Journal of Information Hiding and Multimedia Signal Processing ©2017 ISSN 2073-4212
Ubiquitous International Volume **8**, Number **4**, July **2017**

An Improved Genetic Algorithm for Thinning Acoustic Array

He Li[1,2], Zhi-Hong Liu[2], Chui-Jie Yi[2] and Zhe-Ming Lu[3]*

[1]School of Communication and Electronic Engineering
Qingdao University of Technology
Qingdao 266033, China
sdlclh@126.com

[2]College of Mechanical Engineering
Qingdao University of Technology
Qingdao 266033, China
lzhqingdao@163.com, chuijieyi@vip.163.com

[3]School of Aeronautics and Astronautics
Zhejiang University
Hangzhou 310027, China
*Corresponding author: zheminglu@zju.edu.cn

Received October, 2016; revised March, 2017

Abstract. *Optimization of large arrays, which improves the efficiency, plays an important role in acquiring signals in the acoustic field. Recently, artificial intelligence algorithms, such as genetic algorithms (GAs) were widely used in sparse array optimization for their excellent performance in global optimization. This paper proposes a new initial deployment forming method: "random place assignment" and proposes new Cross and Mutation rules in GA algorithm to improve the performance of acoustic array. The improved GA algorithm reduces the amount of computation and increases the convergence rate. The number of generations decrease from 30 [1,12] to less than 20 under the same condition. Also the max side lobe level (max SLL) is reduced. Simulations are conducted on a linear array composed of 200 isotropy or directional elements. For isotropy elements, the max SLL is -27.06dB which is lower than [1](-22.09 dB),[3](-24.03),[8](-23.05 dB) and [13] (-22.13 dB) when* 77% *of elements are left. While for directional elements, the max SLL is -27.15dB, lower than [1](-23.69 dB)and [3](-25.25 dB when* 75% *of elements are left.The improved GA performs better than the previous methods in terms of optimization for asymmetric line arrays. Furthermore, our algorithm is robust to the directivity of elements and the steering angle of the linear array.*
Keywords: Acoustic array, Thinned array, Genetic algorithm, Linear array, Max SLL, Binary optimization.

1. Introduction.

It is difficult to record speech or other types of audio signals in the environment with a large amount of noise and crosstalk, which consequently influences the multimedia signal processing. Currently, the method to solve this issue is to adopt large-scale arrays such as Uniform Linear Array (ULA) accompanied with beamforming algorithms. Researchers have achieved some results in improving the beamforming algorithms. However, the optimization of the arrangement of the array is still an efficient way to enhance the accuracy of receiving audio signals. Because the thinned linear arrays are the most widely used configuration for a thinned array and many people thinned their arrays on the base of the ULA [1,2,4,13], we followed it. In this paper, we optimize the arrangement of the array to receive the audio signal from a specific direction accurately, and maximum restrain the voice and other types of signals from other directions. After arrangement optimization, the array performs well on restraining the side lobes. This idea resembles the optimization of the sparse antenna array[1], from which we have learned the spirit.

To optimize arrays with a large number of elements, researchers used natural methods especially global evolution methods like GAs to get the best solution.

First, Haupt used the simple genetic algorithm (SGA) to thin the symmetrical line array for narrower main lobe and lower maximum side lobe [1]. Global optimal solution made optimizing a large number of parameters or discrete parameters possible. But there are other troublesome questions like heavy computation and slower convergence speed. Also, researchers have tried to change the structure of the array. Chen et al. replaced a symmetry linear array with an unsymmetrical linear array [2]. The simulation result implies lower max SLL.

In addition, researchers tried to improve GA even by fusing it with other global optimization algorithms. Hamici and Ismail came up with Stochastic Immunity Genetic Algorithm (IGA) which changed the cross operator [3]. New expression of the array factor took full advantage of the linear Discrete Cosine Transform (DCT) and resulted in a high-speed computation. However, its population size was so large that the order of magnitude already reached millions. Oliveri and Massa combined ADS (Almost Difference Set) with GA to create a new algorithm named ADSGA (genetic algorithm (GA)-enhanced almost difference set (ADS)-based methodology) which could not only break the limit of complexity and optimization but also reduced the computational cost if only with GA [4]. We aim to find a method which has advantages both in the physical meaning of algorithm and excellent performances.

In this paper,a new initial deployment forming method: "random place assignment" and new Cross and Mutation rules in GA algorithm are presented. Simulations prove that our algorithm has lower max SLL compared with other methods. The rest of this paper is organized as follows. In Section 2, we explain the framework of improved GA especially innovation. In Section 3, we conduct simulations, and compare results under different conditions. In the last section we summarize the paper with our contribution and future work.

2. **Improved Genetic Algorithm.** The improved GA scheme includes encoding the array parameters, defining the fitness function, original population forming, selection, cross and mutation.

2.1. **Encode.** The first step is to encode the parameters to gene with clear physical meaning. To illustrate the process clearly, let us take the linear array for far fields as an example. The geometry of N-element linear array is shown in Figure 1. The interval between two positions is $d = 0.5\lambda$ and the aperture $D = (N-1)\lambda/2$. An "on" means there is a microphone element at this position and an "off" means no microphone element. We use a solid dot to denote "on" and a small circle for "off". Parameter "a_n" represents whether there is a microphone element placed at the position. If a_n=1, there is an element, otherwise no element.

An N-bit gene consist of $\{a_0, a_1, ..., a_{n-1}\}$ which embodies the information about the allocation of microphones. Assume there are M elements "on" and $N-M$ elements "off",

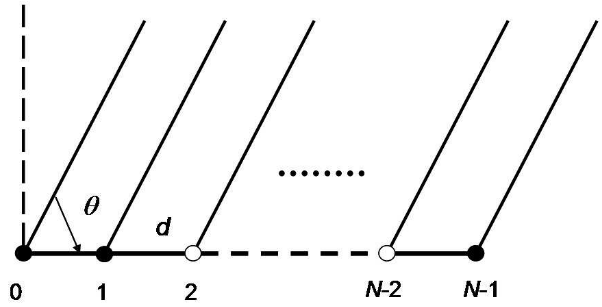


FIGURE 1. Geometry of an N-element linear array

accordingly every gene is a binary code with M "1"s and $N-M$ "0"s. We define the sparse rate as $\eta = M/N$.

2.2. **Fitness function.** Normally a fitness function in array optimization is a performance parameter of the array such as maximum side lobe level (max SLL) [1, 2, 4-7], combination of side lobe level (SLL) and null control in specific directions [8, 9], combination of max SLL and half power beam width [10]. According to [11], when the aperture is fixed, the width of the main lobe varies little. It means the signal reception ability in an expected direction is fixed. So we select the max SLL which represents the ability to restrain signals in unexpected directions as the fitness function. Fig.1 shows how to calculate the output from the linear array and its max SLL.

For simplicity, we take the leftmost element numbered "0" as the reference one. So the coordinate of position from left to right is marked with $x_0, x_1, ..., x_{N-1}$ as shown in Fig 1. The interval between acoustic signal reaching two adjacent positions is $\tau_n = x_n \cdot cos(\theta)/c$, ($c$: wave speed, θ: angle between axis and direction of wave, $\theta \in [0, \pi]$). So the output of all microphone elements is:

$$Y(\theta) = \sum_{n=0}^{N-1} a_n \omega_n e^{-j\omega\tau_n} \cdot eplat(\theta) = \sum_{n=0}^{N-1} a_n \omega_n e^{-jkx_n cos\theta} \cdot eplat(\theta) \tag{1}$$

Where k is the wave number which equals to $2\pi/\lambda$, x_n is the n-th microphone element, λ is the wavelength, and ω is the angular frequency of the acoustic signal. We denote the element pattern as $elpat(\theta)$, for directional elements, $elpat(\theta) = |sin\theta|$, while for isotropic elements, $eplat(\theta) = 1$. We use a_n to denote coefficients, $a_n \in \{1, 0\}$. If a_n=1, the n-th element is kept and corresponds to a solid point in the figure; otherwise if a_n=0, the n-th element is removed and corresponds to an empty point in the figure. The weight of the n-th element is denoted by $\omega_n, n = 0, 1, \cdots, N-1$. For the steering angle θ_d,ω_n=$e^{jkx_n cos\theta_d}$. Thus, the output $Y(\theta)$ can be derived as:

$$Y(\theta) = \sum_{n=0}^{N-1} a_n e^{-jkx_n(cos\theta - cos\theta_d)} \cdot eplat(\theta) \tag{2}$$

We define max SLL as the fitness function, which can be expressed as follows.

$$fitness(d_1, d_2, ..., d_{N-1}) = max\,SLL = 20log\left|\frac{Pr_{max\_rsl}}{Pr_{max}}\right| = 20log\max_{\theta\in S}\left\{|\frac{Y(\theta)}{\max(Y(\theta))}|\right\} \tag{3}$$

Where Pr_{max_rsl} is the maximum of side lobe, Pr_{max} is the maximum of the main lobe. When the array direction is θ_d , the first null point is $\theta_d \pm \theta_0$, S=$\{\theta|0^o \leq \theta \leq \theta_d - \theta_0 \, or \, \theta_d + \theta_0 \leq \theta \leq 180^o\}$. Our goal is to minimize fitness $(d_1, d_2, ..., d_{N-1})$.

2.3. **Initial population.** We propose a new initial deployment forming method: "random place assignment". As we learn from [12], if the positions of microphone satisfy the Gaussian distribution, the random microphone array can record acoustic data more precisely. So the method can get initial populations which need less evolution. Another virtue is that the number of elements can be fixed. It means we should choose M positions randomly from N positions to place elements ($M < N$). First, we assign "1"s to both ends of the gene to guarantee the aperture of the array. Then we create a sequence of $N-2$ digits of which $M-2$ digits are "1"s and the remaining are "0"s. We randomly rearrange the order of integer numbers from 1 to $N-2$.

Then we compare each integer with M-2, if the integer is less than or equal to M-2, then we overwrite it with "1". Otherwise, we overwrite it with "0". It includes three steps:

(1) Generates a random series of integers $B_{N-2} = \{b_i | b_i \in [1, N-2], i = 1, 2, ..., N-2 \, and \, \forall \, i \neq j, b_i \neq b_j\}$.

(2) Confirm the middle bits $A_{N-2} = \{a_i, i = 1, 2, ..., N-2\}$. If $b_i > M-2$, then a_i ="0". Otherwise, a_i="1".

(3) Add two "1"s to both ends of the gene.

To explain the process clearly, we take N=10, thinning rate $\eta = 0.8$ for example. Under this circumstance, $M = N \times \eta = 8$. Both ends of the gene are "1" and we should find other six digits in the gene.

(1) First we produce a random integer sequence $B_{N-2} = [8\,2\,7\,4\,3\,6\,5\,1]$.

(2) Get the middle bits $A_{N-2} = [0\,1\,0\,1\,1\,1\,1\,1]$ according to M-2.

(3) Add two "1"s to both ends and get the initial gene $A_N = [1\,0\,1\,0\,1\,1\,1\,1\,1\,1]$.

2.4. **Selection, Cross and Mutation.** Selection, cross and mutation operations are three basic operations in GA. The improved GA can reatain the numer of "1"s and only change their places in the corss and mutation process.

(1) Selection

Evaluate the fitness function for every gene and sort them in the ascending order. Assume the select rate is a constant $p_s (0 < p_s < 1)$ and the population size is K, we keep the first $K \times p_s$ genes reserved as parents and discard the others. Then, we generate offspring from their parents.

(2) Cross

Cross is a process which helps keep the effective chromosomes. The specific procedure is as follows. First, we select the different bits between parents. Here we assume the cross rate is $p_c (0 < p_c < 1)$ and the number of different bits is N_d. Then we should select $N_d \times p_c$ bits randomly among "0"s and change them into "1"s. Also, we select $N_d \times p_c$ bits randomly among "1"s to change them into "0"s. Since the number of changes from "0"s to "1"s is equal to the number of changes from "1"s to "0"s, the number of "1"s in one gene remain unchanged.

(3) Mutation

Mutation is a process to change the same bits in two parent genes in a certain small probability p_m. The process is similar to the cross operation except that the change is performed on the positions with the same bits between parents.

Cross and mutation operations can make sure the number of "1"s does not change. We can guarantee that the total number of "1"s are unchanged, meanwhile the distribution of elements are changed with cross and mutation operations.

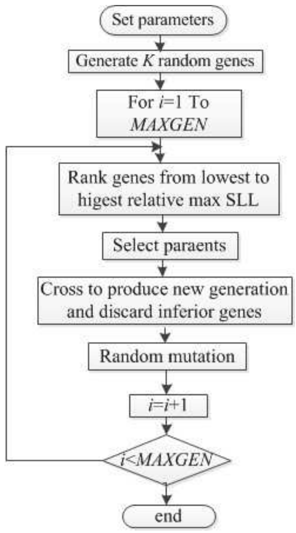
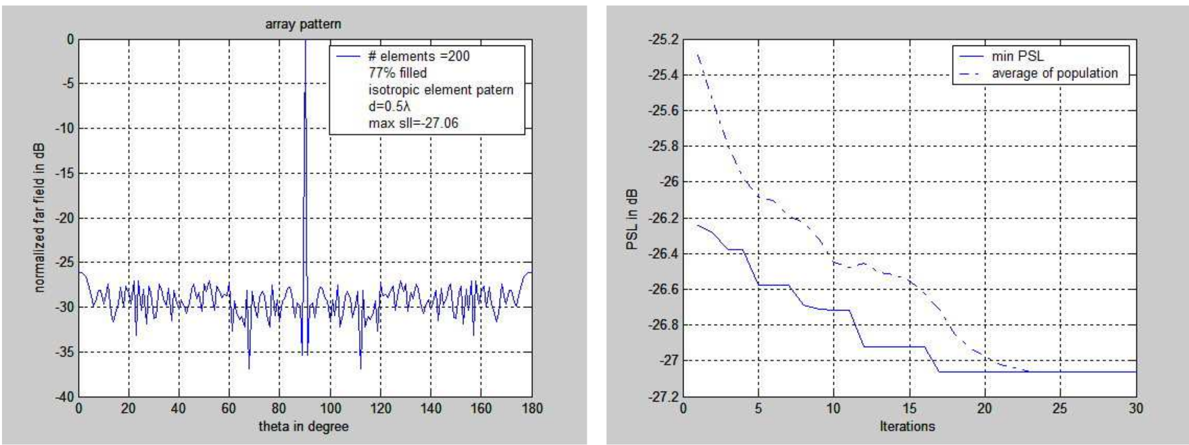


FIGURE 2. Flow chart of improved genetic algorithm



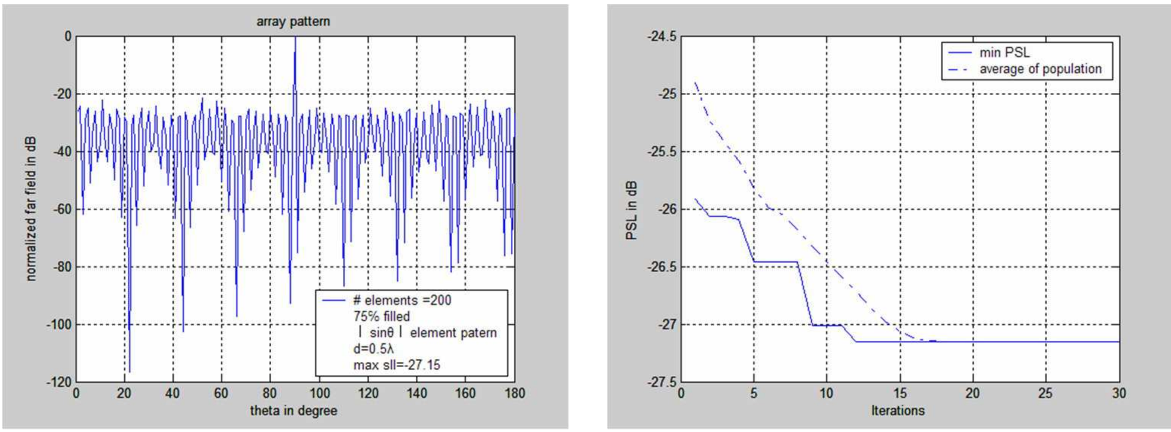
(a) (b)

FIGURE 3. (a) Far-field pattern of a thinned array with 200 isotropic elements($\theta_d = 90°$)(b)convergence curves

Flowchart for our improved GA algorithm is shown in Figure 2, where $MAXGEN$ stands for the maximum number of generations.

3. **Simulation.** Because a large-scale array helps evaluate the performance of an algorithm, a linear array with 200 elements is chosen for simulation [1-4, 13]. We use the same number of elements. Assignments for other parameters are as follows, the size of population K=50, the select rate ps=0.5, the cross rate $p_c = 0.1$, the mutation rate $p_m = 0.01$, and the max number of generations $MAXGEN$=30.

3.1. **Results for Different Kinds of Microphone Elements.** Our simulation is carried out on isotropic microphone elements and directional microphone elements respectively for general use. To compare with [1, 2, 4, 13], we run our algorithm to optimize a linear array of isotropic microphone elements with the sparse rate $\eta = 0.77$ and the steering angle $\theta_d = 90°$, which is also called broadside linear array. The max SLL and the convergence curve are shown in Figure 3.



(a) (b)

FIGURE 4. (a) Far-field pattern of a thinned array with 200 directional elements($\theta_d = 90°$)(b)convergence curves

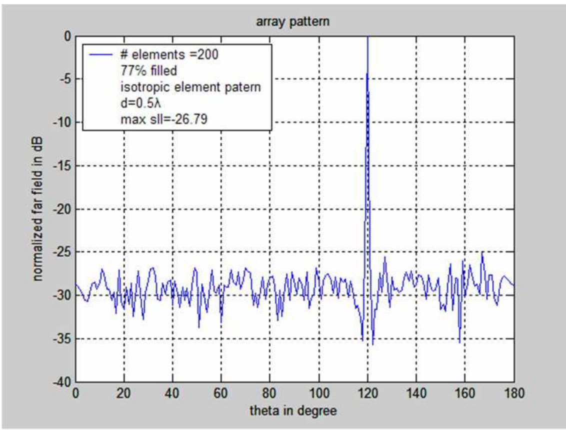
To compare with [1, 2], we take the sparse rate $\eta = 0.75$ with the element pattern $elpat(\theta) = |sin\theta|$. The max SLL and the convergence curve are shown in Figure 4.

Obviously, our algorithm enhances the optimization performances of the broadside linear array with isotropic elements. The max SLL of the new array is -27.06dB, better than -22.09dB in [1], -24.03 dB in [2], -23.05dB in [4], and -22.13dB in [13] as shown in Table 1. Also for the broadside linear array composed of directional elements, the max SLL of the optimization array is -27.15dB, which is better than -23.69dB in [1] and -25.25dB in [2]. We show the improvement by our algorithm in the column "Improvement".

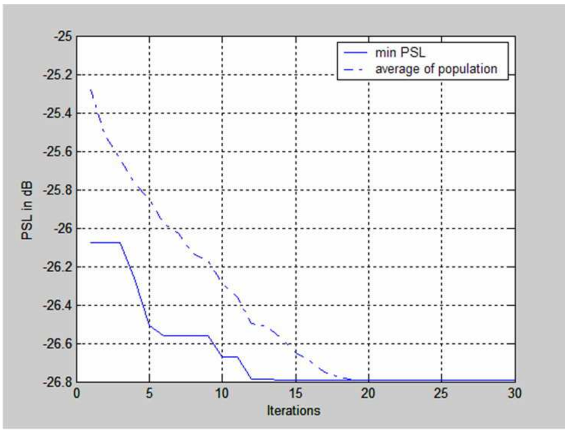
There is little difference between the optimization result of the broadside linear array with directional microphone elements and that of isotropic elements. To compare easily, we list all the results obtained by different methods in Table 1. In [1], the linear array of directional elements ($\eta = 0.75$, max SLL=-23.69 dB) is better than that of isotropic elements ($\eta = 0.77$, max SLL=-22.09 dB).In [2] the values are -24.03 dB and -25.25dB respectively. With our method, we get -27.06dB and -27.15dB for directional elements and isotropic elements respectively. So our algorithm is robust to the directivity of the elements.

TABLE 1. COMPARISON OF MAX SLL VALUES FOR DIFFERENT SCENARIOS WITH 200 ELEMENTS IN THE LINEAR ARRAY (THE STEERING ANGLE IS 90°).

References	**Thinning rate η**	**Max SLL(dB)**	**Improvement**	**Element pattern**
[1]	0.77	-22.09dB	4.97dB	1
[2]		-24.03dB	3.03dB	
[4]		-23.05dB	4.01dB	
[13]		-22.13dB	4.93dB	
Our algorithm		-27.06dB		
[1]	0.75	-23.69dB	3.46dB	$\|sin\theta\|$
[2]		-25.25dB	1.9dB	
Our algorithm		-27.06dB		



(a)



(b)

FIGURE 5. (a) Far-field pattern of a thinned array with 200 isotropic elements($\theta_d = 120°$)(b)convergence curves

Another advantage lies in the convergence speed. With our algorithm, a steady value can be reached after 10 iterations and the convergence occurs after 17 iterations. In [1], they need 30 iterations to reach convergence. In [12], it takes 30 iterations to achieve a steady value for the best result. Also our algorithm can help decide the thinning rate in advance in array optimization.

TABLE 2. COMPARISON OF MAX SLL WITH DIFFERENT STEERING ANGLES FOR ISOTROPIC ELEMENTS.

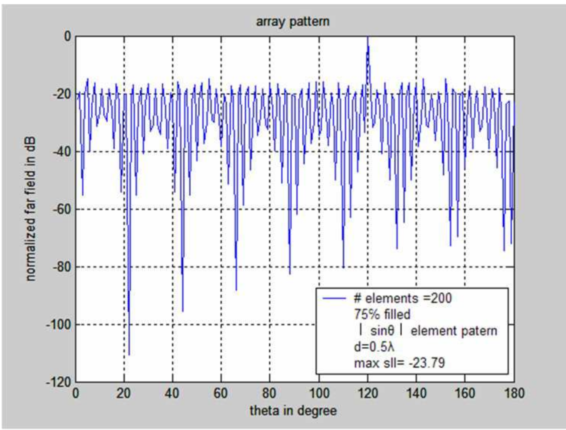
References	90°	120°	**Deterioration Rate**
[1]	-22.09dB		
[2]	-24.03dB	-23.57	19.14%
Our algorithm	-27.06dB	-26.79dB	1.00%

3.2. **Results for Various Steering Angles.** When the steering angle θ_d is 120°, we obtained the results in Figure 5 and Figure 6. In [1, 2] , they also tried with the steering angle of 120°.

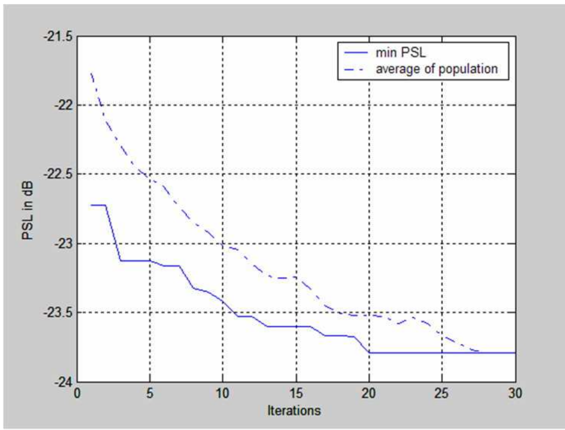
First, let us investigate the performance of our algorithm when the steering angle is $\theta_d = 120°$. For the linear array with isotropic elements, when steering angle is $\eta = 0.77$ and steering angle is $\theta_d = 120°$, the max SLL is -26.79dB, better than -23.57dB in [2] as shown in Table 2. With our algorithm,we improve max SLL by 3.22dB compared to the result in [2].

For the linear array of directional elements with the element pattern is $elpat(\theta) = |sin\theta|$, where the sparse rate is $\theta = 0.75$ and the steering angle is $\theta_d = 120°$, we obtain the max SLL -23.79dB, better than -18.75dB in [1] and -21.2dB in [2] as shown in Table 3. With our algorithm, we make improvement of 5.04dB and 2.59dB when compared to [1] and [2] respectively.

Then let us observe the change of max SLL when the steering angle is changed from 90° to 120°. According to [2], only a little change will happen for the linear array composed of isotropic elements, but max SLL will deteriorate for directional elements based linear array. All max SLL values for different methods dealing with the linear array of isotropic elements are listed in Table 2.



(a)



(b)

FIGURE 6. (a) Far-field pattern of a thinned array with 200 directional elements with $elpat(\theta) = |sin\theta|$ ($\theta_d = 120°$)(b)convergence curves

In our algorithm, max SLL changes from -27.06dB to -26.79dB for the linear array composed of isotropic element when the steering angle is changed from 90° to 120° and the change is only 0.27dB. We call the increase of max SLL as deterioration. We can calculate the deterioration rate as the increased amount of max SLL over original max SLL. Here, we find that the deterioration rate in our case is only 1%. It is much better than 19.14% in [2]. This result indicates that the steering angle has little influence on the linear array of isotropic elements.

To analyze the influence of the steering angle on the array composed of directional elements, we compare different methods with different steering angles in Table 3. In our algorithm, max SLL changes from -27.15dB to -23.79dB when the steering angle is changed from 90° to 120°. And the deterioration rate is 12.35% which is lower than 20.6% in [1] and 15.8% in [2]. This result shows that our new algorithm is more robust than the existing ones.

TABLE 3. COMPARISON OF MAX SLL WITH DIFFERENT STEERING ANGLES FOR DITECTIONAL ELEMENTS.

References	90°	120°	**Deterioration Rate**
[1]	-23.69dB	-18.75dB	20.6%
[2]	-25.25dB	-21.2dB	15.8%
Our algorithm	-27.15dB	-23.79dB	12.35%

4. **Conclusion.** This paper proposes an improved genetic algorithm based on binary coding for thinning acoustic arrays. The initial population is formed by "random place assignment" and new Cross and Mutation rules are used in GA. The algorithm can improve the performance of linear array and accelerate the convergence of optimization. This paper investigates the influence of element patterns and steering angles on max SLL for linear arrays. The results proves that our algorithm is robust to different element patterns and steering angles. Deterioration of the max SLL is lessened when steering angle is changed. Also, the fixed thinning rate and aperture before optimization make it more valuable in practice. The algorithm can be extendable to antenna arrays especially for large scale antenna arrays. In the future, we are planning to apply it to the arrays

with different types of sound resources, such as near-field acoustic arrays with simple or composite sound resources.

Acknowledgment. This work was partly supported by the National Natural Scientific Foundation of China(Grant No. 61271387 and No. 61401245), the Project of domestic visiting scholars of young teachers of higher education institutions in Shandong Province, the Zhejiang Provincial Natural Science Foundation of China (Grant No. LY15F010003 and No. LY14F020024), the Public Good Research Project of Science and Technology Program of Zhejiang Province, China. (NO.2016C31097 and and NO.2015C33230), the Science and Technology Project of QuZhou, Zhejiang, China.(NO.2015Y005), the Project for Visiting Scholars' Professional Development in Universities of Zhejiang Province, China (No.FX2014102),the Project for Young Academic Leaders' Academic Promotion in Colleges and Universities of Zhejiang Province, China(No. pd2013450), and the Natural Science Foundation of Zhejiang Province, China (No. LY15F020041).

REFERENCES

[1] R. L. Haupt, Thinned arrays using genetic algorithms, *IEEE Transactions on Antennas & Propagation*, vol.42, no.7, pp.993-999, 1994.
[2] K. S. Chen, Z. S. He and C. L. Han, Sidelobe reduction of asymmetric linear thinned arrays using genetic algorithm, *Journal of Electronics & Information Technology*, vol.29, no.4, pp.987-990, 2007.
[3] Z. M. Hamici and T. H. Ismail, Optimization of thinned arrays using stochastic immunity genetic algorithm, *International Symposium on Signal Processing and Information Technology*, pp.378-383, 2009.
[4] G. Oliveri and A. Massa, Genetic algorithm (GA)-enhanced almost difference set (ADS)-based approach for array thinning, *IET MICROW ANTENNA P*, vol.5, no.3, pp.305-315, 2011.
[5] G. Oliveri, M. Donelli and A. Massa, Linear array thinning exploiting almost difference sets, *IEEE Transactions on Antennas & Propagation*, vol.57, no.12, pp.3800-3812, 2009.
[6] C. Liu and H. Wu, Synthesis of thinned array with side lobe levels reduction using improved binary invasive weed optimization, *Progress in Electromagnetics Research M*, vol.37, pp.21-30, 2014.
[7] J. S. Roy, B. B. Mishra and A. Deb, Design of thinned planar array using genetic algorithm and hadamard matrix arrangement, *Worldairco Org*, vol.2, pp.6-9, 2014.
[8] M. M. Khodier and C. G. Christodoulou, Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization, *IEEE Transactions on Antennas & Propagation*, vol.53, no.8, pp.2674-2679, 2005.
[9] B. Goswami and D. Mandal, A genetic algorithm for the level control of nulls and side lobes in linear antenna arrays, *Journal of King Saud University - Computer and Information Sciences*, vol.25, no.25, pp.117-126, 2013.
[10] H. Wu, C. Liu and X. Xie, Thinning of concentric circular antenna arrays using improved binary invasive weed optimization algorithm, *MATH PROBL ENG*, vol.2015, pp.1-8, 2015.
[11] S. Gade, J. Hald and B. Ginn, Noise source identification with increased spatial resolution, *Sound & Vibration*, vol.131, no.4, pp.9-13, 2013.
[12] H. Li, C. Yi, Z. Lu and L. Zhou, Random acoustic array deployment based on compressed sensing, *ICIC Express Letters*, vol.9, no.9, pp.2511-2516, 2015.
[13] R. Jain and G. S. Mani, Dynamic thinning of antenna array using genetic algorithm, *Progress in Electromagnetics Research B*, vol.32, no.32, pp.1-20, 2011.