sidelobe Level (PSL) of the synthesized beam pattern has been successfully lowered. In addition, the computational cost of the proposed algorithm can be as low as being about 10% of that of a recently reported genetic algorithm based synthesis method. The robustness of the proposed IGA has been illustrated clearly from the statistic of multiple independent runs too. The excellent performance of the IGA makes it a promising optimization algorithm where expensive cost functions are involved.

***Index Terms*—**linear arrays, aperiodic arrays, genetic algorithms (GAs), peak sidelobe level (PSL), beam pattern synthesis

## 1. INTRODUCTION

Unequally-spaced arrays [1], also termed as aperiodic arrays have been studied for several decades. Compared with equally-spaced array, aperiodic arrays have the advantages that they are capable of achieving higher spatial resolution and lower sidelobe with a smaller number of sensors. Over the past several decades, many analytical and numerical based array beam pattern synthesis techniques have been developed [2]-[8]. An example of the analytical techniques is reported in [6], in which the inter-sensor spacings for a given array weight distribution are determined by performing a Legendre transformation on the array factor. Examples of numerical techniques include the use of linear or non-linear optimization methods such as the simplex algorithm [2], simulated annealing algorithm [3], differential evolution algorithm [4], and genetic algorithm [5], [7]-[8].

The challenge of determining optimum parameter values simultaneously stems from the non-linear and non-convex dependency of the array factor to the weights and the sensor positions [4]. The performance of the employed optimization scheme is an important factor in the success of a pattern synthesis method, in terms of solution quality, computational load, and robustness. The intrinsic ability to cope with nonlinear problem makes Genetic Algorithms (GAs) a suitable solution [7]. It has been employed in the pattern synthesis of aperiodic arrays for many years [5], [7]-[8]. However, as stochastic search techniques, the major disadvantages of the GAs are premature convergence and slow convergence speed, especially when they are employed to solve complicated problems with a large solution space and numerical local optima [9].

To address the problem, an Improved Genetic algorithm (IGA) is proposed in this paper. In the IGA, a multi-section based chromosome arrangement is proposed, which allows the optimization to handle a wide variety of constraints and evolution trends. Associated with the multi-section encoding scheme, a novel multi-section based crossover process is proposed for the proceeding of real variables. In order to improve the convergence performance of stochastic optimization process, a novel self-supervised mutation method is presented. These improvements make the proposed IGA more robust, statistically sound and faster in convergence.

The remaining part of this paper is organized as follows. The array synthesis formulation for aperiodic arrays and the GA based synthesis method are briefly described in Section 2 and 3. The proposed IGA for the synthesis of aperiodic arrays is presented and discussed in Section 4. Section 5 describes the simulation study and shows the comparative performance of the proposed technique. Concluding remarks are given in Section 6.

## 2. APERIODIC ARRAY SYNTHESIS FORMULATION

Assume an aperiodic and asymmetrical linear array with $N$ sensors as shown in Fig. 1. The array factor $AF(\theta)$ can be characterized as [1]

1-4244-1484-9/08/$25.00 ©2008 IEEE

$$AF(\theta)=\sum_{n=0}^{N-1} w_n \exp[jkx_n \cos\theta], \text{where } x_n=\begin{cases}\sum_{i=1}^{n} d_i, \text{ for } n\geq 1\\ 0, \text{ for } n=0\end{cases} \quad (1)$$

and $k$=2π/$\lambda$, $\lambda$ is the wavelength, $\theta$ is the steering angle measured with respect to the $x$ axis, $x_n$ is the distance between the first and the $n^{\text{th}}$ sensors measured in wavelength $\lambda$, $d_i$ is the inter-sensor spacing between the $(i\text{-}1)^{\text{th}}$ and the $i^{\text{th}}$ sensors, and $w_n$ is the weight coefficient of the $n^{\text{th}}$ sensor. Since $w_n$ is complex, it can be expressed as $w_n = a_n \exp(j\varphi_n)$ , where $a_n$ and $\varphi_n$ are the amplitude and phase of $w_n$ respectively. Consequently, the array factor can be expressed as

$$AF(\theta)=a_0\exp(j\varphi_0)+\sum_{n=1}^{N-1} a_n \exp[j(k\sum_{i=1}^{n} d_n \cos\theta+\varphi_n)]. \quad (2)$$

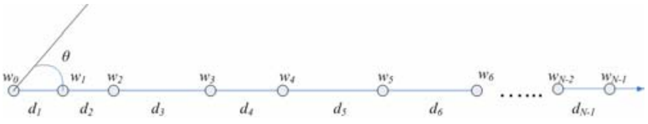

Fig. 1 Geometry of an aperiodic and asymmetric linear array with $N$ sensors.

## 3. ARRAY SYNTHESIS USING GENETIC ALGORITHM

The task of array synthesis is to design the parameters of the array so that it will produce a pattern close to the desired beam pattern. When applying the GA for array pattern synthesis, the sensor parameters are encoded and cascaded to form a chromosome which represents a potential solution. A specified number of chromosomes can be used to construct a population, which will then evolve through selection, breeding and genetic variation. With the help of such an evolutionary process, the parameters of the array can be synthesized.

In the GA, the fitness function (or objective function), *fit*, serves to evaluate the performance of each chromosome. A chromosome having a larger fitness value implies a fitter individual and hence a better solution. As the objective of optimization is to minimize the sidelobe level of the array pattern by adjusting the parameters of the array, subject to given design specifications and constraints, the fitness function can be defined with the evaluation of the Peak Sidelobe Level (PSL) as

$$fit=-20\log_{10}\left(\max_{\theta_s\in\text{Sidelobe}}\left|\frac{AF(\theta_s)}{\max(AF(\theta))}\right|\right), \quad (3)$$

where $\theta_s$ is the spanned angles within the sidelobe band. In EQ. (3), the PSL is measured in the unit of decibel. Here, we introduce the minus "-" in order to make it a maximization problem.

## 4. PROPOSED IMPROVED GENETIC ALGORITHM

Over the past decades, the Genetic algorithm (GA) has been widely applied to array synthesis [5], [7]-[8], [10]-[14]. Although the GA is capable of solving non-linear and non-convex problems, it is computationally intensive and its convergence is not guaranteed, especially when the solution space is larger [9]. In order to enhance the convergence performance of the GA, an Improved Genetic Algorithm (IGA) is proposed in this paper.

In the IGA, a multi-section based real encoding scheme for the gene arrangement in a chromosome is proposed, which allows the optimization to handle a wide variety of constraints and evolution trends. The chromosomes are represented using floating-point numbers, which represent the parameter vectors of the array. Here, we arrange the sensor weights and locations in different variable-sections, e.g. weight-section and spacing-section (or amplitude-section, phase-section and spacing section if the weights are complex numbers). A chromosome is formed by cascading all the sections. Arranging the chromosome in this multi-section way has the advantage that, it takes care of the possibility of different types of parameters that may have different constraints and evolutionary trends.

Associated with our encoding scheme, a section based crossover process is proposed for real variables. This allows the independent control of evolution process for each type of array parameters. Three methods of crossover, i.e. uniform crossover, single-point crossover and multi-point crossover, are randomly applied in each generation during the evolutionary process. In the uniform crossover, the crossover is performed over the entire chromosome via a randomly generated mask. In the single-point crossover, one cut-point is randomly chosen from the parents and the parts located in the right of the cut-point are exchanged. Similarly, for the two-point crossover, two cut-points are selected and the parts between the two points are exchanged.

Here, we compute the new cut-point genes, $g_{c1}$ and $g_{c2}$, by taking a linear combination of the old cut-point genes, $g_{p1}$ and $g_{p2}$, the upper and lower bounds, $g_{low}$ and $g_{up}$, of the gene, and a randomly generated value, $c$,

$$\begin{aligned} g_{c1}&=\min\left[g_{low}+c(g_{up}-g_{low}),\frac{g_{p1}+g_{p2}}{2}+c\frac{g_{p1}-g_{p2}}{2}\right], \\ g_{c2}&=\max\left[g_{low}+(1-c)(g_{up}-g_{low}),\frac{g_{p2}+g_{p1}}{2}+(1-c)\frac{g_{p2}-g_{p1}}{2}\right], \end{aligned} \quad (4)$$

where $g_{c1}$ and $g_{c2}$ are the new cut-point genes in the two children, $g_{p1}$ and $g_{p2}$ are the old cut-point genes in the two parents, $g_{low}$ and $g_{up}$ are the lower and upper bounds of the gene, respectively, and $c$ is a randomly generated number between [0,1].

After the offspring are produced from the crossover process, they will then undergo a mutation process. It is well known that as stochastic search techniques the GA does not always evolve towards a good solution; it only evolves away from bad circumstances. Therefore, the GA risks finding a suboptimal solution and has low convergence speed in complex applications. To address the problem, a novel self-supervised (in place of the usual stochastic) mutation is proposed in the IGA. The results from the previous searches are used to adjust the direction and step size of the subsequent search in the mutation process, so that it is capable of finding better solutions in "good" searching areas, and preventing the evolution from repeatedly exploiting "bad" areas. By this way, the IGA can be more efficient with less computational load over the GA.

Let us define each update of a gene during mutation as a Search. In the self-supervised scheme, the results obtained from the previous Searches are used to adjust the direction and the step size of the subsequent Search. At the beginning of the mutation process, a Search is performed on the gene in an arbitrary direction. If a better solution can be obtained in that direction, the mutation will continue with a new Search. The Searching step sizes are gradually decreased within a local area of the gene value. The Search in the same direction will stop when there is no significant solution improvement. This approach finds the best or near-best solution in one direction, and as such, the IGA can be more robust, statistically sound and faster in convergence.

Let the gene to be mutated be denoted as $g_i$, the new gene $g_i{}'$ after one Search is calculated as

$$g_i{}' = \begin{cases} g_i + M_i \cdot (g_{upi} - g_i), g_i \geq 0 \\ g_i + M_i \cdot (g_{lowi} - g_i), g_i < 0 \end{cases}, \text{ if } 0<M \leq 1, \qquad (5)$$

$$g_i{}' = g_i + M_i \cdot g_i \qquad \text{if } -1 \leq M \leq 0,$$

where $g_{lowi}$ and $g_{upi}$ are the lower and upper bounds of $g_i$. $M_i$ is a real number within the range of [-1, 1], which is randomly initialized at the beginning of the mutation. After each Search, $M_i$ is decreased by multiplying a factor $r_{dec}$ to it.

$$M_i{}' = r_{dec} \cdot M_i, \text{ if } fit_{new} - fit_{old} > e \text{ AND } fit_{old} > fit_{ave}, \qquad (6)$$

where $M_i{}'$ is the new value of $M_i$, $r_{dec}$ is the decreasing rate satisfying $0<r_{dec}<1$, $fit_{old}$ and $fit_{new}$ are the original and the new fitness values, respectively, $fit_{ave}$ is the average fitness value of the offspring from the crossover, and $e$ is the minimum acceptable improvement. After updating $M_i$ and $g_i$, a new Search will start with $M_i{}'$ and $g_i{}'$. The mutation Search on $g_i$ is considered to be fully completed when the condition in (6) cannot be met.

Through the process of crossover and mutation, new chromosomes are produced. Such process is iterated till the algorithm converges or the termination conditions are met.

## 5. NUMERICAL EXAMPLE AND RESULTS ANALYSIS

In order to compare the performance of the IGA with that of the Modified Genetic Algorithm (MGA) proposed in [7], let us consider the synthesis of two pencil-beam patterns. The main beams are confined to be within $0 \leq u \leq 2/N$, where $u = \cos(\theta)$, and $N = 17$ and 37, respectively. In [7], the two patterns were synthesized with the MGA method for a symmetrical aperture and using a fixed set of weights. Lower Peak Sidelobe Levels (PSL) were achieved compared to those obtained from the analytical technique proposed in [6]. Our proposed IGA will be compared with the MGA[7] in terms of PSL, convergence speed and algorithm robustness.

In the IGA, the population size is set to 30. According to our experience, the parameters in the crossover and the mutation process are selected as $p_c$ (crossover probability) = 0.8, $p_m$ (mutation probability) = 0.2, $r_{de}$ = 0.6, and $e$ = 0.001. The computational complexity is measured in terms of the number of fitness function evaluated. In order to find the average performance of the IGA and gain an insight of its robustness, the IGA is executed for 10 runs for each set of parameter values considered. The number of generations is 50 for each run.

The resultant lowest PSL of each run is shown in Fig. 2. The best-case (i.e. lowest) lowest PSL for the IGA is -20.32 dB. In Fig. 2, the dashed line shows the best-case lowest PSL reported in [7] using the MGA. For the MGA, the population size is 200 and the number of generations is 300 [7]. It can be seen from Fig. 2 that the best–case lowest PSL for the MGA is 0.5 dB higher than that of our proposed IGA. Fig. 2 also shows that the lowest PSLs obtained by the IGA in 9 out of 10 runs are better than the best-case value obtained by the MGA. Simulation runs also show that the worst-case and average lowest PSL for the MGA are also poorer than that of our proposed IGA. The resultant sensor positions obtained by the IGA are shown in Fig. 3 (a). The resultant beam patterns obtained by the IGA and the MGA are shown in Fig. 3 (b).The above are reported for a 37-sensor array (i.e. $N$=37) and the corresponding results are shown in Figs. 4 and 5. As the Figures show, the improvements made by the IGA are even clearer fro this (larger) value of $N$.

It should be noted from the comparative results shown in Figs. 2-5 that the IGA not only achieves a better PSL performance, but also requires a smaller population size (30 individuals) and involves a shorter process time (50 generations). Furthermore, the average numbers of the fitness function evaluated required by the IGA are only about 10% ($N$=17) or 17% ($N$=37), of that required by the MGA. This is a significant saving in the computational effort as compared to that needed by the MGA.

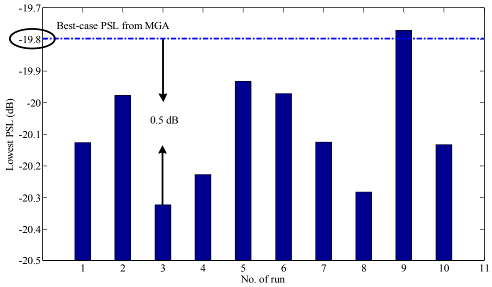

Fig. 2 Resultant lowest PSLs obtained by the proposed IGA in 10 runs ($N$=17). The dashed line shows the best-case lowest PSL for the MGA.

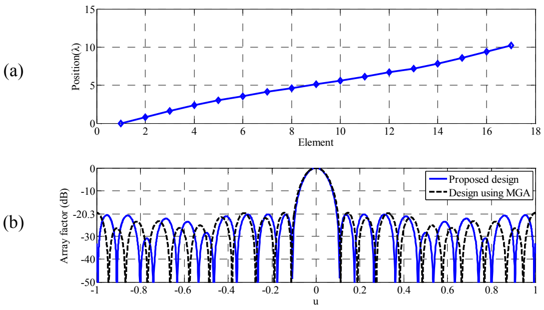

Fig. 3 (a) Sensor positions obtained using the proposed IGA. (b) Resultant beam patterns of IGA and MGA.

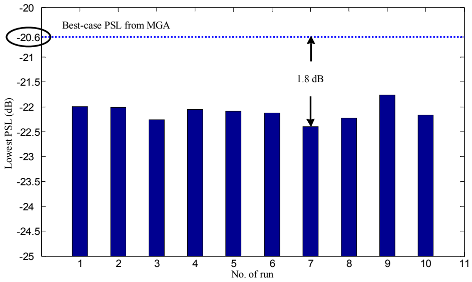

Fig. 4 Resultant lowest PSLs obtained by the proposed IGA in 10 runs ($N$=37). The dashed line shows the best-case lowest PSL for the MGA.

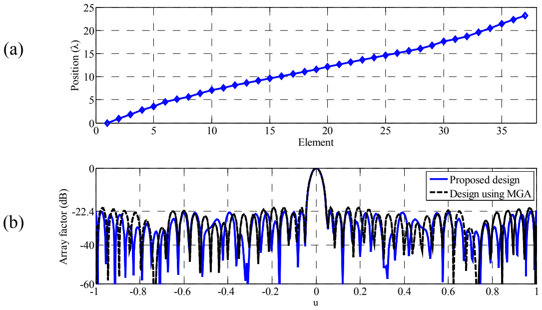

Fig. 5 (a) Sensor positions obtained using the proposed IGA. (b) Resultant beam patterns of IGA and MGA.

## 6. CONCLUSION

In this paper, a novel beam pattern synthesis algorithm has been proposed for linear aperiodic arrays with arbitrary geometrical configuration. To address the high-dimension non-linear problem involved in pattern synthesis, an Improved Genetic Algorithm (IGA) is proposed in this paper. In the IGA, a novel crossover process is proposed for the optimization of real variables. To overcome the drawbacks of stochastic process, a novel self-supervised mutation is applied in the IGA. Simulation results show that the proposed IGA is able to achieve lower PSL and higher convergence rate compared to a recently reported genetic algorithm based synthesis approach [7]. The new design leads to savings on computational efforts of up to 89.9% compared to the results reported in [7]. The excellent comparative performance of the proposed IGA makes it a promising optimization algorithm for aperiodic array synthesis where a high cost function is involved.

## 7. REFERENCES

[1] A. Ishimaru, “Theory of unequally-spaced arrays,” *IEEE Trans. Antennas Propagat.*, vol. 10, no. 6, pp. 691–702, Nov. 1962.

[2] R. M. Leahy and B. D. Jeffs, “On the design of maximally sparse beamforming arrays,” *IEEE Trans. Antennas Propagat.*, vol. 39, no. 8, pp. 1178-1187, Aug. 1991.

[3] A. Trucco and V. Murino, “Stochastic optimization of linear sparse arrays,” *IEEE J. Oceanic Eng*, vol. 24, no. 3, pp. 291–299, Jul. 1999.

[4] D. G. Kurup, M. Himdi, and A. Rydberg, “Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm,” *IEEE Trans. Antennas Propagat.*, vol. 51, no. 9, pp. 2210–2217, Sep. 2003.

[5] R. L. Flaupt, “Thinned arrays using genetic algorithms,” *IEEE Trans. Antennas Propagat.*, vol. 42, no. 7, pp. 993–999, Jul. 1994.

[6] B. P. Kumar and G.R. Branner, “Generalized analytical technique for the synthesis of unequally spaced arrays with linear, planar, cylindrical or spherical geometry,” *IEEE Trans. Antennas Propagat.*, vol. 53, no. 2, pp. 621–634, Feb. 2005.

[7] K. S. Chen, Z. S. He, and C. L. Han, “A modified real GA for the sparse linear array synthesis with multiple constraints”, *IEEE Trans. Antennas Propagat.*, vol. 54, no. 7, pp. 2169-2173, Jul. 2006.

[8] A. Lommi, A. Massa, E. Storti, and A. Trucco, “Sidelobe reduction in sparse linear arrays by genetic algorithms,” *Microwave and Optical Technology Letter*, vol. 31, no. 3, pp. 194-196, Feb. 2002.

[9] K. F. Man, K. S. Tang, and S. Kwong, *Genetic Algorithms: Concepts and Designs*, Springer Verlag, London, 1999.

[10] M. J. Buckley, “Linear array synthesis using a hybrid genetic algorithm,” *IEEE Antennas and Propagation Society International Symposium*, 1996. vol. 1, pp. 584-587, Jul. 1996.

[11] K. K. Yan and Y. L. Lu, “Sidelobe reduction in array-pattern synthesis using genetic algorithm,” *IEEE Trans. Antennas Propagat.*, vol. 45, no. 7, pp. 1117-1122, Jul. 1997.

[12] D. Marcano and F. Duran, “Synthesis of antenna arrays using genetic algorithms,” *IEEE Antennas and Propagation Magazine*, vol. 42, no. 3, pp. 12-20, Jun. 2000.

[13] D. W. Boeringer and D. H. Werner, “Particle swarm optimization versus genetic algorithms for phased array synthesis,” *IEEE Trans. Antennas Propagat.*, vol. 52, no. 3, pp. 771-779, March 2004.

[14] L. L. Wang and D. G. Fang, “Synthesis of nonuniformly spaced arrays using genetic algorithm,” *Asia-Pacific Conference on Environmental Electromagnetics*, pp. 302-305, Nov. 2003.