

Research Article

Design of Large Thinned Arrays Using Different Biogeography-Based Optimization Migration Models

Sotirios K. Goudos¹ and John N. Sahalos²

¹*Department of Physics, Aristotle University of Thessaloniki, Thessaloniki, Greece*

²*Department of Electrical and Computer Engineering, University of Nicosia, Nicosia, Cyprus*

Correspondence should be addressed to Sotirios K. Goudos; sgoudo@physics.auth.gr

Received 6 May 2016; Revised 29 July 2016; Accepted 10 August 2016

Academic Editor: Ikmo Park

Copyright © 2016 S. K. Goudos and J. N. Sahalos. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Array thinning is a common discrete-valued combinatorial optimization problem. Evolutionary algorithms are suitable techniques for above-mentioned problem. Biogeography-Based Optimization (BBO), which is inspired by the science of biogeography, is a stochastic population-based evolutionary algorithm (EA). The original BBO uses a linear migration model to describe how species migrate from one island to another. Other nonlinear migration models have been proposed in the literature. In this paper, we apply BBO with four different migration models to five different large array design cases. Additionally we compare results with five other popular algorithms. The problems dimensions range from 150 to 300. The results show that BBO with sinusoidal migration model generally performs better than the other algorithms. However, these results are considered to be indicative and do not generally apply to all optimization problems in antenna design.

1. Introduction

Array thinning is a common discrete-valued combinatorial optimization problem. By array thinning we mean the removal (turning “off”) of some radiating elements from a periodic array antenna in order to create an array with lower sidelobe level than that with uniform excitation. The elements connected to the feed network are turned “on,” while the turned “off” elements are connected to a matched load. Array thinning results in reduction in cost and weight. An exhaustive search for all possible combinations would result in the best array design; however the computational cost will increase exponentially as the array size increases. Therefore, array thinning can be categorized as discrete, combinatorial N-P complete optimization problem [1]. EAs are suitable techniques for solving the array thinning problem. The problem of array thinning has been addressed in the literature using different EAs like genetic algorithms (GAs) [2], Particle Swarm Optimization (PSO) [3], and Ant Colony Optimization (ACO) [4]. The original PSO is inherently real-valued and operates only in continuous spaces. In order to solve binary-coded combinatorial optimization problems with

PSO several binary versions of this algorithm have been proposed. Binary PSO (BPSO) proposed by Kennedy and Eberhart extends the original PSO algorithm using a sigmoid function (or S-shaped transfer function) to map real numbers to bits [5]. In [6] a new BPSO that uses V-shaped transfer function is proposed (VBPSO). The results from [6] show that VBPSO outperforms the original BPSO and other BPSO variants with other transfer functions. Differential evolution (DE) [7, 8] is a population-based stochastic global optimization algorithm, which has been used in several real world engineering problems. Several DE variants or strategies exist. Oppositional differential evolution (ODE) [9] is a DE variant based on opposition-based learning (OBL) [10] concepts. Harmony Search (HS) [11] is an evolutionary algorithm which is inspired by the way that musicians experiment and change the pitches of their instruments to improvise better harmonies. HS has been applied successfully to antenna array synthesis problems [12, 13]. Ant Colony Optimization (ACO) is a population-based metaheuristic introduced by Dorigo et al. [14] and inspired by the behavior of real ants. The authors in [4, 15] have used ACO for thinned array design.

Biogeography-Based Optimization (BBO) [16] is an EA based on mathematical models that describe how species migrate from one island to another, how new species arise, and how species become extinct. The way the problem solution is found is analogous to nature's way of distributing species. Ma in [17] showed that sinusoidal migration models generally outperform linear migration models like the one in original BBO algorithm. The authors in [18] extend the migration model performance analysis and they propose two nonlinear migration models, which they call model 7 and model 8. BBO has been applied to design problems in electromagnetics, including Yagi-Uda synthesis [19], microstrip antenna design [20, 21], and antenna array synthesis [22–24].

The purpose of this paper is to use BBO with different migration models in order to design large thinned arrays. To the best of our knowledge this is the first time that BBO with other than the linear migration mode is applied to an antenna design problem. This paper contributes to compare performance of several state-of-the-art EAs to high dimensional thinned array design problems. Additionally the paper contributes so that the design cases presented in this paper could also be used as a framework of benchmark functions for testing evolutionary algorithms to large array design problems. More specifically, we compare the original BBO linear migration model with the sinusoidal model [17], model 7, and model 8 [18], two BPSO variants, binary HS, ODE, and ACO. We apply the algorithms to five different design cases of linear and planar arrays. A comparative study of BBO variants performance on benchmark functions is also given. Moreover, a study of the influence on the BBO performance of the boundary constraint handling method is also presented. To the best of our knowledge this is the first time that such study is carried out on BBO in general. Numerical results show that BBO with sinusoidal migration model generally performs better than or equally with the other migration models. Additionally it performs better than the original BBO algorithm in terms of solution accuracy. Additionally, results show VBPSO outperforms the initial BPSO algorithm; however its performance is inferior to that of the BBO algorithms.

This paper is organized as follows. We describe the problem formulation in Section 2. A brief description of the EAs used in this paper is given in Section 3. In Section 4 we present the numerical results. Finally, the conclusion is given in Section 5.

2. Formulation

We consider an N -element linear array of isotropic elements. The array factor is expressed as

$$\text{AF}(u, \bar{I}) = \sum_{n=1}^N I_n e^{j(2\pi/\lambda)ndu}, \quad (1)$$

where λ is the wavelength, I_n is the complex excitation of the n th element, \bar{I} is the corresponding vector of element amplitudes, d is distance between two adjacent elements, $u = \sin(\vartheta)$ is the direction cosine, and ϑ is the steering angle measured from broadside of the array. In the thinned array case I_n could be only 1 (turned “on”) or 0 (turned “off”).

For a symmetrically excited array (1) becomes

$$\text{AF}(u, \bar{I}) = 2 \sum_{n=1}^M I_n \cos \left[\frac{2\pi}{\lambda} dnu \right] + I_0, \quad (2)$$

where $M = \lfloor N/2 \rfloor$ is the largest integer less than or equal to $N/2$. For an even number of elements $I_0 = 0$, and for an odd number of elements we set $I_0 = 1$. We assume that $d = 0.5\lambda$ for all cases. The fill factor percentage is the percentage of the ratio of the turned “on” elements to the total number of elements. The optimization goal is the peak sidelobe level (SLL) suppression by finding the optimum element amplitudes. This design problem is therefore defined by the minimization of the objective function:

$$F_1(\bar{u}, \bar{I}) = \max_{u \in S_u} \{ \text{AF}_{\text{dB}}(\bar{u}, \bar{I}) \}, \quad (3)$$

where S_u is the set of direction cosines that are outside the angular range of the main lobe.

Additionally, we study the effect of maintaining the same aperture length as the original uniform array for this case. We therefore force the first and the last element to be one (turned on).

To test the algorithms ability to design planar thinned arrays we consider a $2N \times 2M$ planar array, which lies on the x - y plane. The array factor of such an array is given by

$$\begin{aligned} \text{AF}(u, v, \bar{I}) = & 4 \sum_{m=1}^M \sum_{n=1}^N I_{nm} \cos \left[\pi (2m-1) d_y u \right] \\ & \cdot \cos \left[\pi (2n-1) d_x v \right], \end{aligned} \quad (4)$$

where d_x, d_y are the distance between two adjunct elements in the x and y direction, respectively, and $u = \sin \vartheta \cos \phi$, $v = \sin \vartheta \sin \phi$ are the direction cosines.

The optimization goal here is the sidelobe level (SLL) suppression at two different phi-planes ($\phi = 0^\circ$, $\phi = 90^\circ$) for a desired fill factor percentage. This design problem is expressed by the minimization of the objective function:

$$F_2(\bar{I}) = \max \left\{ \text{PSLL}_{\text{dB}}^{\phi=0^\circ}, \text{PSLL}_{\text{dB}}^{\phi=90^\circ} \right\}, \quad (5)$$

where $\text{PSLL}_{\text{dB}}^{\phi=0^\circ}$, $\text{PSLL}_{\text{dB}}^{\phi=90^\circ}$ are the calculated peak sidelobe levels at $\phi = 0^\circ$ and $\phi = 90^\circ$ planes, respectively.

3. Optimization Algorithms

3.1. Binary PSO. The binary PSO (BPSO) model was presented by Kennedy and Eberhart and is based on a very simple modification of the real-valued PSO [5]. In binary PSO, the particle positions belong to a D -dimensional binary space (bit strings of length D), while the particle velocities remain real-valued.

The n th coordinate of each particle's position is a bit, whose state is given by

$$x_{G+1,ni} = \begin{cases} 1 & \text{if } \text{rand}_{[0,1]} < T(u_{G+1,ni}) \\ 0 & \text{otherwise,} \end{cases} \quad (6)$$

where $\text{rand}_{[0,1]}$ is a uniformly distributed random number in $[0, 1]$ and $T(x)$ is a sigmoid limiting transformation that maps real numbers to the interval $[0, 1]$. Such a function is defined by

$$T(x) = \frac{1}{1 + e^{-x}} \quad (7)$$

which defines S-shaped transfer function.

In [6] new BPSO variants that use V-shaped transfer functions are proposed (VBPSO). The BPSO variant that produces the best results according to [6] is that with a transfer function expressed by

$$T(x) = \left\lfloor \frac{2}{\pi} \arctan\left(\frac{\pi}{2}x\right) \right\rfloor. \quad (8)$$

3.2. Binary Harmony Search Algorithm. Harmony Search (HS) [11] is an evolutionary algorithm which is inspired by the way that musicians experiment and change the pitches of their instruments to improvise better harmonies. The main control parameter in HS is the Harmony Memory Size (HMS), which determines the number of solutions (harmonies) inside the HM, and it is equivalent to population size in another algorithms. Another HS parameter is the Harmony Memory Consideration Rate (HMCR), which determines whether pitches (decision variables) should be selected from the HM or randomly from the predefined range. HMCR is a number between 0 and 1. An additional control parameter of HS is the Pitch Adjusting Rate (PAR), which determines the probability of adjusting the original value of the selected pitches from the HM.

3.3. Oppositional Differential Evolution. Differential evolution (DE) [7, 8] is a population-based stochastic global optimization algorithm, which has been used in several real world engineering problems. Several DE variants or strategies exist. One of the DE advantages is that very few control parameters have to be adjusted in each algorithm run. In [9] a new DE algorithm based on opposition-based learning (OBL) [10], oppositional differential evolution (ODE), was introduced. The basic idea of OBL is not only to calculate the fitness of the current individual but also to calculate the fitness of the opposite individual. Then the algorithm selects the individual with the lower (higher) fitness value. The benefits of using such a technique are that convergence speed may be faster and better approximation of the global optimum can be found. ODE uses an additional control parameter to those of standard DE called the jumping rate $j_r \in [0, 1]$, which controls in each generation if the opposite population is created or not.

3.4. Ant Colony Optimization. Ant Colony Optimization (ACO) [14, 25, 26] is a meta-heuristic inspired by the ants' foraging behavior. At the core of this behavior is the indirect communication between the ants by means of chemical pheromone trails, which enables them to find short paths between their nest and food sources. Ants can sense pheromone. When they decide to follow a path, they tend to choose the ones with strong pheromone intensities way back to the nest or to the food source. Therefore, shorter paths

would accumulate more pheromone than longer ones. This feature of real ant colonies is exploited in ACO algorithms in order to solve combinatorial optimization problems considered to be NP-Hard.

3.5. Biogeography-Based Optimization. The mathematical models of biogeography are based on the work of Robert MacArthur and Edward Wilson in the early 1960s. Using this model, it was possible to predict the number of species in a habitat. The habitat is an area that is geographically isolated from other habitats. The geographical areas that are well suited as residences for biological species are said to have a high habitat suitability index (HSI). Therefore, every habitat is characterized by the HSI which depends on factors like rainfall, diversity of vegetation, diversity of topographic features, land area, and temperature. Each of the features that characterize habitability is known as suitability index variables (SIVs). The SIVs are independent variables while HSI is the dependent variable.

Therefore, a solution to a D -dimensional problem can be represented as a vector of SIV variables $[SIV_1, SIV_2, \dots, SIV_D]$, which is a habitat or island. The value of HSI of a habitat is the value of the objective function that corresponds to that solution and it is found by

$$\text{HSI} = F(\text{habitat}) = F(SIV_1, SIV_2, \dots, SIV_D). \quad (9)$$

Habitats with a high HSI are good solutions of the objective function, while poor solutions are those habitats with a low HSI. The immigration and emigration rates are functions of the rank of the given candidate solution. The rank of the given candidate solution represents the number of species in a habitat. These are given by the following:

(1) Linear migration model (original BBO):

$$\begin{aligned} \mu_k &= E\left(\frac{k}{S_{\max}}\right), \\ \lambda_k &= I\left(1 - \frac{k}{S_{\max}}\right). \end{aligned} \quad (10)$$

(2) Sinusoidal migration model [17]:

$$\begin{aligned} \mu_k &= \frac{E}{2} \left(-\cos\left(\frac{k\pi}{S_{\max}}\right) + 1 \right), \\ \lambda_k &= \frac{I}{2} \left(\cos\left(\frac{k\pi}{S_{\max}}\right) + 1 \right). \end{aligned} \quad (11)$$

(3) Model 7 [18]:

$$\begin{aligned} \mu_k &= E\left(\frac{k}{S_{\max}}\right)^4, \\ \lambda_k &= \frac{I}{2} \left(\cos\left(\frac{k\pi}{S_{\max}}\right) + 1 \right). \end{aligned} \quad (12)$$

(4) Model 8 [18]:

$$\begin{aligned}\mu_k &= E \left(\frac{k}{S_{\max}} \right)^{16}, \\ \lambda_k &= \frac{I}{2} \left(\cos \left(\frac{k\pi}{S_{\max}} \right) + 1 \right).\end{aligned}\quad (13)$$

I is the maximum possible immigration rate, E is the maximum possible emigration rate, k is the rank of the given candidate solution, and S_{\max} is the maximum number of species (e.g., population size). The rank of the given candidate solution or the number of species is obtained by sorting the solutions from most fit to least fit, according to the HSI value (e.g., fitness). BBO uses both mutation and migration operators. The application of these operators to each SIV in each solution is decided probabilistically. The mutation rate m of a possible solution S is defined to be inversely proportional to the solution probability and it is given by

$$m(S) = m_{\max} \left(\frac{1 - P_s}{P_{\max}} \right), \quad (14)$$

where P_s is the probability that a habitat contains S species and m_{\max} is a user-defined parameter. As with other evolutionary algorithms, BBO also incorporates elitism. This is implemented with a user-selected elitism parameter p . This means that the p best phase vectors remain from one generation to the other. The BBO algorithm is outlined below:

- (1) Initialize the BBO control parameters. Map the problem solutions to habitats (vectors). Set the habitat modification probability P_{mod} , the maximum immigration rate I , the maximum emigration rate E , the maximum migration rate m_{\max} , and the elitism parameter p (if elitism is desired).
- (2) Initialize a random population of N_p habitats from a uniform distribution. Set the number of generations G to one.
- (3) Evaluate objective function values for each antenna array of the population.
- (4) Map the objective function value to the number of species S , the immigration rate λ_k , and the emigration rate μ_k for each solution (antenna array) of the population.
- (5) Apply the migration operator for each nonelite habitat based on immigration and emigration rates with the one of the linear, sinusoidal, or nonlinear migration models.
- (6) Update the species count probability using

$$\begin{aligned}\dot{P}_s &= \begin{cases} -(\lambda_s + \mu_s)P_s + \mu_{s+1}P_{s+1} & S = 0 \\ -(\lambda_s + \mu_s)P_s + \mu_{s+1}P_{s+1} + \lambda_{s-1}P_{s-1} & 1 \leq S \leq S_{\max} - 1 \\ -(\lambda_s + \mu_s)P_s + \lambda_{s-1}P_{s-1} & S = S_{\max}. \end{cases} \quad (15)\end{aligned}$$

- (7) Apply the mutation operator according to (14).
- (8) Sort the population according to the objective function value from best to worst.
- (9) Apply elitism by replacing the p worst habitats of the previous generation with the p best ones.
- (10) Repeat step (3) until the maximum number of generations G_{\max} is reached.

4. Numerical Results

4.1. Comparison with Test Functions. In order to evaluate the different BBO migration models, first we use a set of test problems from the literature. In this paper, the test problems consist of eight well-known benchmark functions. We have chosen two unimodal and six multimodal functions. These are expressed as follows [27, 28].

(A) *Unimodal Functions*

(1) *Sphere Function*

$$f_1(x) = \sum_{j=0}^{D-1} x_j^2, \quad |x_j| \leq 100, \quad f_1(0, 0, \dots, 0) = 0. \quad (16)$$

(2) *Rosenbrock's Function*

$$\begin{aligned}f_2(x) &= \sum_{j=0}^{D-2} \left[100(x_{j+1} - x_j^2)^2 + (x_j - 1)^2 \right], \\ &|x_j| \leq 2.048, \quad f_2(1, 1, \dots, 1) = 0.\end{aligned}\quad (17)$$

(B) *Multimodal Functions*

(3) *Ackley's Function*

$$\begin{aligned}f_3(x) &= -20 \exp \left(-\frac{1}{5} \sqrt{\frac{1}{D} \sum_{j=0}^{D-1} x_j^2} \right) \\ &- \exp \left(\frac{1}{D} \sum_{j=0}^{D-1} \cos(2\pi x_j) \right) + 20 + e, \\ &|x_j| \leq 32, \quad f_3(0, 0, \dots, 0) = 0.\end{aligned}\quad (18)$$

(4) *Griewank's Function*

$$\begin{aligned}f_4(x) &= \sum_{j=0}^{D-1} \frac{x_j^2}{4000} - \prod_{j=0}^{D-1} \cos \left(\frac{x_j}{\sqrt{j}} \right), \\ &|x_j| \leq 600, \quad f_4(0, 0, \dots, 0) = 0.\end{aligned}\quad (19)$$

TABLE 1: Comparative results for the test functions. The smaller values are in bold.

Migration model	Test function							
	$f_1(x)$	$f_2(x)$	$f_3(x)$	$f_4(x)$	$f_5(x)$	$f_6(x)$	$f_7(x)$	$f_8(x)$
Linear	0 ± 0	50.592 ± 26.042	0.133 ± 0.084	0.744 ± 0.295	0 ± 0	0.012 ± 0.033	0.200 ± 0.400	3.934 ± 2.437
Sinusoidal	0.861 ± 0.552	29.844 ± 16.559	0.091 ± 0.090	0.725 ± 0.205	0 ± 0	0.003 ± 0.009	0.003 ± 0.010	2.561 ± 1.894
Model 7	0.893 ± 0.503	42.243 ± 23.893	0.201 ± 0.092	0.673 ± 0.142	0 ± 0	0.018 ± 0.023	0 ± 0	4.144 ± 3.704
Model 8	0.767 ± 0.898	70.062 ± 33.001	0.131 ± 0.081	0.701 ± 0.250	0 ± 0	0.004 ± 0.014	0 ± 0	3.209 ± 2.088

(5) Weierstrass Function

$$f_5(x) = \sum_{j=0}^{D-1} \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (x_j + 0.5))] \right) - D \left(\sum_{k=0}^{k_{\max}} [a^k \cos(\pi b^k)] \right), \quad (20)$$

$$a = 0.5, b = 3, k_{\max} = 20 \quad |x_j| \leq 0.5, f_5(0, 0, \dots, 0) = 0.$$

(6) Rastrigin's Function

$$f_6(x) = \sum_{j=0}^{D-1} [x_j^2 - 10 \cos(2\pi x_j) + 10], \quad (21)$$

$$|x_j| \leq 5.12, f_6(0, 0, \dots, 0) = 0.$$

(7) Noncontinuous Rastrigin's Function

$$f_7(x) = \sum_{j=0}^{D-1} [y_j^2 - 10 \cos(2\pi y_j) + 10], \quad (22)$$

$$y_j = \begin{cases} x_j & |x_j| < 0.5 \\ \frac{\text{round}(2x_j)}{2} & |x_j| \geq 0.5 \end{cases}$$

$$\text{for } j = 0, 1, \dots, D-1, f_7(0, 0, \dots, 0) = 0.$$

(8) Schwefel's Function

$$f_8(x) = 418.9829 \times D - \sum_{j=0}^{D-1} \left[x_j \sin(|x_j|^{1/2}) \right], \quad (23)$$

$$|x_j| \leq 500, f_8(420.96, 420.96, \dots, 420.96) = 0.$$

All algorithms are executed 50 times. The results are compared. The population size is set to 100 and the maximum number of generations is set to 2000 iterations. The problems dimension is set to $D = 30$. For all migration models the habitat modification probability, P_{mod} , is set to 1, and the maximum mutation rate, m_{\max} , is set equal to 0.005. The maximum immigration rate I and the maximum emigration rate E are both set to one. The elitism parameter p is set to two. Table 1 reports the comparative results in terms of mean and

TABLE 2: Average algorithm rankings obtained by Friedman test.

Algorithm	Average ranking
Linear	2.94
Sinusoidal	1.94
Model 7	2.88
Model 8	2.25

standard deviation values. We notice that BBO with the sinusoidal migration model performs better in 5 out of the 8 test functions. Moreover, in order to compare the algorithms performance on all problems, we have conducted the Friedman test [29]. Table 2 shows the average ranking of all algorithms. The highest ranking is shown in bold. It is obvious that the best average ranking was obtained by the BBO with sinusoidal migration model which outperforms the other models.

4.2. Comparison with Array Thinning Problems. We compare the four BBO migration models' performance with the two BPSO variants, BHS, ODE, and ACO on different thinned array design cases. All algorithms are executed 20 times. The results are compared. The population size is set to 200 and the maximum number of generations is set to 1000 iterations. For all migration models the habitat modification probability, P_{mod} , is set to 1, and the maximum mutation rate, m_{\max} , is set equal to 0.005. The maximum immigration rate I and the maximum emigration rate E are both set both to one. The elitism parameter p is set to two. In both BPSO algorithms the learning factors c_1, c_2 are both set equal to two, and inertia weight w is linearly decreased from 0.9 to 0.4 as in [6]. For BHS the HMCR is set to 0.99 and the PAR is set to 0.4. For ODE $F = 0.5$, CR = 0.9, and the jumping rate is set to 0.3. For ACO the initial pheromone value τ_0 is set to $1.0e-6$, the pheromone update constant Q is set to 20, the exploration constant q_0 is set to 1, the global pheromone decay rate ρ_g is 0.9, the local pheromone decay rate ρ_l is 0.5, the pheromone sensitivity α is 1, and the visibility sensitivity is β is 5.

The first case is that of a symmetrically excited 300-element linear array. The total number of unknowns for this case is 150. Table 3 reports the comparative results. We notice that the sinusoidal model and the model 7 seem to outperform the other algorithms. Both models have obtained the same best value. Both BPSO variants perform worse than the BBO algorithms. ACO and VBPSO perform similarly. The convergence rate graph for this case is depicted in Figure 1. All BBO models seem to converge at similar speed faster than the other algorithms. The VBPSO converges faster than BPSO. It must be pointed out that the BBO algorithms converge

TABLE 3: Comparative results for 300-element symmetric thinned array. The smaller values are in bold.

Migration model	Best	Worst	Mean	St. dev.
Linear	-24.54	-24.24	-24.43	0.096
Sinusoidal	-24.67	-24.17	-24.41	0.146
Model 7	-24.67	-24.27	-24.43	0.110
Model 8	-24.44	-23.51	-24.11	0.243
VBPSO	-24.32	-23.67	-24.05	0.175
BPSO	-21.10	-19.88	-20.35	0.326
BHS	-24.10	-22.79	-23.61	0.431
ODE	-24.06	-23.30	-23.76	0.203
ACO	-24.20	-23.93	-24.06	0.084

TABLE 4: Comparative results for 300-element symmetric thinned array with same aperture size. The smaller values are in bold.

Migration model	Best	Worst	Mean	St. dev.
Linear	-24.45	-24.06	-24.32	0.112
Sinusoidal	-24.59	-24.30	-24.45	0.087
Model 7	-24.70	-24.27	-24.41	0.127
Model 8	-24.56	-23.64	-24.18	0.257
VBPSO	-24.41	-23.97	-24.22	0.141
BPSO	-21.50	-20.52	-20.95	0.281
BHS	-24.16	-22.90	-23.79	0.413
ODE	-24.30	-23.51	-23.91	0.231
ACO	-23.36	-22.76	-23.16	0.171

at their final value at less than 300 iterations. The radiation pattern of the best obtained result is shown in Figure 2. The best array found is filled 72% and has a peak SLL of -24.67 dB. Additionally, we study the effect of maintaining the same aperture length as the original uniform array for this case. We therefore force the first and the last element to be one (turned on) for a design case. The comparative results for this case are shown in Table 4. The sinusoidal model presents the best performance except the best value found where the model 7 outperforms the others. VBPSO is complete with the BBO algorithms for this case and achieves better performance than BBO with model 8. ODE performs better than ACO and BHS for this case. Figure 3 shows the convergence rate for this case. It is obvious that all BBO algorithms converge at similar speeds, while the PSO algorithms converge slower. The best obtained radiation pattern is shown in Figure 4. The obtained peak SLL value is -24.70 dB slightly smaller than the previous case. The array filling percentage is again 72%. We notice that there is not a significant difference if we choose an array without constraints or with the same aperture size for this case.

Next, we consider asymmetric array designs in which the problem dimension is 300. Therefore, we evaluate the algorithms ability to solve high dimensional problems. It must be pointed out that although the number of unknowns has doubled compared with the symmetric case the population size and iterations remain the same as previously. Therefore, the level of difficulty increases for all algorithms. Table 5 reports the comparative results for this case. The algorithms

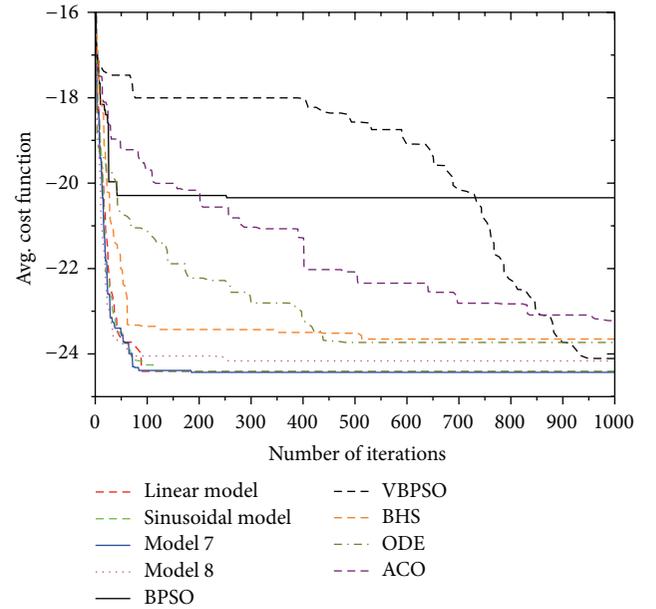


FIGURE 1: Convergence rate graph for the 300-element symmetric thinned array case.

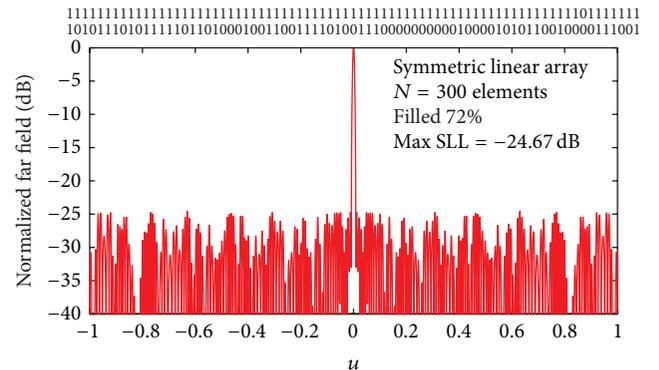


FIGURE 2: Radiation pattern of the best obtained array for the 300-element symmetric thinned array.

performance differences are clearer in this case. It is obvious that the sinusoidal model clearly outperforms the others in this case. The results obtained by model 7 and linear model are quite similar in this case. The VBPSO results are better again than BPSO, BHS, ODE, and ACO but worse than the BBO algorithms. Figure 5 shows the convergence rate graph for this case. Model 7 seems to convergence slightly faster than the other algorithms. The radiation pattern of the best obtained array is shown in Figure 6. The obtained array SLL is -26.11 dB and the fill percentage is about 72%.

In order to further evaluate the algorithms performance we choose again an asymmetric array case with the same aperture length. Table 6 holds the comparative results. The results show that sinusoidal model clearly outperforms the other algorithms. We notice that VBPSO and BPSO results are worse than the BBO algorithms. BHS is complete with the VBPSO algorithm for this case. The convergence rate graph of Figure 7 shows that models 7 and 8 converge slightly

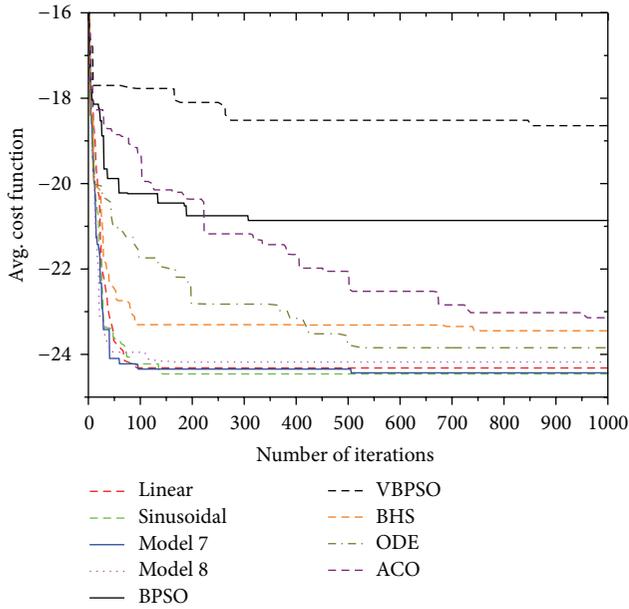


FIGURE 3: Convergence rate graph for the 300-element symmetric thinned array case with same aperture size.

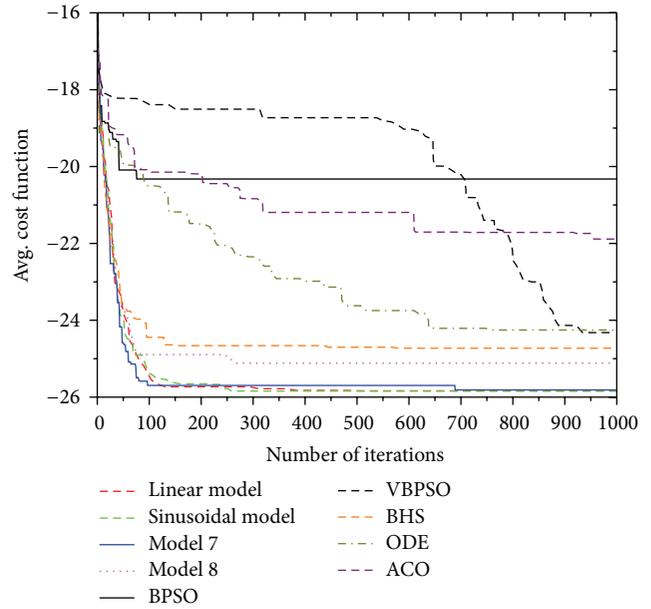


FIGURE 5: Convergence rate graph for the 300-element asymmetric thinned array.

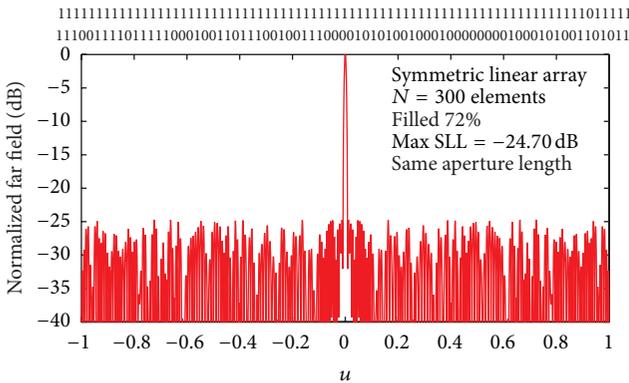


FIGURE 4: Radiation pattern of the best obtained array for the 300-element symmetric thinned array with same aperture length.

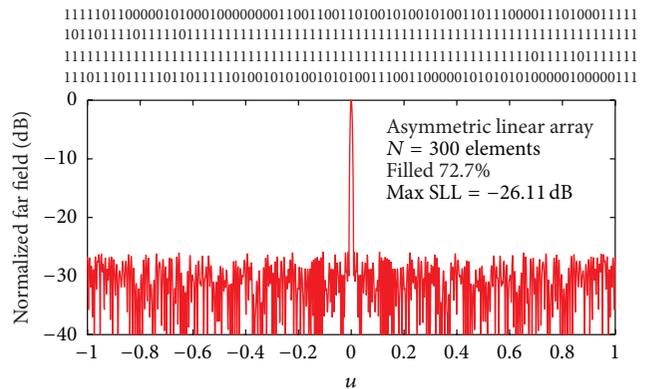


FIGURE 6: Radiation pattern of the best obtained array for the 300-element asymmetric thinned array.

TABLE 5: Comparative results for 300-element asymmetric thinned array. The smaller values are in bold.

Migration model	Best	Worst	Mean	St. dev.
Linear	-26.03	-25.65	-25.83	0.119
Sinusoidal	-26.11	-25.73	-25.88	0.110
Model 7	-26.05	-25.58	-25.82	0.146
Model 8	-25.57	-24.21	-25.13	0.362
VBPSO	-24.98	-23.49	-24.36	0.468
BPSO	-20.83	-19.98	-20.32	0.228
BHS	-25.32	-24.63	-25.02	0.214
ODE	-24.88	-23.51	-24.19	0.424
ACO	-22.10	-21.55	-21.83	0.187

faster than the linear and the sinusoidal model for this case. Again it is obvious that the BBO algorithms require fewer iterations than the other algorithms in order to reach the

TABLE 6: Comparative results for 300-element asymmetric thinned array with same aperture size. The smaller values are in bold.

Migration model	Best	Worst	Mean	St. dev.
Linear	-25.97	-25.58	-25.78	0.113
Sinusoidal	-26.08	-25.77	-25.91	0.107
Model 7	-25.87	-25.60	-25.76	0.081
Model 8	-25.67	-24.46	-25.07	0.383
VBPSO	-24.90	-22.54	-23.92	0.658
BPSO	-20.80	-19.87	-20.21	0.276
BHS	-25.58	-23.92	-24.76	0.633
ODE	-24.92	-23.44	-24.26	0.522
ACO	-21.91	-21.63	-21.77	0.008

final values. Figure 8 presents the radiation pattern of the best obtained array. The peak SLL is -26.08 dB and the filling

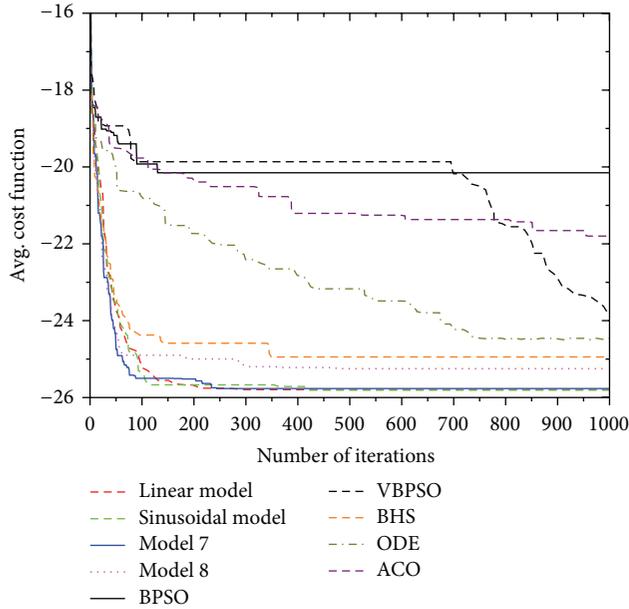


FIGURE 7: Convergence rate graph for the 300-element asymmetric thinned array with same aperture size.

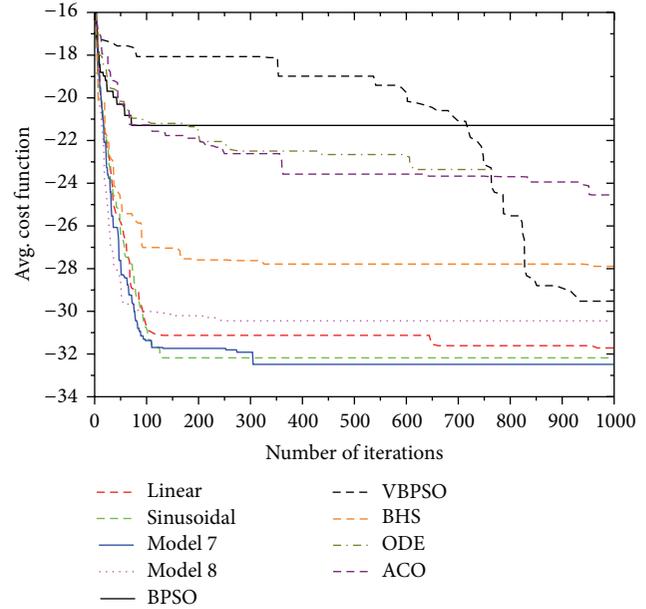


FIGURE 9: Convergence rate graph for the planar thinned array design case.

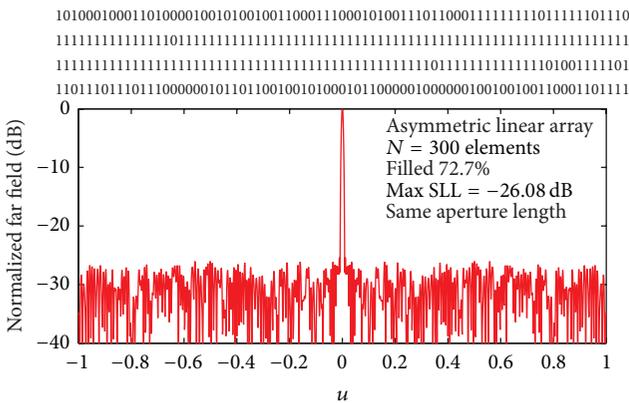


FIGURE 8: Radiation pattern of the best obtained array for the 300-element asymmetric thinned array with same aperture length.

percentage is 73%. Again the results are quite similar with the previous case with the peak SLL slightly higher than previous.

The final example is that of planar thinned array. We consider a 50×20 planar array. All elements are equally spaced along x - and y -axis at half-wavelength distance. Again, a population of 200 vectors is selected for all algorithms. The total number of generations is set to 500. Table 7 holds the comparative results for the planar array case. The sinusoidal model obtains the best value. However, the model 7 results are better than the results of the other algorithms models in terms of best mean value. The convergence rate graph for this case is shown in Figure 9. The BBO algorithms seem to converge at similar speed, faster than the other algorithms. The 3D radiation pattern of the best obtained array is shown in Figure 10(a). Figure 10(b) shows the array pattern at the two phi-planes. For $\phi = 0^\circ$ the PSL is -33.96 dB while for $\phi = 90^\circ$ the PSL is -33.85 dB. The filling percentage is 47.6%.

TABLE 7: Comparative results for 1000-element planar thinned array. The smaller values are in bold.

Migration model	Best	Worst	Mean	St. dev.
Linear	-32.34	-30.80	-31.38	0.421
Sinusoidal	-33.85	-30.11	-32.13	1.076
Model 7	-33.44	-31.15	-32.27	0.710
Model 8	-32.30	-29.34	-30.55	0.899
VPBSO	-31.55	-28.44	-29.55	1.107
BPSO	-21.87	-20.89	-21.27	0.311
BHS	-31.96	-27.78	-29.84	1.290
ODE	-24.63	-22.90	-23.40	0.454
ACO	-25.19	-23.98	-24.58	0.346

Overall BBO with sinusoidal model has been ranked first regarding mean and standard deviation values in 3 out of the 5 array design cases presented. Additionally, BBO with sinusoidal model in 4 out of the 5 design cases produced the best result. The results obtained by the BBO with model 7 outperformed the linear model in 2 out of the 5 cases. In all cases BBO with migration model 8 seems to perform worse than the other models. The VBPSO has outperformed the original BPSO in all cases and the BBO with migration model 8 in 2 out of the 5 cases. In order to compare the algorithms performance on all problems, we have conducted the Friedman test [29]. Table 8 shows the average ranking of all algorithms. The highest ranking is shown in bold. It is obvious that the best average ranking was obtained by the BBO with sinusoidal migration model which outperforms the other algorithms. VBPSO and BHS perform similarly, while ODE outperforms ACO.

Similar to the other evolutionary algorithms (EAs) such as differential evolution (DE), Harmony Search (HS), and

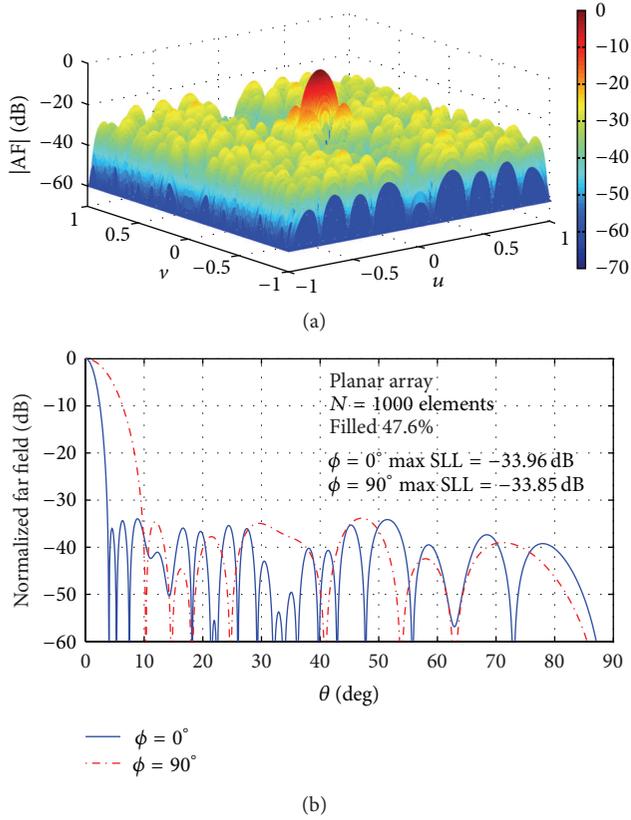


FIGURE 10: Best obtained 1000-element planar array (a) 3D radiation pattern of the (b) far-field patterns at $\phi = 0^\circ$ and $\phi = 90^\circ$ planes.

TABLE 8: Average algorithm rankings obtained by Friedman test.

Algorithm	Average ranking
Linear	2.1
Sinusoidal	1.8
Model 7	2.1
Model 8	4.4
VBPSO	5.8
BPSO	8.8
BHS	5.8
ODE	6.6
ACO	7.2

Particle Swarm Optimization (PSO), in the BBO approach there is a way of sharing information between solutions [16]. This feature makes BBO suitable for the same types of problems that the other algorithms are used for, namely, high dimensional data. Additionally, BBO has some unique features that are different from those found in the other evolutionary algorithms. For example, quite different from DE and PSO, from one generation to the next the set of the BBO's solutions is maintained and improved using the migration model, where the emigration and immigration rates are determined by the fitness of each solution. BBO differs from PSO in the fact that PSO solutions do not change directly; the velocities change. The BBO solutions share

directly their attributes using the migration models. These differences can make BBO outperform other algorithms [16, 17, 30]. It must be pointed out that if PSO or DE are constrained to discrete space then the next generation will not necessarily be discrete [30]. However, this is not true for BBO; if BBO is constrained to a discrete space then the next generation will also be discrete to the same space. As the authors in [30] suggest this indicates that BBO could perform better than other EAs on combinatorial optimization problems, which makes BBO suitable for application to the antenna array thinning problems.

4.3. Boundary Conditions Constraint Handling Methods Study. In this subsection, we apply different boundary conditions handling methods for the BBO with the sinusoidal model in order to find if the settings used in the previous sections could be improved. The boundary constraint handling methods that we will test include the following [31, 32].

(1) *Reflection Method*

$$x_j^v = \begin{cases} 2x_{L,j} - x_j & \text{if } x_j < x_{L,j} \\ 2x_{U,j} - x_j & \text{if } x_j > x_{U,j} \end{cases} \quad (24)$$

where x_j^v is a valid value, x_j is the value which violates the bound constraint, and $x_{L,j}$ and $x_{U,j}$ are the lower and upper bounds for the j th variable, respectively.

(2) *Projection Method*

$$x_j^v = \begin{cases} x_{L,j} & \text{if } x_j < x_{L,j} \\ x_{U,j} & \text{if } x_j > x_{U,j} \end{cases} \quad (25)$$

In this case, the variables that violate the bound constraints are trimmed to the lower and upper bounds, respectively.

(3) *Reinitialization by Position.* In this case each variable that violates the constraints is randomly reinitialized with

$$x_j^v = \text{rand}_{j[0,1]} (x_{U,j} - x_{L,j}) + x_{L,j}, \quad (26)$$

where $\text{rand}_{j[0,1]}$ is a uniformly distributed random number between 0 and 1.

(4) *Reinitialize All.* In this case, if at least one of the solution variables violates the boundaries, a complete new vector is generated within the allowed boundaries using (26).

(5) *Conservatism.* This technique was proposed to work particularly with DE. In this case, the infeasible solution is rejected and it is replaced by the original feasible vector.

The test function we use in all cases is a symmetrically thinned array with 100 elements. We run each case for 50 independent trials for each different boundary condition setting. The population size is set to 200 and the number of iterations to 1000. The best value, the worst value, the mean, and the standard deviation at the last generation are presented

TABLE 9: Comparative results using different boundary conditions constraint handling methods. The smaller values are in bold.

Boundary condition	Best	Worst	Mean	St. dev.
Projection	-21.11	-20.50	-20.80	0.138
Reflection	-21.02	-20.41	-20.77	0.143
Reinitialization	-21.28	-20.49	-20.76	0.155
Reinitialize all	-21.08	-20.41	-20.73	0.159
Conservatism	-21.33	-20.38	-20.78	0.189

here. Table 9 holds the comparative results for all methods. We notice that the projection method in this case has obtained the best mean, worst, and standard deviation values. The conservatism method has obtained the best objective function value. The mean value results for all methods are close, while the best obtained values seem to differ.

5. Conclusion

In this paper, we have addressed the problem of designing thinned arrays using the BBO algorithm. We have compared performance of four different BBO migration models with other popular EAs. The results showed that BBO with sinusoidal model is highly competitive for the array thinning problem. BBO with sinusoidal model outperformed the other migration models in general. All the BBO algorithms converge faster and produce better results than the other algorithms. However, these results are considered to be indicative and cannot be generalized in all array design problems. Further tests should be carried out to test the BBO performance on other array design problems. The BBO algorithm is a powerful and efficient optimizer especially in combinatorial optimization problems. In our future work, we will study further the performance of BBO migration models to different antenna design problems.

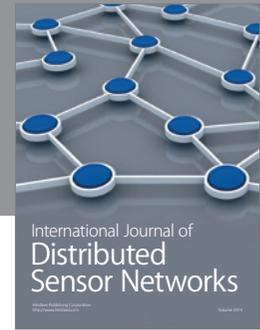
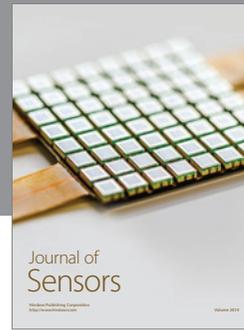
Competing Interests

The authors declare that they have no competing interests.

References

- [1] R. Jain and G. S. Mani, "Solving 'antenna array thinning problem' using genetic algorithm," *Applied Computational Intelligence and Soft Computing*, vol. 2012, Article ID 946398, 14 pages, 2012.
- [2] R. L. Haupt, "Thinned arrays using genetic algorithms," *IEEE Transactions on Antennas and Propagation*, vol. 42, no. 7, pp. 993–999, 1994.
- [3] N. Jin and Y. Rahmat-Samii, "Advances in particle swarm optimization for antenna designs: real-number, binary, single-objective and multiobjective implementations," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3 I, pp. 556–567, 2007.
- [4] Ó. Quevedo-Teruel and E. Rajo-Iglesias, "Ant colony optimization in thinned array synthesis with minimum sidelobe level," *IEEE Antennas and Wireless Propagation Letters*, vol. 5, no. 1, pp. 349–352, 2006.
- [5] J. Kennedy and R. C. Eberhart, "A discrete binary version of the particle swarm algorithm," in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics. Computational Cybernetics and Simulation*, pp. 4104–4108, IEEE, Orlando, Fla, USA, 1997.
- [6] S. Mirjalili and A. Lewis, "S-shaped versus V-shaped transfer functions for binary Particle Swarm Optimization," *Swarm and Evolutionary Computation*, vol. 9, pp. 1–14, 2013.
- [7] R. Storn and K. Price, "Differential evolution—a simple and efficient adaptive scheme for global optimization over continuous spaces," 1995, <http://citeseer.ist.psu.edu/article/storn95differential.html>.
- [8] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.
- [9] R. S. Rahnamayan, H. R. Tizhoosh, and M. M. A. Salama, "Opposition-based differential evolution," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, pp. 64–79, 2008.
- [10] H. R. Tizhoosh, "Opposition-based learning: a new scheme for machine intelligence," in *Proceedings of the International Conference on Computational Intelligence for Modelling, Control and Automation (CIMCA '05) and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC '05)*, vol. 1, pp. 695–701, Vienna, Austria, November 2005.
- [11] Z. W. Geem, J. H. Kim, and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60–68, 2001.
- [12] S.-H. Yang and J.-F. Kiang, "Optimization of sparse linear arrays using harmony search algorithms," *IEEE Transactions on Antennas and Propagation*, vol. 63, no. 11, pp. 4732–4738, 2015.
- [13] K. Guney and M. Onay, "Optimal synthesis of linear antenna arrays using a harmony search algorithm," *Expert Systems with Applications*, vol. 38, no. 12, pp. 15455–15462, 2011.
- [14] M. Dorigo, V. Maniezzo, and A. Colnari, "Ant system: optimization by a colony of cooperating agents," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 26, no. 1, pp. 29–41, 1996.
- [15] E. Rajo-Iglesias and Ó. Quevedo-Teruel, "Linear array synthesis using an ant-colony-optimization-based algorithm," *IEEE Antennas and Propagation Magazine*, vol. 49, no. 2, pp. 70–79, 2007.
- [16] D. Simon, "Biogeography-based optimization," *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 6, pp. 702–713, 2008.
- [17] H. Ma, "An analysis of the equilibrium of migration models for biogeography-based optimization," *Information Sciences*, vol. 180, no. 18, pp. 3444–3464, 2010.
- [18] W. Guo, L. Wang, and Q. Wu, "An analysis of the migration rates for biogeography-based optimization," *Information Sciences*, vol. 254, pp. 111–140, 2014.
- [19] U. Singh, H. Kumar, and T. S. Kamal, "Design of Yagi-Uda antenna using biogeography based optimization," *IEEE Transactions on Antennas and Propagation*, vol. 58, no. 10, pp. 3375–3379, 2010.
- [20] M. R. Lohokare, S. S. Pattnaik, S. Devi, K. M. Bakwad, and J. G. Joshi, "Parameter calculation of rectangular microstrip antenna using biogeography-based optimization," in *Proceedings of the Applied Electromagnetics Conference (AEMC '09)*, pp. 1–4, Kolkata, India, December 2009.

- [21] M. R. Lohokare, S. S. Pattnaik, S. Devi, B. K. Panigrahi, K. M. Bakwad, and J. G. Joshi, "Modified BBO and calculation of resonant frequency of circular microstrip antenna," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 487–492, IEEE, Coimbatore, India, December 2009.
- [22] U. Singh, H. Kumar, and T. S. Kamal, "Linear array synthesis using biogeography based optimization," *Progress in Electromagnetics Research M*, vol. 11, pp. 25–36, 2010.
- [23] A. Sharaqa and N. Dib, "Design of linear and circular antenna arrays using biogeography based optimization," in *Proceedings of the 2011 1st IEEE Jordan Conference on Applied Electrical Engineering and Computing Technologies (AEECT '11)*, pp. 1–6, Amman, Jordan, December 2011.
- [24] S. K. Goudos, K. B. Baltzis, K. Siakavara, T. Samaras, E. Vafiadis, and J. N. Sahalos, "Reducing the number of elements in linear arrays using biogeography-based optimization," in *Proceedings of the 6th European Conference on Antennas and Propagation (EuCAP '12)*, pp. 1615–1618, Prague, Czech Republic, March 2012.
- [25] M. Dorigo and L. M. Gambardella, "Ant colonies for the travelling salesman problem," *BioSystems*, vol. 43, no. 2, pp. 73–81, 1997.
- [26] M. Dorigo and T. Stutzle, *Ant Colony Optimization*, The MIT Press, Cambridge, Mass, USA, 2004.
- [27] E. Mezura-Montes, J. Velázquez-Reyes, and C. A. Coello Coello, "A comparative study of differential evolution variants for global optimization," in *Proceedings of the 8th Annual Genetic and Evolutionary Computation Conference (GECCO '06)*, pp. 485–492, Seattle, Wash, USA, July 2006.
- [28] J. J. Liang, A. K. Qin, P. N. Suganthan, and S. Baskar, "Comprehensive learning particle swarm optimizer for global optimization of multimodal functions," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 3, pp. 281–295, 2006.
- [29] S. García, A. Fernández, J. Luengo, and F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power," *Information Sciences*, vol. 180, no. 10, pp. 2044–2064, 2010.
- [30] H. Ma, D. Simon, M. Fei, and Z. Chen, "On the equivalences and differences of evolutionary algorithms," *Engineering Applications of Artificial Intelligence*, vol. 26, no. 10, pp. 2397–2407, 2013.
- [31] J. Arabas, A. Szczepankiewicz, and T. Wroniak, "Experimental comparison of methods to handle boundary constraints in differential evolution," in *Parallel Problem Solving from Nature, PPSN XI: 11th International Conference, Kraków, Poland, September 11–15, 2010, Proceedings, Part II*, R. Schaefer, C. Cotta, J. Kołodziej, and G. Rudolph, Eds., pp. 411–420, Springer, Berlin, Germany, 2010.
- [32] E. Juarez-Castillo, N. Perez-Castro, and E. Mezura-Montes, "A novel boundary constraint-handling technique for constrained numerical optimization problems," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '15)*, pp. 2034–2041, Sendai, Japan, May 2015.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

