ARCHIVES OF ELECTRICAL ENGINEERING VOL. 69(3), pp. 561–579 (2020)

DOI 10.24425/aee.2020.133918

Performance investigation and element optimization of 2D array transducer using Bat Algorithm

DINA MOHAMED TANTAWY[1], MOHAMED ELADAWY[2], MOHAMED ALIMAHER HASSAN[3], ROAA MUBARAK[2]

[1]*Mechatronics Engineering and Automation Department, Faculty of Engineering*
Egyptian Chinese University
Egypt

[2]*Electronics and Communication Engineering Department, Faculty of Engineering*
Helwan University
Egypt

[3]*Department of Biomedical Engineering, Faculty of Engineering*
Helwan University
Egypt
e-mail: dina_mohamed.tantawy@hotmail.com

(Received: 11.01.2020, revised: 05.03.2020)

Abstract: One of the least expensive and safest diagnostic modalities routinely used is ultrasound imaging. An attractive development in this field is a two-dimensional (2D) matrix probe with three-dimensional (3D) imaging. The main problems to implement this probe come from a large number of elements they need to use. When the number of elements is reduced the side lobes arising from the transducer change along with the grating lobes that are linked to the periodic disposition of the elements. The grating lobes are reduced by placing the elements without any consideration of the grid. In this study, the Binary Bat Algorithm (BBA) is used to optimize the number of active elements in order to lower the side lobe level. The results are compared to other optimization methods to validate the proposed algorithm.

Key words: 2D ultrasound arrays, Binary Bat Algorithm, Genetic Algorithm, Optimization

1. Introduction

A three-dimensional (3D) region of ultrasonic imaging needs sweeping across a volume of an ultrasound beam. In most available systems, the sweeping is done by manually moving a

conventional (linear or convex) array. Another way is by using a step motor to drive the probe [1]. However, the operator's skills affect sensitivity of these techniques and they are characterized by poor time resolution. To overcome these disadvantages, 2-dimensional (2D) matrix arrays can be used. The 2D array can be electronically controlled to steer the beams in both the elevation and lateral directions [2–6]. Unfortunately, the 2D array probe design must obey the spatial sampling condition (i.e. the pitch < half the wavelength), which imposes small-sized elements. Such 2D array controlling is technically challenging. Since, it is difficult to connect and drive several hundred (up to thousands) of elements, not to say unrealistic.

Another method has been suggested to use a random sparse-array technique in order to reduce the number of elements while maintaining acceptable performance which is proposed in [7–13]. As the number of active elements is large, the beam produced suffers from more severe deteriorations. With optimization algorithms the sparse array techniques are typically combined to activate the most suitable elements.

In literature, in order to find the best trade-offs between the footprint characteristics and the probe performance several optimization algorithms [14–20] have been applied. Five trade-offs affect the performance probe, these trade-offs are the number of elements, the energy loss with respect to the reference full 2D array, the Main Lobe Width (MLW), the Side Lobes Level (SLL), and the Grating Lobe Level (GLL). Initially by disabling some elements of a full regular transducer array, sparse arrays are obtained [21–25].

According to the literature [26, 27], the Simulated Annealing (SA) Algorithm and Genetic Algorithm (GA) are the most used optimization techniques for medium and large 2D ultrasound arrays which are capable of providing excellent results in the distribution of both active elements on the arrays' surface and the beam profiles [28]. The main drawback of the SA lies in the resources it requires to run. In this work, the Binary Bat Algorithm (BBA) [29] is proposed to perform the 2D array element optimization. The BBA results are compared to Binary Differential Evolution (BDE) [30] and the GA to see the potential of the BBA in competing with the BDE and GA for medium 2D arrays in order to visualize beam profile comparison in terms of beam width transmitted energy and SSL. A full comparison is done as well as a statistical analysis to verify the efficiency of the BBA to optimize the array elements and to lower the SLL. The BBA, BDE and GA are compared in terms of element optimization, the SLL, mean and minimum values for optimized elements, array configuration and the algorithm complexity. Finally a section about choosing the tuning parameter for BBA is presented.

2. Deactivated elements effect on 2D array transducer

The beam profile affected by the number of dead elements and their placement to assess the features of the beam profile, evaluates the performance of an ultrasound transducer. A plot of the intensity or pressure of lateral distance from the transducer center is the assessment normally done. In Fig. 1, this is referred to as the main lobe, side lobe and grating lobe. Due to deactivated elements in 2D array transducer that can appear on the sides of the main lobe, they are called side lobes and can be visualized. And grating lobes are special case of side lobes. While SL is directed forward, GL is directed away from the main beam with a large angle SL and GL can give rise to artifacts.

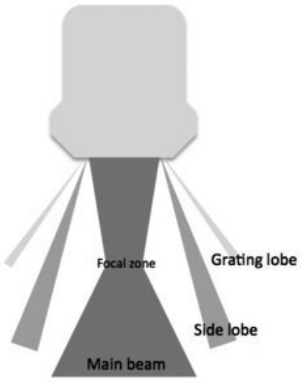

Fig. 1. Main, side and grating lobes

3. The proposed 2D array

The array contains sparsely filled $8 \times 24 = 192$ elements. In this paper the proposed array uses a pitch larger than the half wavelength. That is because, in the manufacturing of the array, this pitch guarantees a good trade-off between the GL and ML. Table 1 illustrates the parameters of the array, and Fig. 2 shows the place of each parameter of the 2D array.

Table 1. 2D array parameters

Parameters	**2D Array**
Central frequency	3.9 MHz
Wavelength	0.39 mm
Element number	$8 \times 24 = 192$
Element size (width and height)	0.37×0.37 mm^2
Pitch	0.4 mm

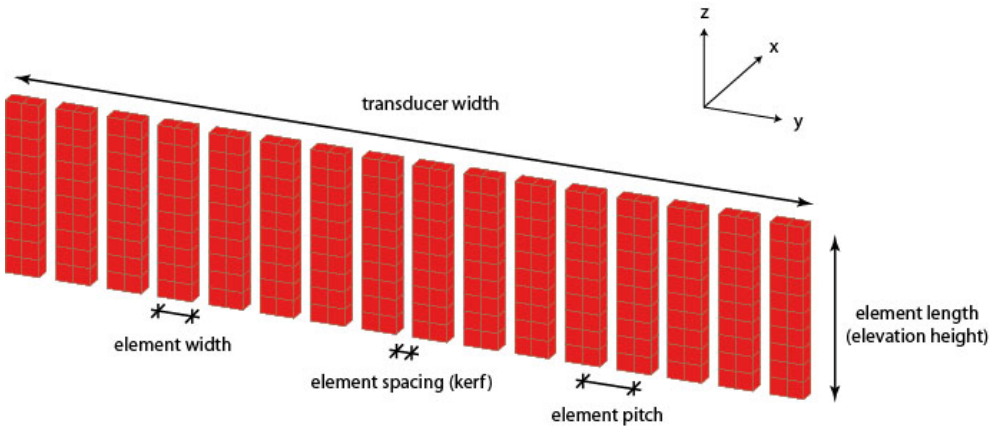

Fig. 2. Parameters of 2D array

4. Proposed optimization methods

4.1. Binary Bat Algorithm (BBA)

The echolocation behavior of bats has been inspired by the BBA. There are many types of bats in nature, they differ in weight and size. When navigating and hunting, bats have similar behaviors, regardless of their types. They use their natural sonar to navigate and hunt. Bats have two main characteristics, useful when hunting. When prey is being chased by bats, bats increase the rate of emitted ultrasonic sound and also decrease the loudness [29]. A mathematical model will be applied to this behavior.

In the BBA, during the course of iterations an artificial bat updates its velocity, position and frequency vectors. This update is done using (1), (2), and (3):

$$V_i(t+1) = V_i(t) + (X_i(t) - G_{\text{best}})F_i\,, \tag{1}$$

$$X_i(t+1) = X_i(t) + V_i(t+1), \tag{2}$$

where the best solution attained so far is G_{best} and the frequency of i-th bat is F_i which is updated over the course of iteration as illustrated in (3):

$$F_i = F_{\min} + (F_{\max} - F_{\min})\beta, \tag{3}$$

where β is a random number in the interval [0, 1]. The artificial bats diverse propensity to the best solution is encouraged by different frequencies as shown in (1) and (3). The exploitability of the BA is guaranteed by these equations. However, to perform the exploitation a random walk procedure is used as follows in (4):

$$X_{\text{new}} = X_{\text{old}} + \varepsilon A^t\,. \tag{4}$$

In this formula, ε is a random number in the interval $[-1,\ 1]$ and A is the loudness of the emitted sound that bats use to perform an exploration instead of exploitation as it is increased. It can be stated that the BBA is a balanced combination of Particle Swarm Optimization (PSO) and intensive local search. The balancing between these techniques is controlled by the loudness (A) and pulse emission rate (r). These two elements are updated according to (5) and (6):

$$A_i(t+1) = \alpha \mathrm{A}_i(t), \tag{5}$$

$$r_i(t+1) = r_i(0)[1 - \exp(-\gamma \times t)], \tag{6}$$

where α and γ are the constants and α is analogous to the cooling factor in Simulated Annealing (SA). Eventually, A_i will equal zero, while the final value of r_i is $r(0)$. Note that to guarantee that the bats are moving toward the best solutions both loudness and rate are updated when the new solutions are improved. Fig. 3 illustrates the flow chart of the BBA.

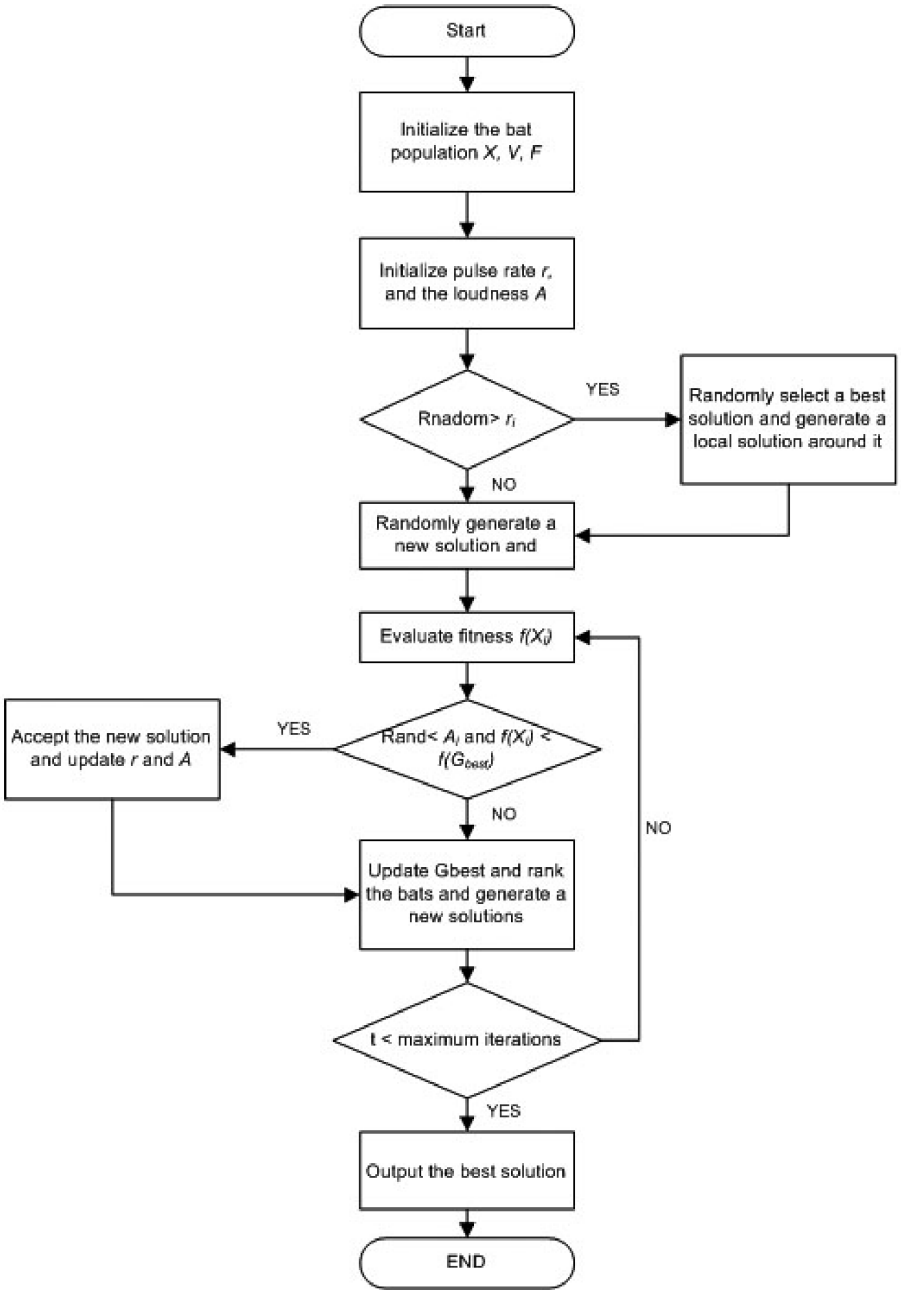

Fig. 3. Flow chart of BBA

4.2. Advantages and disadvantages of BBA

Advantages of the BBA:

– Few parameters to control (only 2 parameters),
– Fast convergence speed,
– High accuracy,
– Simple, flexible and easy to implement.

Disadvantages are as follows:

– The BBA converges faster in early stages but gets slower at its final stages,
– Algorithm parameters needs tuning for each application,
– Its not the most reliable algorithm for large dimension problems.

4.3. Genetic Algorithm (GA)

Part of the family of algorithms are genetic algorithms, which can be applied to both linear and non-linear optimization problems [28]. Two main families of the GA exist: the binary and continuous GA, depending on the type of the optimization problem a suitable family can be chosen. The common basic optimization problems use the binary GA, and the binary GA contains in coding binary variables. By using (7) and (8) the offspring of two parent arrays are obtained. Xi and Xj are two parent arrays, the two offspring are obtained through a random parameter α which belongs to the interval [1, 2]. The flow chart of the GA is illustrated in Fig. 4.

$$X_i' = \alpha X_i + (1-\alpha)X_i\,, \tag{7}$$

$$X_j' = \alpha X_j + (1-\alpha)X_j\,. \tag{8}$$

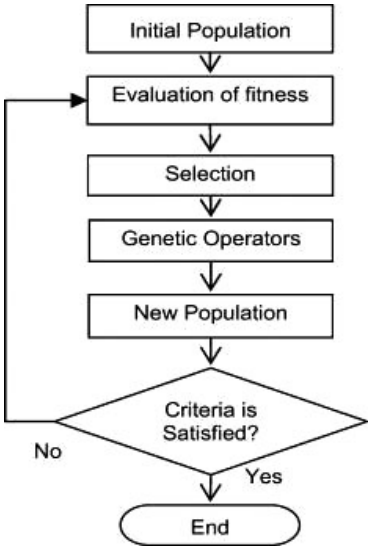

Fig. 4. Flow Chart of GA

4.4. Binary Differential Evolution (BDE) Algorithm

This section proposes a new approach to solve binary valued problems using BDE. The BDE borrows concepts from the Binary Particle Swarm Optimizer (BPSO), developed by Kennedy and Eberhart [30]. The BDE is developed for continuous-valued problems. A new non-parameter binary mutation operator is obtained. The new mutation process is subscribed for the $g-$th iteration of initial individual mutation and the individual mutation is then formulated as:

$$v_{i,j,g} = \begin{cases} x_{r0,j,g} + (-1)^{x_{ro,j,g}} \times \left| x_{xr1,j,g} - x_{r2,j,g} \right|, & \text{if} \quad \theta < 0.5 \\ x_{rb,j,g} + (-1)^{x_{rb,j,g}} \times \left| x_{xr1,j,g} - x_{r2,j,g} \right|, & \text{otherwise} \end{cases}, \tag{9}$$

where: $r0$, $r1$, $r2 \in 1, 2, \ldots, n$ and $r0 \neq r1 \neq r2$, rb is the index of the best individual in the current population. While j is the dimension of feature variables.

As presented in [30], a new daptive mechanism based on individual fitness for obtaining a reasonable factor is presented. The adaptive crossover factor $CF(Xi)$ is shown as:

$$CF(Xi) = \frac{IFF(Xi) - IEF_l(X) + \mu}{IFF_u(Xi) - IEF_l(X) + \mu}, \tag{10}$$

where the maximum and minimum fitness value of all individuals are expressed by $IEF_u(X)$ and $IEF_l(X)$, respectively. In order to prevent the crossover factor to be 0, the parameter μ is introduced and it is the absolute value of the difference between the minimum fitness value and the second minimum. Note that we select a random number from a normal distribution by $\eta = \text{rand}\, n(CF(Xi), 0:1)$. So the crossover operator is presented as follows:

$$u_{i,j,g} = \begin{cases} v_{i.j,g}, & \text{if} \quad \text{rand}_j[0,\ 1] \leq \eta \quad \text{or} \quad (j = j_{\text{rand}}) \\ x_{i.j,g}, & \text{otherwise} \end{cases}, \tag{11}$$

where j_{rand} is the dimension selected randomly from an individual. Hence, we can infer that the selection of the crossover factor is associate with the fitness value of the individual. In other words, the number of an inherit element depends on the fitness difference of the individuals. The new individual selection strategy can be presented as:

$$x_{i,j,g+1} = \begin{cases} u_{i.j,g}, & \text{if} \quad IEF(u_{i,j,g}) < IEF(x_{i,j,g}) \\ x_{i.j,g}, & \text{otherwise} \end{cases}, \tag{12}$$

where $u_{i,j,g}$ is the $i-$th offspring individual in the g-th iteration. If the fitness value of a new individual yields the corresponding parent individual, then $u_{i,j,g}$ is set to $x_{i,j,g}C1$. Otherwise the current individual will be inherited in the next generation. The flowchart of the BDE, which is used in this paper, is shown in Fig. 5.

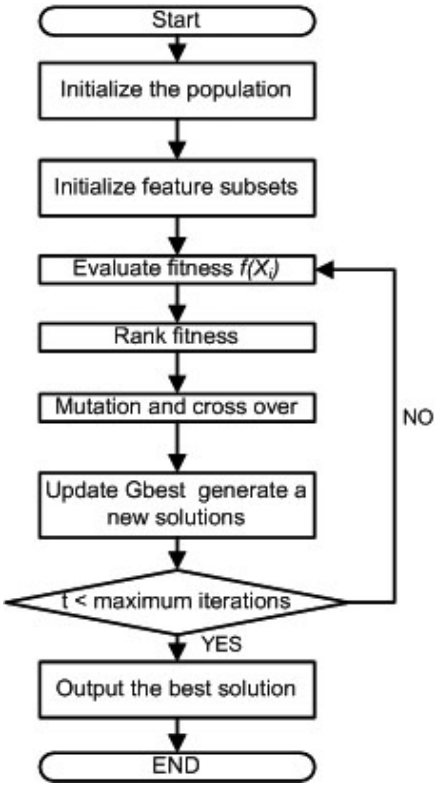

Fig. 5. BDE flowchart

5. 2D array optimization

In 2D ultrasound optimization, different configurations exist in the literature. The optimization can be realized through both the apodization and position of elements or with only the position. The population of the BBA is defined as the 2D array elements, where the BBA is used to optimize the number of active elements in the array in order to get a minimum SLL without distorting the beam width of the ML. The optimization is done using the fitness function presented in [11–20]. The fitness of the BBA corresponds to the beam profile and the constraints on the side lobes outside the main lobe region and is given by (13)

$$f(Xi) = \left(\iint\limits_{A} \left(20 \log_{10}(pa(Xi)) - SL_{dB}\right) \mathrm{d}\theta \mathrm{d}\phi \right)^2, \tag{13}$$

where $X(X_1, X_2, \ldots, X_N)$ represents a 3D matrix containing N 2D sparse arrays of N_E elements among which N_a is active, the pressure field and SL_{dB} stand for the maximum side lobes allowed in the space excluding the main lobe area.

6. Simulation results

6.1. BBA results

The BBA is applied to ten 2D arrays of 192 elements (population number is 10). Each array is filled with a random number of active elements. The positions of the random elements are then optimized to obtain best possible beam patterns. The BBA results are shown below in Fig. 6, Fig. 7 and Fig. 8.

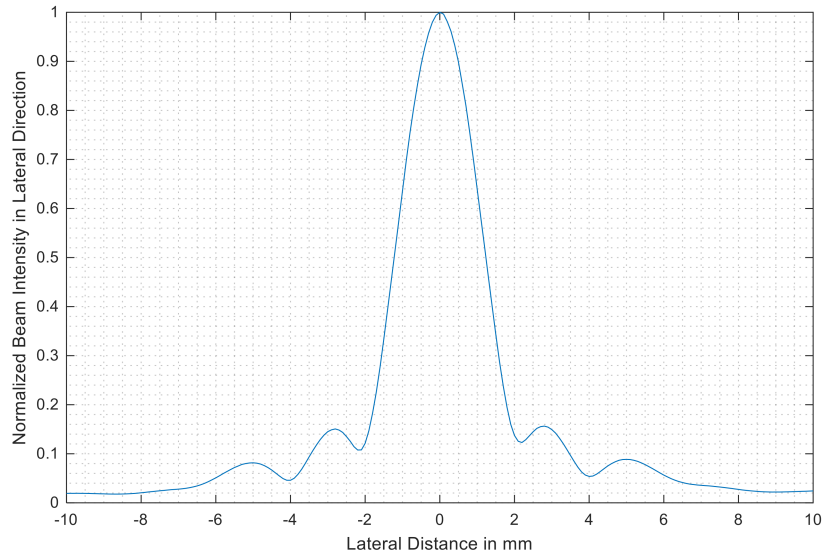

Fig. 6. Lateral direction output of BBA

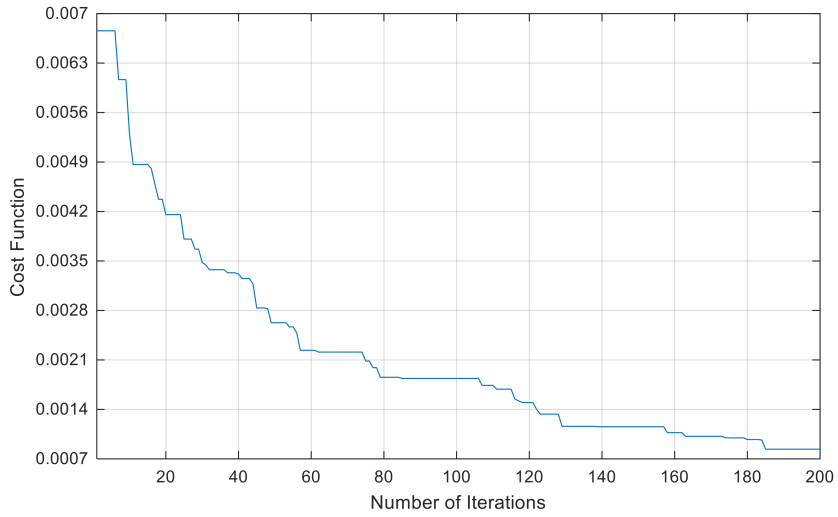

Fig. 7. Cost function of BBA

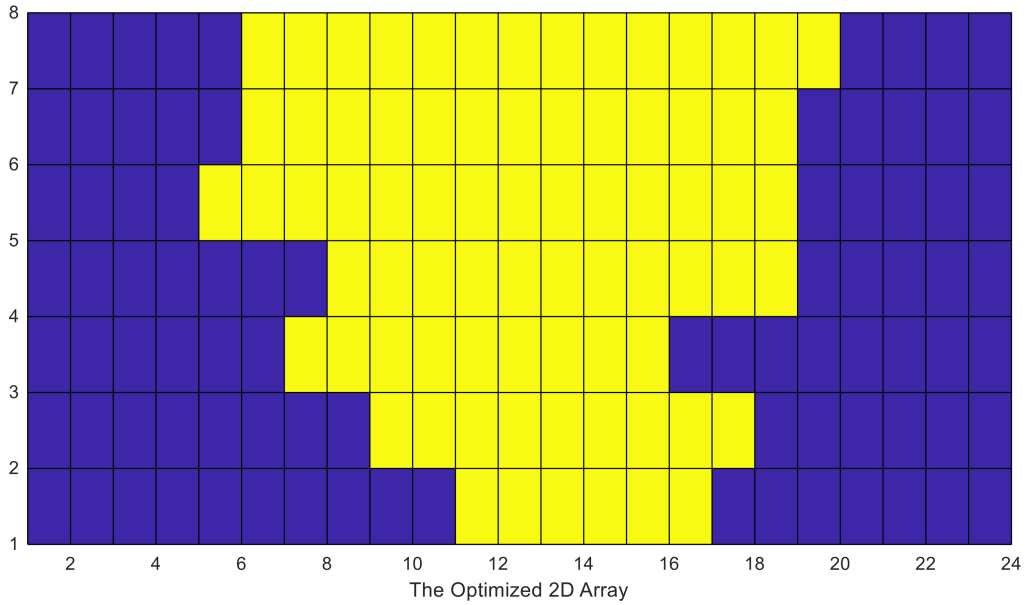

Fig. 8. Active elements (yellow) of BBA

Fig. 6 shows the normalized beam in lateral direction of the optimized 2D array using the BBA. The optimized array has 87 active elements only. Fig. 7 shows the cost function of the BBA over the iteration course. It's observed that the cost function is minimized in a range of 200 iteration, which is low. This shows that the BBA converges faster than other algorithms. The following Fig. 8 shows the optimized 2D array after the optimization of the original array by the BBA. The blue elements are the deactivated elements and the yellow elements are the active elements. It is clear that the optimized array has fewer active elements that the original array.

6.2. GA results

The GA is applied to the same 2D array as the BBA. Each array is filled with a random number of active elements. The positions of the random elements are then optimized to obtain best possible beam patterns. The GA results are shown below in Fig. 9, Fig. 10 and Fig. 11.

Fig. 9 shows the normalized beam in the lateral direction of the optimized 2D array, using the GA.

The optimized array has 95 active elements only. Fig. 10 shows the cost function of the GA over the iteration course. It's observed that the cost function is minimized in a range of 200 iteration but as observed in the BBA it converges faster in early stages but gets slower at its final stages, in the GA it converges very slowly and in all iterations. Also the GA does not reach the minimum value that the BBA reaches, the minimum value in the GA is 0.00138, the minimum value in the BBA is 0.0008, as shown in Fig. 10 and Fig. 7.

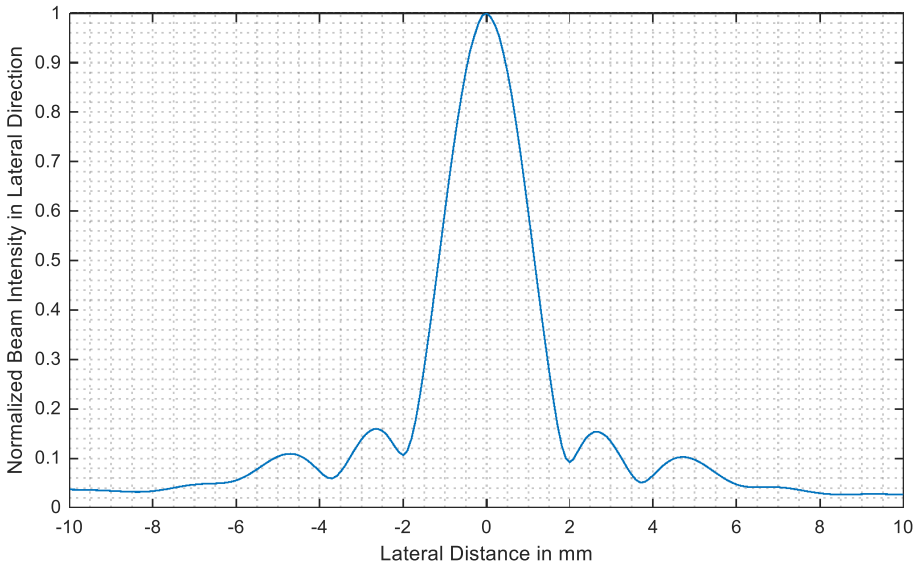

Fig. 9. Lateral direction output of GA

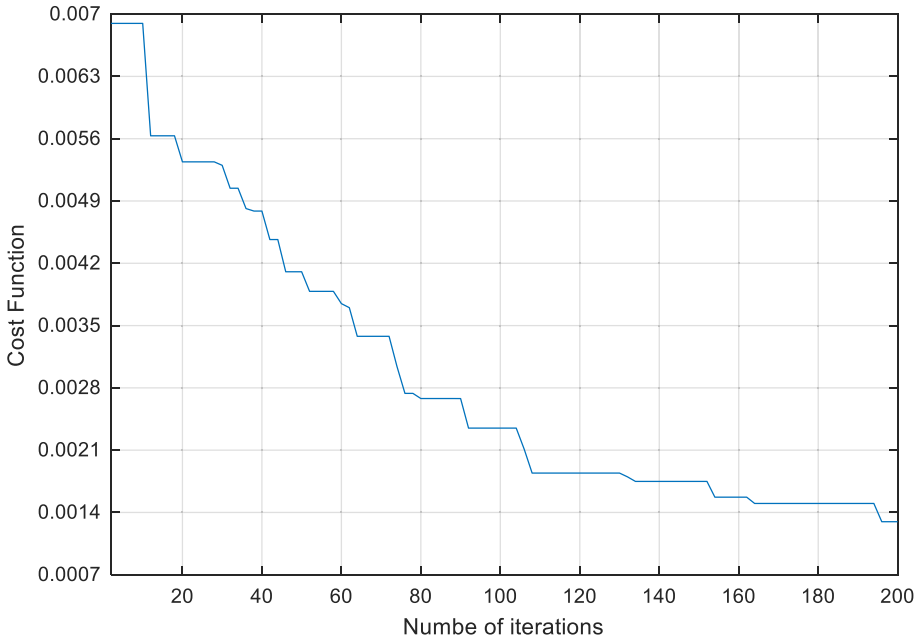

Fig. 10. Cost function of GA

Fig. 11 shows the optimized 2D array after the optimization of the original array by the GA. The blue elements are the deactivated elements and the yellow elements are the active elements. It is clear that the optimized array has fewer active elements that the original array.

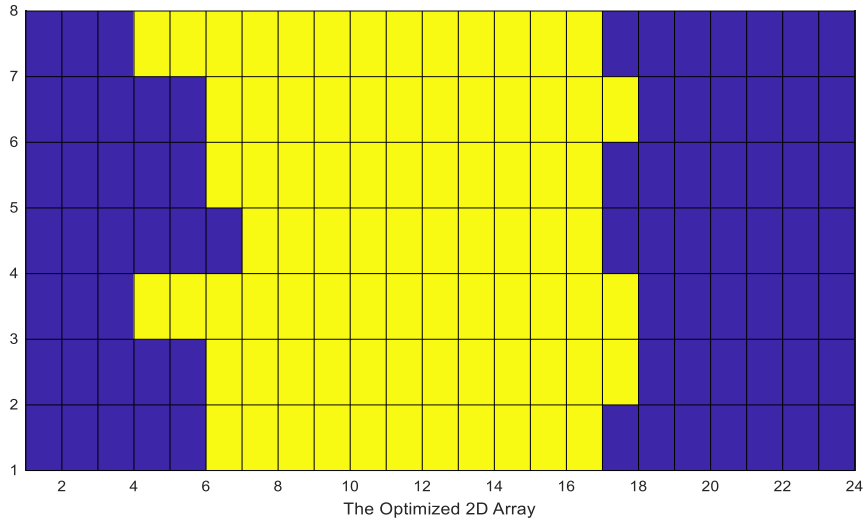

Fig. 11. Active elements (yellow) of GA

6.3. BDE results

For more validation on the efficiency of the BBA and the effectiveness of its optimization to the 2D array The BBA is compared to one of the most recent algorithms named BDE. The BDE is one of the most recent optimization algorithms for binary optimization problems. The BDE is applied to the same array and under the same conditions to differentiate between both algorithms, and to know why the BBA is suggested in this work. The BDE results are shown below in Fig. 12, Fig. 13 and Fig. 14.

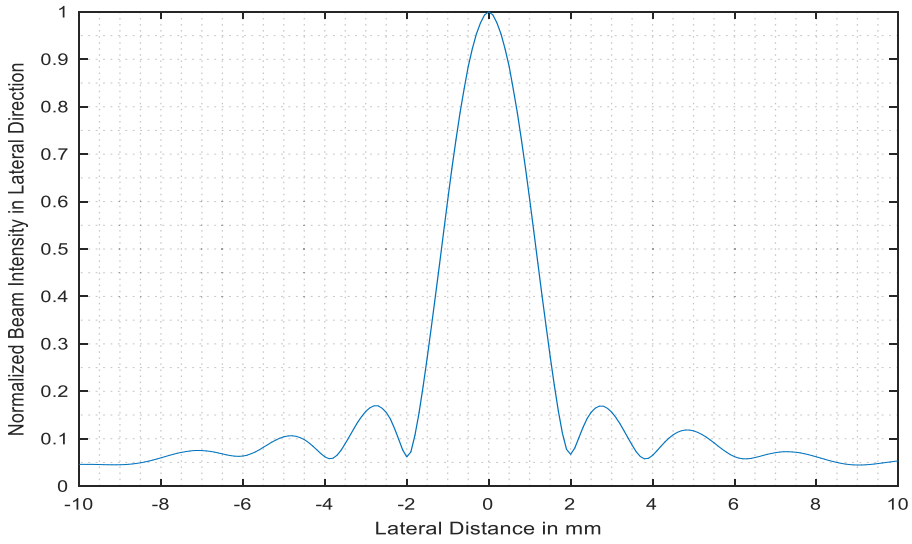

Fig. 12. Lateral direction output of BDE

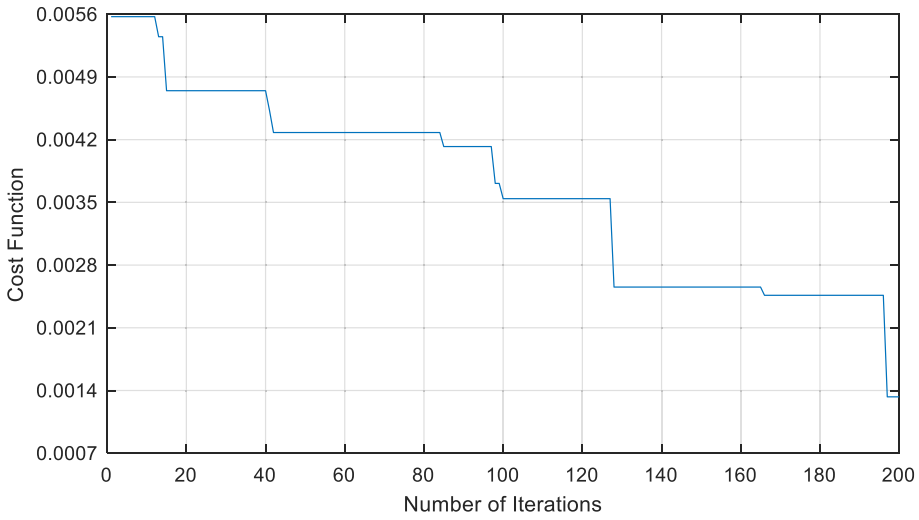

Fig. 13. Cost function of BDE

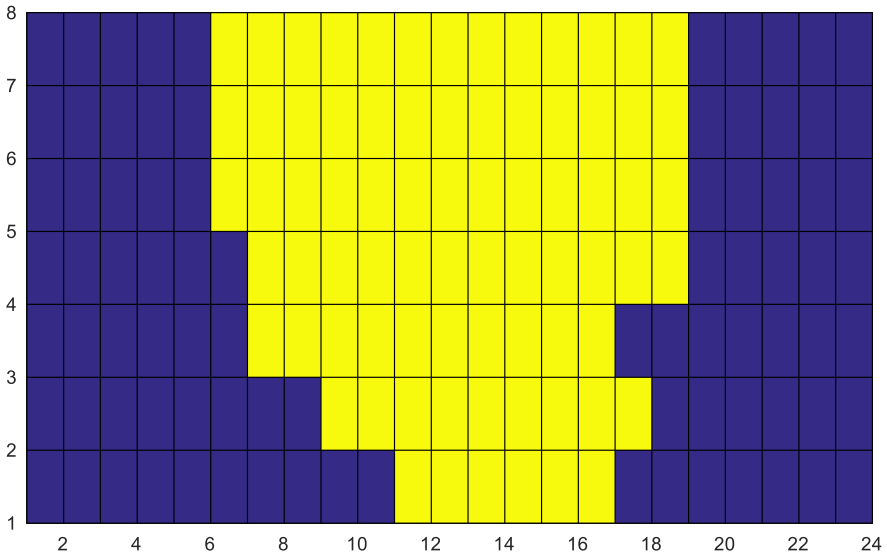

Fig. 14. Active elements (yellow) of BDE

Fig. 12 shows the normalized beam in the lateral direction of the optimized 2D array, using the BDE. It's observed that there is no significant improvement in the output, as the lowering of the elements will affect the main beam (MLL) itself not only the SL. Fig. 13 shows the cost function of the BDE over the iterations course. It's observed that the cost function is minimized in a range of 200 iterations and the BDE has a lower value than the BBA in early stages but gets slower at its final stages, also the BDE has a final value slightly higher than that of the BBA.

As observed the cost function has lower value than that of the BBA in early stages but it ends with a higher value than the BBA at the final stage. As for the number of elements as shown in Fig. 14, the BDE has 87 active elements but with a slightly different configuration than that of the BBA. It shows that the BDE and BBA reach almost the same optimum array configuration.

7. Performance and statistical analysis

From the above results it can be observed that the BBA gives a better optimized array than the GA and BDE. For more validation the beam profiles for 192 elements (original array), the BBA, GA and BDE are shown in Fig. 15.

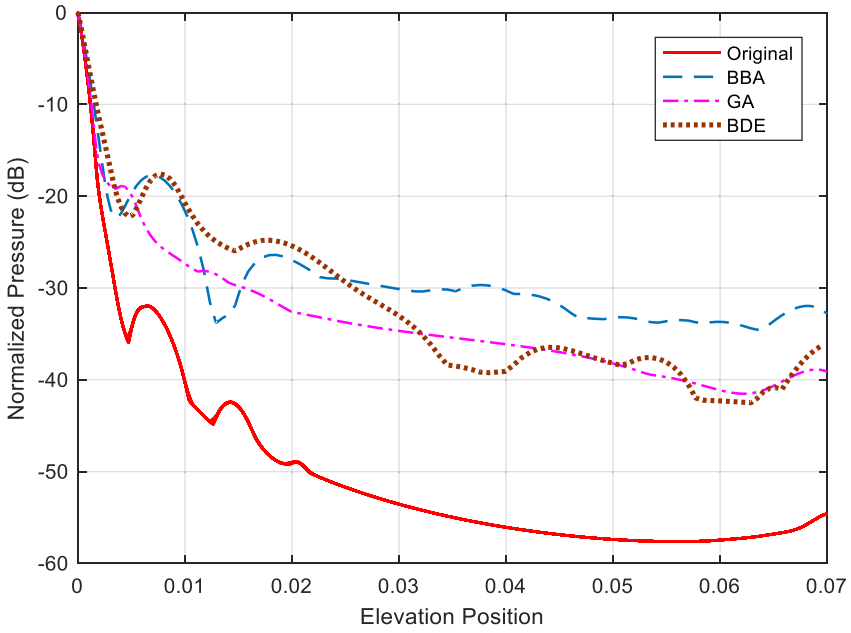

Fig. 15. Beam profile comparison

Fig. 15 illustrates the effect on the appearance of the side lobe and on the width of the main lobe that shows beam profiles for 87 elements for the BBA, 95 elements for the GA and 87 elements for the BDE. Fig. 16 illustrates the output comparison in the lateral direction. As observed, the BBA gives a better SLL output than the GA, BDE and the original array. This is due to the element optimization of the BBA which lowers the SLL while maintaining the beam profile and pressure outputs. A comprehensive statistical analysis of the three algorithms the BBA, BDE and GA are shown in Table 2.

As observed from the above figure, the BBA and BDE almost have the same SLL but due to the difference in array elements, the BBA configuration has a slightly lower SLL than that of the BDE.

Table 2. Statistical analysis of BBA, BDE and GA

Algorithm	Number of trials	Mean active elements number	Minimum active elements number	Original array
BBA	50	90	87	192
GA		100	95	
BDE		91	87	

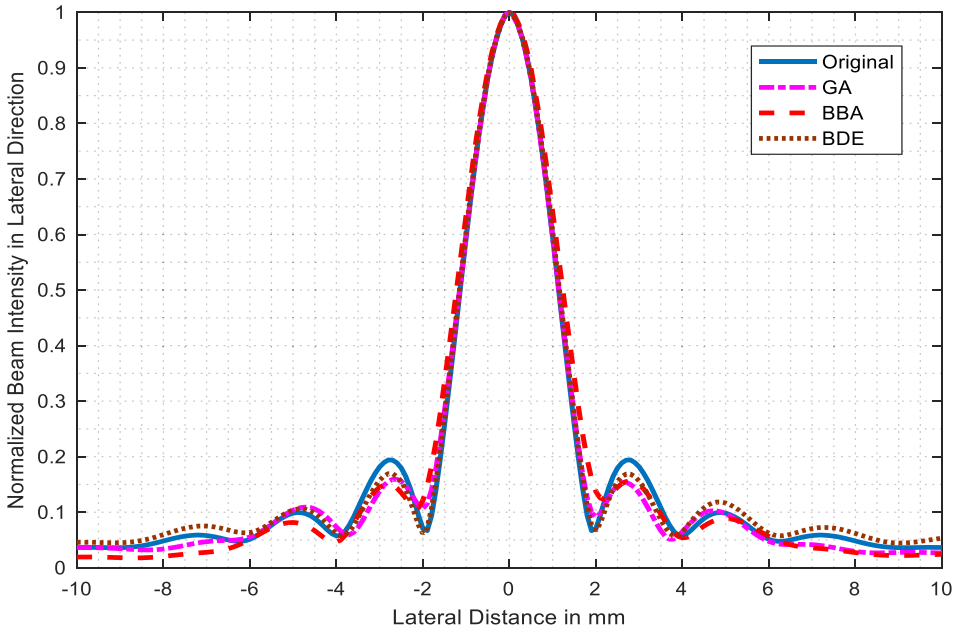

Fig. 16. Comparison of lateral direction output

From the above results it can be observed that all the algorithms and also the original array have the same width of the main beam. But in terms of the SLL, the BBA gives a lower SLL than both the BDE and GA. Also it is observed that the BDE gives a slightly higher SLL than the BBA and almost equal to that of the GA, but it gives a lower number of elements than the GA. So in terms of element optimization both the BBA and BDE were successful to minimize the number of elements, but the BDE gives a higher SLL due to the slightly different configuration of the optimized elements.

Since all the algorithms are stochastic, a statistical analysis of the BBA, GA and BDE algorithm is done by performing 50 trail runs for each algorithm. The analysis is done to get the mean and the minimum values for each algorithm in terms of element number optimization. Table 2 shows the obtained results.

In Table 2, it is observed that both the mean and the minimum number of elements for the BBA are lower than that of the BDE and GA.

8. Time complexity

In this section, a full discussion on the proposed GA, BDE and BBA has been presented. In the discussion, different perspectives were taken into consideration such as algorithm complexity, execution time, advantages, and disadvantages. Computational complexity of any algorithm can be measured using different methods. One of these methods is the big O. In optimization algorithms, the big O is used to clarify algorithms according to how their running time or space requirements grow as the input size grows. In this section complexity for the proposed GA and BBA will be discussed.

8.1. GA complexity

The complexity depends on the genetic operators, their implementation (which may have a very significant effect on overall complexity), the representation of the individuals and the population, and obviously on the fitness function. The big O representation can be obtained as:

$$\mathrm{O(GA)} = l(nm + (\mathrm{O(selection)} + \mathrm{O(mutation)} + \mathrm{O(elitism)})), \tag{14}$$

where: l is the maximum generation number, n is the number of chomsomes (population number), m is the dimension of the problem. The GA complexity depends on the selection, mutation and elitism. So by simplifying (14) we get the GA complexity as follows:

$$\mathrm{O(GA)} = l(nm + n^2). \tag{15}$$

Thus from Eq. (15), it's clear that the GA time complexity depends on the number of iterations, and the population size. As observed, the execution time increases non linearly and it is higher than other algorithms even under the same simulation conditions.

8.2. BBA complexity

The BBA complexity can be obtained as follows. The computational complexity of the BBA depends on the number of iterations, number of bats (population) and the problem dimension. Therefore, the overall time complexity is:

$$\mathrm{O(BBA)} = l(nm), \tag{16}$$

where: n is the number of bats, l is the maximum number of iterations, and m is the number of objects. It is observed that BBA complexity depends on the number of bats and the number of iterations beside the dimension, so when these numbers are large, the complexity increases and the execution time increases too. Also the complexity is unlike the GA, which increases non linearly.

8.3. BDE complexity

The computational complexity of the BDE depends on the number of iterations, number of population and the problem dimension. Therefore, the overall time complexity can be obtained as stated in [30]:

$$\mathrm{O(BDE)} = n(3 + 2l + 2lm), \tag{17}$$

where: n is the population size, l is the maximum number of iterations, and m is the problem dimension. It is observed that the BDE complexity depends on the number of vectors and the number of iterations beside the dimension, so when these numbers are large, the complexity increases and the execution time increases much more than that of the BBA. Table 3 shows the effect of the increased number of bats on the execution time and the variation in chromosomes on the execution time of the GA, as well as increasing number of vectors in the BDE. The execution time is calculated using MATLAB. The iteration number is 200 for all trials and the population size is the same in all algorithms just to insure a fair comparison.

Table 3. Execution time comparison

Number of bats (n)	**Execution time of BBA**	**Number vectors**	**Execution time of BDE**	**Number of chromosomes**	**Execution time of GA**
10	1.423 s	10	2.212 s	10	3.241 s
15	1.627 s	15	4.384 s	15	4.752 s
20	2.343 s	20	5.781 s	20	7.378 s

It's obvious that the BBA is much faster than the GA and BDE. Also, as number of population increases, the GA and BDE algorithm times increase much more than that of the BBA. So, in this work the suggestion of using the BBA is illustrated from the point of view of the execution time, better element number and lower SLL.

9. Parameters selection of BBA

Based on the basic principle of the BBA the pulse emission rate (r) has an effect on the convergence precision. Loudness (A) effects the convergence rate but each function requires different loudness (A). Therefore, for different functions, the simulation experiments should be carried out in order to obtain the appropriate parameter setting. The simulation results show that the convergence speed of the algorithm is relatively sensitive to the setting of the algorithm parameters. So, by simulation and testing it is found that the best setting for the pulse emission rate (r) and loudness (A) is 0.2 and 0.7, respectively. These settings give the best results in terms of fast and precise convergence of the BBA towards the best solution for this application (2D array optimization).

10. Conclusion

The optimization of a 2D ultrasound array is a solution to make large arrays usable on contemporary scanners with acceptable image features. This work evaluates to what extent the BBA could replace the GA and BDE in the problem of optimizing 2D ultrasound arrays, enabling fast and reliable results with limited resources. The results obtained confirm the suitability of the

BBA to be used for optimization problems in small arrays. The BBA, BDE and GA are compared from the point of view of the number of active elements, the SLL and algorithm complexity. The results show that the BBA outruns the BDE and GA in terms of the number of active elements, the SLL, time complexity and array configuration. Moreover, a statistical analysis shows that the BBA has lower average and minimum values for the number of elements for the optimized array. So, the suggestion that the BBA is more effective in solving optimization problems of small arrays has been confirmed.

References

[1] Fenster A., Downey D.B., Cardinal H.N., *Three-dimensional ultrasound imaging*, Phys. Med. Biol., vol. 46, pp. 67–99 (2001).

[2] Davidsen R.E., Smith S.W., *A multiplexed two-dimensional array for real time volumetric and B-mode imaging*, IEEE Ultrasonics Symposium, Proceedings, San Antonio, TX, USA, pp. 1523–1526 (1996).

[3] Chi Hyung Seo, Yen J.T., *A 256 × 256 2-D array transducer with row-column addressing for 3-D rectilinear imaging*, IEEE Transactions on Ultrasonics Ferroelectronics and Frequency Control, vol. 56, pp. 837–847 (2009).

[4] Chen P., Shen B., Zhou L., Chen Y., *Optimized simulated annealing algorithm for thinning and weighting large planar arrays*, Journal of Zhejiang University Science C, vol. 11, pp. 261–269 (2010).

[5] Trucco A., *Thinning and weighting of large planar arrays by simulated annealing*, IEEE Transactions on Ultrasonics Ferroelectronics and Frequency Control, vol. 46, pp. 347–355 (1999).

[6] Diarra B., Robini M., Tortoli P., Cachard C., Liebgott H., *Design of optimal 2-D non-grid sparse arrays for medical ultrasound*, IEEE Transactions on Biomedical Engineering, no. 99 (2013).

[7] Bavaro V., Caliano G., Pappalardo M., *Element shape design of 2-D CMUT arrays for reducing grating lobes*, IEEE Transactions on Ultrasonics Ferroelectronics and Frequency Control, vol. 55, no. 2, pp. 308–318 (2008).

[8] Bhuyan A. *et al.*, *Integrated Circuits for Volumetric Ultrasound Imaging with 2-D CMUT Arrays*, IEEE Transactions on Biomedical Circuits and Systems, vol. 7, no. 6, pp. 796–804 (2013).

[9] Matrone G., Savoia A., Terenzi M., Caliano G., Quaglia F., Magenes G., *A volumetric CMUT-based ultrasound imaging system simulator with integrated reception and 3-beamforming electronics models*, IEEE Transactions on Ultrasonics Ferroelectronics and Frequency Control, vol. 61, no. 5, pp. 792–804 (2014).

[10] Savoia A.S., Scaglione G., Caliano G., Mazzanti A., Sautto M., Quaglia F., *Second-harmonic reduction in CMUTs using unipolar pulsers*, in 2015 IEEE International Ultrasonics Symposium (IUS), pp. 1–4 (2015).

[11] Diarra B. *et al.*, *Comparison of different optimized irregular sparse 2D ultrasound arrays*, IEEE International Ultrasonics Symposium (IUS), Tours, France, 2016, pp. 1–4 (2016).

[12] Diarra B., Liebgott H., Robini M., Tortoli P., Cachard C., *Novel strategies in 2D sparse arrays for 3D ultrasound imaging*, European Journal of Medical Physical, vol. 32, no. 2, pp. 420–421 (2016).

[13] Roux E., Ramalli A., Liebgott H., Cachard C., Robini M.C., Tortoli P., *Wideband 2-D Array Design Optimization With Fabrication Constraints for 3-D US Imaging*, IEEE Transactions on Ultrasonics Ferroelectronics and Frequency Control, vol. 64, no. 1, pp. 108–125 (2017).

[14] Diarra B., Robini M., Liebgott H., Cachard C., Tortoli P., *Variable-size elements in 2D sparse arrays for 3D medical ultrasound*, IEEE International Ultrasonics Symposium (IUS), Prague, pp. 508–511 (2013).

[15] Roux E., Varray F., Petrusca L., Cachard C., Tortoli P., Liebgott H., *Experimental 3-D Ultrasound Imaging with 2-D Sparse Arrays using Focused and Diverging Waves*, Scientific Reports, vol. 8, no. 1, p. 9108 (2018).

[16] Haupt R.L., *Optimized Weighting of Uniform Subarrays of Unequal Sizes*, IEEE Transactions on Antennas and Propagation, vol. 55, no. 4, pp. 1207–1210 (2007).

[17] Kesong Chen, Yun Xiaohua, He Zisku, Han Chunlin, *Synthesis of Sparse Planar Arrays Using Modified Real Genetic Algorithm*, Transactions on Antennas and Propagation, vol. 55, no. 4, pp. 1067–1073 (2007).

[18] Jensen J.A., *FIELD: A Program for Simulating Ultrasound Systems*, 10th Nordic Baltic Conference on Biomedical Imaging, vol. 4, pp. 351–353 (1996).

[19] Jensen J.A., Svendsen N.B., *Calculation of pressure fields from arbitrarily shaped, apodized, and excited ultrasound transducers*, IEEE Transactions on Ultrasonics Ferroelectronics and Frequency Control, vol. 39, pp. 262–267 (1992).

[20] Tortoli P., Bassi L., Boni E., Dallai A., Guidi F., Ricci S., *ULA-OP: an advanced open platform for ultrasound research*, IEEE Transactions on Ultrasonics Ferroelectronics and Frequency Control, vol. 56, pp. 2207–2216 (2009).

[21] Anderson-Cook C.M., *Practical Genetic Algorithms (2nd ed.): Randy L. Haupt and Sue Ellen Haupt*, Journal of the American Statistical Association, vol. 100, pp. 1099–1099 (2005).

[22] Pandey H.M., *Performance Evaluation of Selection Methods of Genetic Algorithm and Network Security Concerns*, Procedia Computer Science, vol. 78, pp. 13–18 (2016).

[23] Blickle T., Thiele L., *A Comparison of Selection Schemes Used in Evolutionary Algorithms*, Evolutionary Computation, vol. 4, no. 4, pp. 361–394 (1996).

[24] Kirkpatrick S., Gelati C.D., Vecchi M.P., *Optimization by simulated annealing*, Science, vol. 220, pp. 671–680 (1983).

[25] Ezhilarasi M., Rajaram M., Sivanandham S.N., *Fractal Geometry in 2D Matrix Array for Real time 3D Ultrasound Imaging*, Calicut Medical Journal 2008, vol. 6 (2008).

[26] Turnbull D.H., Foster F.S., *Beam steering with pulsed two dimensional transducer arrays*, IEEE Transactions on Ultrasonic, Ferroelectrics and Frequency Control, vol. 38, no. 4, pp. 320–333 (1991).

[27] Cardone G., Cincotti G., Gori P., Pappalardo M., *Optimization of wide-band linear arrays*, IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control, vol. 48, no. 4, pp. 943–952 (2001).

[28] Diarra B. *et al.*, *Feasibility of Genetic Algorithms in 2D Ultrasound Array Optimization*, IEEE International Ultrasonics Symposium (IUS), Kobe, pp. 1–9 (2018).

[29] Mirjalili S., Mirjalili S.M., Yang X., *Binary bat algorithm*, Neural Computing and Applications, vol. 25, pp. 663–681 (2014).

[30] Li T., Dong H., Sun J., *Binary Differential Evolution Based on Individual Entropy for Feature Subset Optimization*, IEEE Access, vol. 7, pp. 24109–24121 (2019), DOI: 10.1109/ACCESS.2019.2900078.