

INTERFERENCE SUPPRESSION OF LINEAR ANTENNA ARRAYS BY AMPLITUDE-ONLY CONTROL USING A BACTERIAL FORAGING ALGORITHM

K. Guney

Department of Electrical and Electronics Engineering
Faculty of Engineering
Erciyes University
Kayseri 38039, Turkey

S. Basbug

Department of Industrial Electronics
Vocational College
Nevsehir University
Nevsehir, Turkey

Abstract—This paper presents a bacterial foraging algorithm (BFA) for null steering of linear antenna arrays by controlling only the element amplitudes. The BFA is a new evolutionary computing technique based on the foraging behavior of *Escherichia (E.) coli* bacteria in human intestine. To show the accuracy and flexibility of the proposed BFA, several examples of Chebyshev array pattern with the imposed single, multiple and broad nulls are given. It is found that the nulling method based on BFA is capable of steering the array nulls precisely to the undesired interference directions.

1. INTRODUCTION

Antenna arrays have been widely used in mobile, wireless, satellite and radar communications systems to improve signal quality, thereby increasing system coverage, capacity and link quality. The performance of these systems depends firmly on the antenna array design. The process of determining the parameters of an antenna array to obtain the required antenna radiation pattern is known as pattern synthesis. Due to the increasing pollution of the electromagnetic environment, the antenna array, which allows placing nulls in the far field pattern at

specified directions, is becoming important in communication systems, sonar, and radar applications for maximizing signal-to-interference ratio [1–26]. Methods of null steering, which have been studied extensively in the past [1–26], include controlling the amplitude-only, the phase-only, the position-only, and the complex weights (both the amplitude and phase) of the array elements. These methods have been used with their own benefits and limitations. In spite of the large variety of nulling methods, it appears that none of them completely satisfies the requirement of a general and flexible solution for the pattern synthesis problem.

Interference suppression with complex weights is the most effective since it has the larger solution alternatives [1, 13, 16, 21]. However, it is also the most expensive considering the cost of the controllers used for phase shifters and variable attenuators for each array element. Moreover, when the number of elements in the array increases, the computational time to find the values of element amplitudes and phases will also increase. The amplitude-only control [3, 4, 13, 14, 19, 23, 25] uses a set of variable attenuators to adjust the element amplitudes. If the element amplitudes have even symmetry about the center of the array, the number of attenuators and the computational time are halved. Without any approximation, the problem for the phase-only and position-only nulling methods is inherently nonlinear and, it can not be solved by analytical methods in a direct way. The phase-only control [5, 6, 9, 17, 22, 24] utilizes the phase shifters while the position-only control [7–11, 15, 26] needs a mechanical driving system such as servomotors to move the array elements. By assuming that the phase perturbations are small, the nulling equations can be linearized, but it makes impossible to place nulls at symmetric location with respect to the main beam [1]. In null synthesizing with small phase perturbations, there is an unavoidable sidelobe level increase in the direction symmetric to nulling direction with respect to the main beam. In order to steer the nulls symmetrically with respect to the main beam, the methods based on nonlinear optimization techniques have been proposed [5, 6], however, the resultant patterns of these methods have considerable pattern distortion because the phase perturbations used are large. Phase-only null synthesizing is less complicated and attractive for the phased arrays since the required controls are available at no extra cost, but it has still common problem.

In recent years, the methods [4, 6, 8, 9, 12–21, 23, 25–52] based on the genetic algorithm, ant colony optimization, particle swarm optimization, tabu search, bees algorithm, differential evolution, and clonal selection have become more popular, and they have been used in solving antenna array pattern synthesis problems. The performances

of these methods are found to be better than those of the classical optimization techniques and the conventional analytical techniques. Each of these methods has its specific advantages and disadvantages.

In this paper, an alternative method based on the bacterial foraging algorithm (BFA) [53] is presented for linear antenna array pattern synthesis with prescribed nulls. The array element amplitudes are the only controlling parameters, and the main aim is to synthesize array patterns with nulls imposed on directions of interferences.

The BFA is a new evolutionary computation technique based on the foraging behavior of *Escherichia (E.) coli* bacteria in human intestine [53]. It was successfully used to solve various kinds of engineering problems [54–63]. It has been shown by Lin and Liu [56] that the BFA is better than the particle swarm optimization in the identification of the Hammerstein model in terms of convergence, robustness and precision. Niu et al. [58] proposed a methodology for optimal design of classical proportional-integral-derivative (PID) controllers based on BFA. The design, implementation, and testing of BFA-PID were compared with those of genetic algorithm (GA)-PID. It was illustrated in [58] that a faster settling time, less or no overshoot and higher robustness are obtained with BFA-PID controller. A nonlinear channel equalizer using BFA was presented in [60]. The recovery performance of the equalizer was obtained through computer simulation study using nonlinear channels. It was shown in [60] that the BFA based equalizer offers superior performance both in terms of bit-error-rate and convergence speed compared to the GA based equalizers. In addition it requires substantially less computation during training. BFA was used in optimizing the coefficients of proportional plus integral controllers [63]. It was demonstrated in [63] that BFA converges faster than the GA.

The remainder of the paper is organized as follows. Section 2 explains formulation of the problem. Section 3 gives an overview of the BFA. In Section 4, the numerical examples are provided and discussed. Finally, the conclusion is given in Section 5.

2. PROBLEM FORMULATION

Let us assume that the array elements are symmetrically situated and excited around the center of the linear array. The far field array factor of such an array with an even number of isotropic elements ($2M$) is defined as

$$AF(\theta) = 2 \sum_{n=1}^M a_n \cos \left(\frac{2\pi}{\lambda} d_n \sin \theta \right) \quad (1)$$

where a_n is the amplitude of the n th element, θ is the angle from broadside, and d_n is the distance between position of the n th element and the array center. Generally, the main beam of the array pattern is required to be directed to the desired signal and the undesired interference signals from other directions to be suppressed as much as possible. To find an appropriate set of required element amplitudes (a_n) that achieve interference suppression, the BFA is used to minimize the following cost function.

$$J = \sum_{\theta=-90^\circ}^{90^\circ} [W(\theta) |AF_o(\theta) - AF_d(\theta)| + ESL(\theta)] \quad (2)$$

where $AF_o(\theta)$ and $AF_d(\theta)$ are, respectively, the pattern obtained by using the BFA and the desired pattern. $W(\theta)$ and $ESL(\theta)$ are employed to control the null depth level and maximum sidelobe level, respectively. The values of $W(\theta)$ and $ESL(\theta)$ are selected by experience such that the cost function is capable of guiding potential solutions to obtain satisfactory array pattern performance with desired properties. The factors $W(\theta)$ and $ESL(\theta)$ give the antenna designer greater flexibility and control over the actual pattern. The trade-off of the relative importance between null depth and sidelobe level can easily be obtained by changing the values of these factors.

3. BACTERIAL FORAGING ALGORITHM (BFA)

3.1. Bacterial Foraging

Natural selection has a tendency to eliminate animals having poor foraging strategies and favor the ones with successful foraging strategies to propagate their genes as these are more likely to reach a successful reproduction. Poor foraging strategies are either completely eliminated or transformed into good ones after many generations are produced. This activity of foraging inspired the researchers to utilize it as a novel optimization tool. The *E. coli* bacteria present in our intestines also practice a foraging strategy. The control system of these bacteria governing their foraging process can be subdivided into four sections, which are chemotaxis, swarming, reproduction and elimination and dispersal [53–55].

Chemotaxis: The chemotaxis process in the control system is attained by swimming and tumbling via flagella. As viewed from the free end of the flagellum looking toward the cell, each flagellum is a left-handed helix that is configured so as to rotate the base of the flagellum in counterclockwise direction to produce a force against the

bacterium which pushes the cell. On the other hand, each flagellum pulls on the cell if they rotate in clockwise direction. The net effect of this that each flagellum rotates relatively independently of the others thereby making the bacterium tumble about. Hence, the modes of operation that an *E. coli* bacterium performs in its entire lifetime are that of running (swimming for a period of time), tumbling or switching between running and tumbling. Running, tumbling, and chemotactic behaviour of *E. coli* bacteria are shown in Fig. 1. The change in the concentration of the nutrient triggers a reaction such that the bacterium will spend more time swimming and less time tumbling.

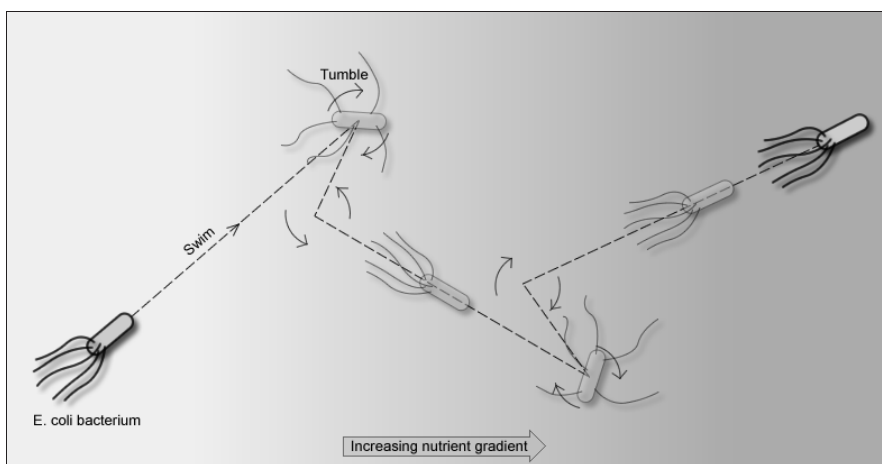


Figure 1. Running, tumbling, and chemotactic behaviour of *E. coli* bacteria

Swarming: When a group of *E. coli* bacteria is placed in the semisolid agar having a single nutrient chemo-effector (sensor), they move from the center to outwards direction in a moving ring of bacteria by following the nutrient gradient produced by the group by consuming the nutrient. Furthermore, the bacteria release attractant aspartate if high levels of succinate are used as the nutrient, which lead the bacteria to concentrate into groups and hence move as concentric patterns of groups with high bacterial density. The spatial order depends both the outward movement of the ring and the local releases of the attractant, which functions as an attraction signal between bacteria to gather into a swarm.

Reproduction: The less healthy bacterium dies and the other healthier bacterium splits into two bacteria, which are placed in the

same location. This method of reproduction keeps the population of the bacteria constant.

Elimination and Dispersal: In the local environment of the bacteria, the lives of a bacteria population may change either gradually (e.g., via consumption of the nutrients) or suddenly as a result of some other influence. All the bacteria in a local region may be killed or a group may be dispersed into a new location in the environment. It is possible that they destroy the chemotactic progress. It is also possible that they assist the chemotaxis if the dispersal places the bacteria near good food source.

3.2. Bacterial Foraging Algorithm

The BFA is an evolutionary computing technique based on the foraging behavior of *E. coli* bacteria in human intestine [53]. A flowchart of the BFA utilized in this paper is shown in Fig. 2. In this figure, $J(i, j, k, l)$ represents the cost function where i is the index of the bacterium, j is the index for the chemotactic step, k is the index for the reproduction step, and l is the index of the elimination-dispersal event.

The BFA is controlled by a number of external parameters. These are the number of bacteria (S), maximum number of chemotactic loop (N_c), maximum number of reproduction (N_{re}), maximum number of elimination and dispersal events (N_{ed}), dimension of the search space (p), divisor of the step size (d_s), probability for the elimination-dispersal event (P_{ed}), swim length (N_s), counter for swim length (m), and chemotactic step sizes $C(i)$, $i = 1, 2, \dots, S$.

The bacterium is pointed in a random direction after a tumble. To represent a tumble, a unit length random direction $\delta(j)$ is produced; this is used to determine the direction of movement after a tumble:

$$\zeta^i(j+1, k, l) = \zeta^i(j, k, l) + C(i)\delta(j) \quad (3)$$

where $\zeta^i(j, k, l)$ is the i th bacterium at j th chemotactic, k th reproductive, and l th elimination and dispersal step.

The BFA algorithm [53] can be described as follows:

Initialization

For initialization, p , S , N_c , N_s , N_{re} , N_{ed} , P_{ed} , d_s , and the $C(i)$, $i = 1, 2, \dots, S$ must be chosen. The BFA starts by randomly distributing the bacteria across the domain of the optimization problem.

Iterative Algorithm

The iterative algorithm models the bacterial population chemotaxis, reproduction, and elimination and dispersal (initially, $j = k = l = 0$)

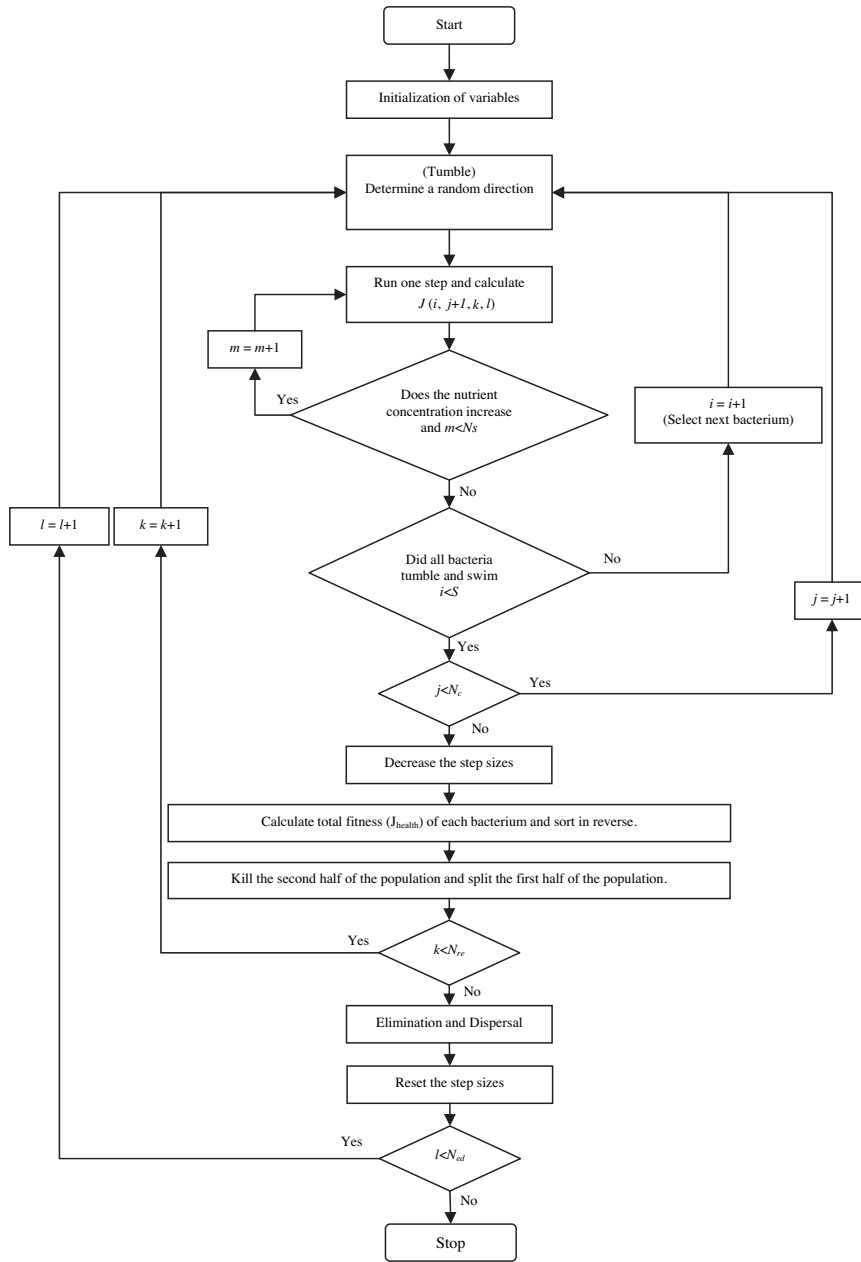


Figure 2. Flowchart of the bacterial foraging algorithm.

- 1) Elimination-dispersal loop: $l = l + 1$
- 2) Reproduction loop: $k = k + 1$
- 3) Chemotaxis loop: $j = j + 1$
 - a) For $i = 1, 2, \dots, S$, take a chemotactic step for bacterium i as follows.
 - b) Compute $J(i, j, k, l)$.
 - c) Let $J_{last} = J(i, j, k, l)$ to save this value since we may find a better cost via a run.
 - d) Tumble: Generate a random vector $\Delta(i) \in \mathfrak{R}^p$ with each element $\Delta_m(i)$, $m = 1, 2, \dots, p$, a random number on $[-1, 1]$.
 - e) Move: Let

$$\zeta^i(j + 1, k, l) = \zeta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

This results in a step of size $C(i)$ in the direction of the tumble for bacterium i .

- f) Calculate $J(i, j+1, k, l)$.
 - g) Swim (note that we use an approximation since we decide swimming behavior of each cell as if the bacteria numbered $\{1, 2, \dots, i\}$ have moved and $\{i+1, i+2, \dots, S\}$ have not; this is much simpler to simulate than simultaneous decisions about swimming and tumbling by all bacteria at the same time):
 - i) Let $m = 0$ (counter for swim length).
 - ii) While $m < N_s$ (if have not climbed down too long)
 - Let $m = m + 1$
 - If $J(i, j+1, k, l) < J_{last}$ (if doing better), let $J_{last} = J(i, j+1, k, l)$ and let
- $$\zeta^i(j + 1, k, l) = \zeta^i(j + 1, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$
- and use this $\zeta^i(j + 1, k, l)$ to compute the *new* $J(i, j+1, k, l)$ as we did in *f*).
- Else, let $m = N_s$. This is the end of the while statement.
 - h) Go to next bacterium ($i + 1$) if $i \neq S$ (i.e., go to b) to process the next bacterium).
- 4) If $j < N_c$, go to step 3. In this case, continue chemotaxis, since the life of the bacteria is not over.
 - 5) Decrease the step sizes $C(i)$, $i = 1, 2, \dots, S$ by dividing d_s

6) Reproduction:

- a) For the given
- k
- and
- l
- , and for each
- $i = 1, 2, \dots, S$
- , let

$$J_{health}^i = \sum_{j=1}^{N_{c+1}} J(i, j, k, l)$$

be the health of bacterium i (a measure of how many nutrients it got over its lifetime and how successful it was at avoiding noxious substances). Sort bacteria and chemotactic parameters $C(i)$ in order of ascending cost J_{health} .

- b) The S_r bacteria with the highest J_{health} values die and the other S_r bacteria with the best values split (and the copies that are made are placed at the same locations as their parent).
- 7) If $k < N_{re}$, go to step 2. In this case, we have not reached the number of specified reproduction steps, so we start the next generation in the chemotactic loop.
- 8) Elimination-dispersal: For $i = 1, 2, \dots, S$, with probability P_{ed} , eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if you eliminate a bacterium, simply disperse one to a random location on the optimization domain.
- 9) Reset the step sizes $C(i)$, $i = 1, 2, \dots, S$ to the initial values.
- 10) If $l < N_{ed}$ then go to step 1; otherwise end.

The details on the BFA can be found in [53]. In this paper, the BFA described above is successfully used to optimize the antenna array element amplitudes to exhibit an array pattern with the imposed single, multiple and broad nulls.

4. NUMERICAL RESULTS

To show the capability and flexibility of the proposed BFA for steering single, multiple and broad nulls with the imposed directions by controlling the amplitude-only, six examples of a linear array with 20 isotropic elements have been performed. A 30-dB Chebyshev pattern given in Fig. 3 for 20 equispaced elements with $\lambda/2$ interelement spacing is utilized as the initial pattern.

The BFA parameter values for the all examples given here are listed in Table 1. These values are selected heuristically and verified empirically. The positions of bacteria in the search space represent the antenna array element amplitudes. The fixed chemotactic step size is

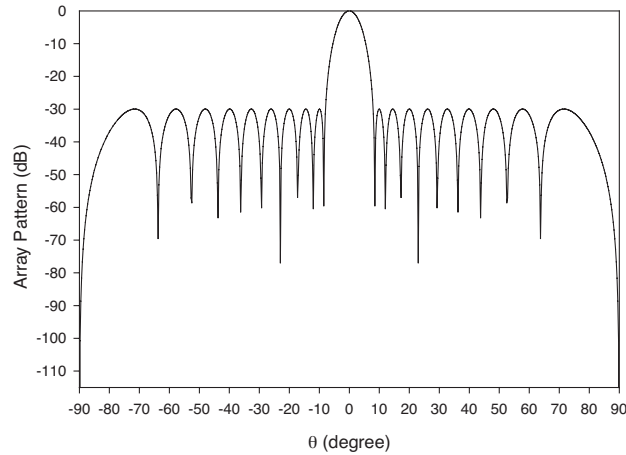


Figure 3. The initial 30-dB Chebyshev pattern.

Table 1. Values of BFA parameters.

Parameters	Values
P	10
S	50
N_c	100
N_s	50
N_{re}	10
N_{ed}	2
P_{ed}	0.2
d_s	1.9
$C(i), i = 1, 2, \dots, S$	0.02

used in the standard BFA. If the chemotactic step size values are too large, the search tends to jump out of the nutrient valley. On the other hand, if the chemotactic step size values are too small, the convergence can be slow [53]. For this reason, in this paper, we used simple variable step sizes that are large in the beginning of the optimization process, and are decreased gradually at the end of the chemotactic loops before the reproduction step. The large step sizes provide fast convergence in the beginning of the optimization process. After some iterations, the step sizes are getting smaller; thus, the intensive searching at the most

promising region of the search space can be done.

The simulation results are obtained within 1–2 minutes on a personal computer with a Core Duo processor running at 1600 MHz with 1 GB RAM. This is sufficient to produce satisfactory patterns with the desired performance on the average. The computation time can be significantly reduced by using faster computer systems. The BFA can also be implemented in real time by using state-of-the art hardware devices, such as FPGAs (Field Programmable Gate Array). In this way, the computation time of the system is limited only by the response time of the FPGA, which is in the order of a few microseconds. C++ programming language was used for implementation of the algorithm. We have run the program for two times to obtain the best result.

In the first example, the Chebyshev pattern with a single null imposed at the direction of the second peak from main beam, which occurs about 14° , is considered. The values of the cost function parameters given in Equation (2) are selected as follows:

$$AF_d(\theta) = \begin{cases} 0, & \text{for } \theta = 14^\circ \\ \text{Initial Chebyshev pattern,} & \text{elsewhere} \end{cases} \quad (4)$$

$$W(\theta) = \begin{cases} 55, & \text{for } \theta = 14^\circ \\ 1, & \text{elsewhere} \end{cases} \quad (5)$$

and

$$ESL(\theta) = \begin{cases} 5, & \text{if } MSL \geq -28.4 \text{ dB} \\ 0, & \text{otherwise} \end{cases} \quad (6)$$

where MSL given in Equation (6) represents the maximum sidelobe level of achieved pattern in the sidelobe region. Fig. 4 shows the nulling pattern obtained from the BFA. As it can be seen from this figure, the pattern closely resembles to the initial Chebyshev pattern except for the nulling direction ($\theta = 14^\circ$). It is also clear from Fig. 4 that the BFA can obtain the pattern with satisfactory null depth level and maximum sidelobe level. Fig. 5 shows the convergence curve. It is apparent from this figure that 2000 iterations are needed to find the optimal solutions.

In the BFA, the elimination and dispersion event happens after a particular number of reproduction processes. A bacterium is chosen, according to the probability P_{ed} , to be eliminated and dispersed to a random location within the search space. These events may prevent the local optima trapping effectively. The effect of the elimination and dispersion on the optimization process is illustrated in Fig. 6 for the first example. In Fig. 6, the positions of bacteria in the search space represent the unnormalized amplitude values of antenna array elements.

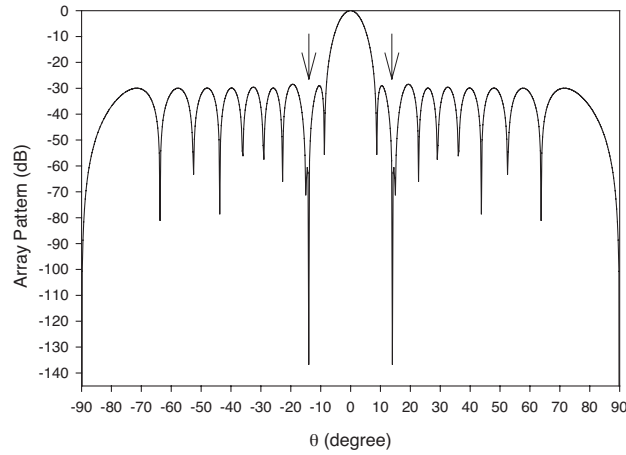


Figure 4. Radiation pattern with one imposed null at 14° .

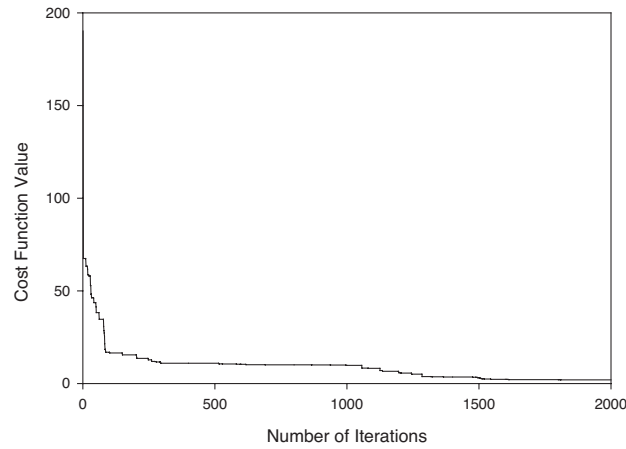


Figure 5. Convergence curve of the nulling pattern achieved by BFA.

In order to show the flexibility of the BFA, in the second example, the restriction is made on the maximum sidelobe level. The value of $ESL(\theta)$ is modified as

$$ESL(\theta) = \begin{cases} 5, & \text{if } MSL \geq -29.3 \text{ dB} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

The nulling pattern with the restricted maximum sidelobe level is then obtained by the BFA and is shown in Fig. 7. The null depth

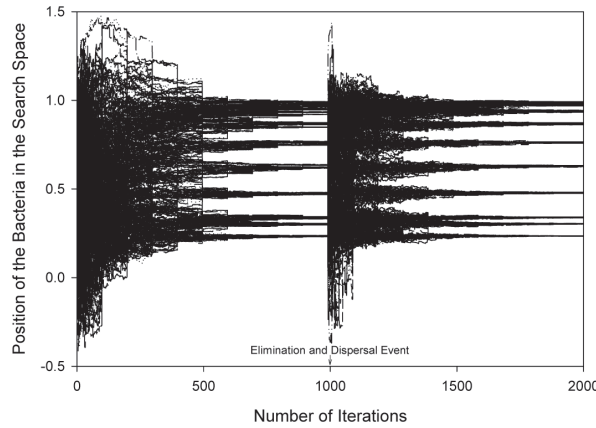


Figure 6. Effect of the elimination and dispersion on the optimization process.

level (NDL), maximum sidelobe level (MSL), and dynamic range ratio ($DRR = |a_{\max}/a_{\min}|$) of the patterns illustrated in Figs. 4 and 7 are given in Table 2. As expected, the maximum sidelobe level performance of the pattern given in Fig. 7 is better than that of the pattern given in Fig. 4. But in response to such an improvement of the maximum sidelobe level, the null depth performance of the pattern in Fig. 7 is worse than that of the pattern in Fig. 4. There is a trade-off between the null depth level and maximum sidelobe level; usually, performance cannot be improved significantly for one without sacrificing the other.

Table 2. NDL, MSL, and DRR of the patterns in Figs. 4 and 7–8.

	Figure 4	Figure 7	Figure 8
NDL (dB)	-136.60	-121.31	-113.56
MSL (dB)	-28.49	-29.33	-28.11
DRR	4.03	4.20	3.89

In this paper, the separation distance value between the adjacent elements is restricted to 0.5λ in order to prevent the mutual coupling effects. The effect of coupling between antenna array elements can also be reduced by minimizing the dynamic range ratio. For this reason, in the third example, the dynamic range ratio is restricted to be 3.89.

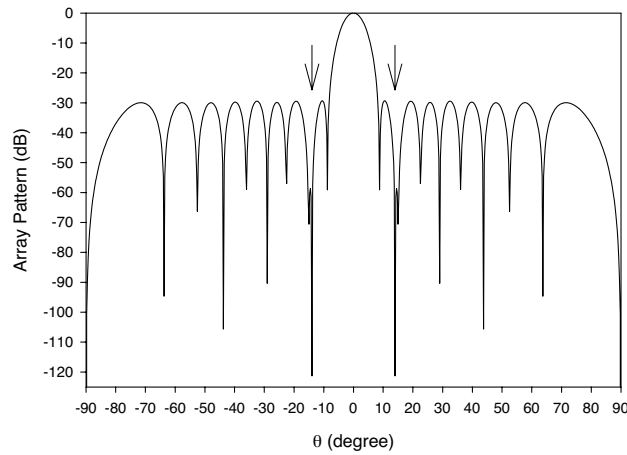


Figure 7. Radiation pattern with the restricted maximum sidelobe level having one imposed null at 14° .

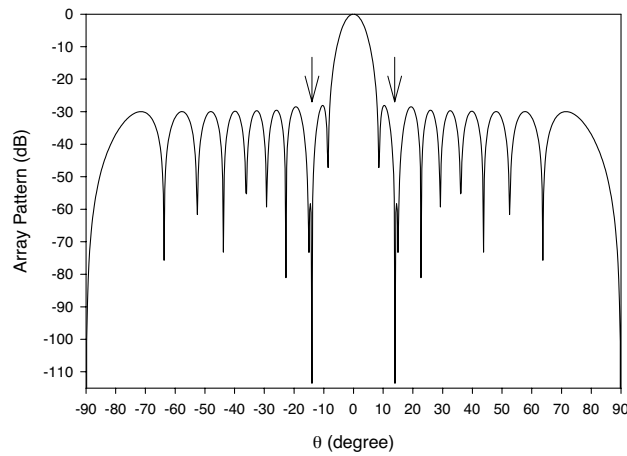


Figure 8. Radiation pattern with the restricted dynamic range ratio having one imposed null at 14° .

The restriction on the dynamic range ratio can easily be achieved by suitably setting the search interval of amplitude weights. The resulting pattern with the restricted dynamic range ratio is shown in Fig. 8. The null depth level, maximum sidelobe level, and dynamic range ratio of the pattern shown in Fig. 8 are also given in Table 2. As expected, the results of this example are worse than those of the previous two

examples because the smaller amplitude range means smaller degrees of freedom for the solution space hence worse sidelobe and null depth performance. The results obtained here illustrate that the null depth level, maximum sidelobe level, and dynamic range ratio of the nulling pattern can easily be controlled by using BFA. The antenna designer should make a trade-off between the achievable and the desired pattern.

The proposed BFA is also capable of synthesizing the pattern with multiple nulls at any desired directions. For this purpose, in the fourth and fifth examples, only the $AF_d(\theta)$ given by Equation (4) is modified to synthesize the patterns with double nulls imposed at the directions of the second and the fourth peaks from main beam (14° and 26°), and with triple nulls imposed at the directions of the second, fourth and fifth peaks from main beam (14° , 26° , and 33°). The patterns with multiple nulls are illustrated in Figs. 9 and 10. It can be seen from Figs. 9 and 10 that all desired nulls are deeper than -85 dB. The results depicted in Figs. 9 and 10 show the ability of the proposed technique for the array pattern synthesis with multiple nulls steered in the prescribed interference directions.

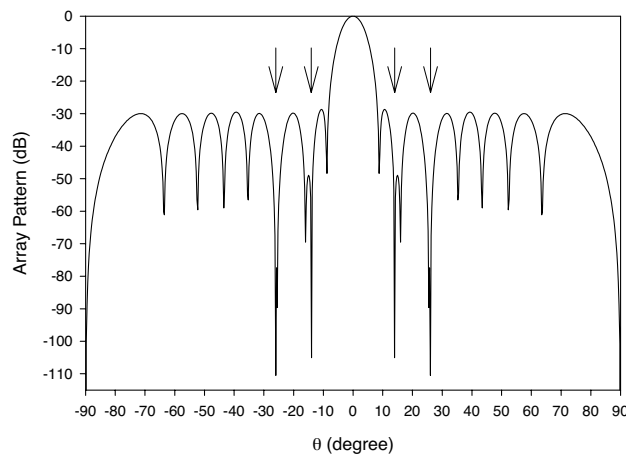


Figure 9. Radiation pattern with double imposed nulls at 14° and 26° .

As the final example, the pattern having a broad null located at 30° with $\Delta\theta = 5^\circ$ is achieved and is shown in Fig. 11. From the figure, a null depth level deeper than -55 dB is obtained over the spatial region of interest. This computer simulation example clearly shows the capacity of BFA to synthesize the array pattern with broad null imposed at the direction of interference.

The element amplitude values normalized according to center

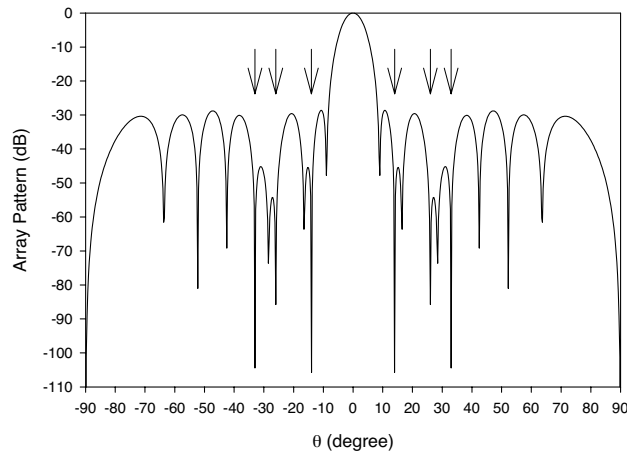


Figure 10. Radiation pattern with triple imposed nulls at 14° , 26° , and 33° .

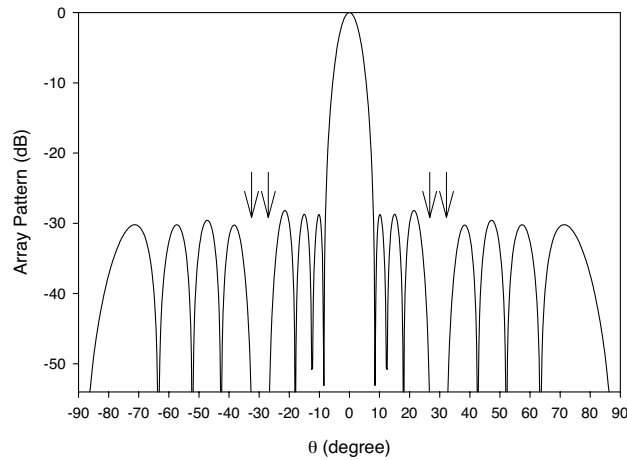


Figure 11. Radiation pattern with a broad null sector centered 30° with $\Delta\theta = 5^\circ$.

elements for the patterns given in Figs. 4 and 7–11 are given in Table 3. It is apparent from Figs. 4 and 7–11 that the patterns are symmetric with respect to the main beam. This is because the symmetry property of the element amplitudes around the array center results in a pattern that is symmetric about the main beam. Therefore, when a null imposed at the one side of the main beam, an image null occurs at

Table 3. Element amplitudes normalized according to center elements for Figs. 4 and 7–11.

k	Initial Chebyshev pattern	Computed with the BFA					
	Figure 3	Figure 4	Figure 7	Figure 8	Figure 9	Figure 10	Figure 11
± 1	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000	1.00000
± 2	0.97010	0.99452	0.98857	0.99249	1.03024	0.99258	0.97404
± 3	0.91243	0.96393	0.95299	0.96429	1.01351	0.96820	0.93899
± 4	0.83102	0.89263	0.87990	0.89894	0.90082	0.90547	0.86503
± 5	0.73147	0.78084	0.77203	0.79107	0.74607	0.74518	0.70220
± 6	0.62034	0.63995	0.63729	0.65034	0.63027	0.56954	0.56285
± 7	0.50461	0.48871	0.48534	0.49786	0.53094	0.51080	0.54207
± 8	0.39104	0.35188	0.34325	0.36075	0.37794	0.40222	0.46450
± 9	0.28558	0.24809	0.23782	0.25686	0.20727	0.18806	0.23882
± 10	0.32561	0.31423	0.30735	0.32027	0.27858	0.23937	0.25372

the other side of the main beam.

The results depicted in Figs. 4 and 7–11 confirm that the BFA proposed in this paper can accurately produce the nulling patterns by controlling the amplitude-only. From the null depth and the maximum sidelobe level points of view, the performances of the patterns are very good. The nulling technique based on BFA preserves the characteristics of the initial Chebyshev pattern with little pattern disturbance except for the nulling directions. The half power beam width of the nulling patterns is almost equal to that of initial Chebyshev pattern. It is apparently demonstrated in this paper that the BFA can be used as an alternative to other antenna synthesis algorithms.

5. CONCLUSIONS

In this paper, the null steering in radiation pattern of a linear antenna array by amplitude-only control has been achieved with the use of BFA, which is a new evolutionary computation technique inspired by the foraging behavior of *E. coli* bacteria in our intestines. The computer simulation results show that the BFA is capable of synthesizing the array patterns with single, multiple and broad nulls imposed at the directions of interferences. A good control over the sidelobe level, dynamic range ratio and null depth level in the interference directions can be achieved. Successful applications show that the proposed technique is simple and easy to implement. Although only linear antenna arrays have been considered here, the BFA can easily be used for arrays with complex geometries as well as nonisotropic-elements. Since the BFA has good accuracy and does not require complicated mathematical functions, it can be very useful to antenna engineers for

the pattern synthesis of antenna arrays. As an evolutionary algorithm, the BFA will most likely be an increasingly attractive alternative, in the electromagnetics and antennas community, to other evolutionary algorithms.

REFERENCES

1. Steyskal, H., R. A. Shore, and R. L. Haupt, "Methods for null control and their effects on the radiation pattern," *IEEE Trans. Antennas Propagat.*, Vol. 34, 404–409, 1986.
2. Er, M. H., "Linear antenna array pattern synthesis with prescribed broad nulls," *IEEE Trans. Antennas Propagat.*, Vol. 38, 1496–1498, 1990.
3. Ibrahim, H. M., "Null steering by real-weight control — A method of decoupling the weights," *IEEE Trans. Antennas Propagat.*, Vol. 39, 1648–1650, 1991.
4. Liao, W. P. and F. L. Chu, "Array pattern synthesis with null steering using genetic algorithms by controlling only the current amplitudes," *Int. J. Electronics*, Vol. 86, 445–457, 1999.
5. Shore, R. A., "Nulling at symmetric pattern location with phase-only weight control," *IEEE Trans. Antennas Propagat.*, Vol. 32, 530–533, 1984.
6. Haupt, R. L., "Phase-only adaptive nulling with a genetic algorithm," *IEEE Trans. Antennas Propagat.*, Vol. 45, 1009–1015, 1997.
7. Ismail, T. H. and M. M. Dawoud, "Null steering in phased arrays by controlling the element positions," *IEEE Trans. Antennas Propagat.*, Vol. 39, 1561–1566, 1991.
8. Tennant, A., M. M. Dawoud, and A. P. Anderson, "Array pattern nulling by element position perturbations using a genetic algorithm," *Electronics Letters*, Vol. 30, 174–176, 1994.
9. Liao, W. P. and F. L. Chu, "Array pattern nulling by phase and position perturbations with the use of the genetic algorithm," *Microwave and Optical Technology Letters*, Vol. 15, 251–256, 1997.
10. Hejres, J. A., "Null steering in phased arrays by controlling the positions of selected elements," *IEEE Trans. Antennas Propagat.*, Vol. 52, 2891–2895, 2004.
11. Abu-Al-Nadi, D. I., T. H. Ismail, and M. J. Mismar, "Interference suppression by element position control of phased arrays using LM algorithm," *Int. J. Electron. Commun.*, Vol. 60, 151–158, 2006.
12. Ares, F., J. A. Rodriguez, E. Villanueva, and S. R. Rengarajan,

- “Genetic algorithms in the design and optimization of antenna array patterns,” *IEEE Trans. Antennas Propagat.*, Vol. 47, 506–510, 1999.
13. Guney, K. and A. Akdagli, “Null steering of linear antenna arrays using modified tabu search algorithm,” *Progress In Electromagnetics Research*, PIER 33, 167–182, 2001.
 14. Karaboga, N., K. Guney, and A. Akdagli, “Null steering of linear antenna arrays by using modified touring ant colony optimization algorithm,” *Int. J. RF and Microwave Computer Aided Eng.*, Vol. 12, 375–383, 2002.
 15. Akdagli, A., K. Guney, and D. Karaboga, “Pattern nulling of linear antenna arrays by controlling only the element positions with the use of improved touring ant colony optimization algorithm,” *Journal of Electromagnetic Waves and Applications*, Vol. 16, 1423–1441, 2002.
 16. Karaboga, D., K. Guney, and A. Akdagli, “Antenna array pattern nulling by controlling both the amplitude and the phase using modified touring ant colony optimisation algorithm,” *Int. J. Electronics*, Vol. 91, 241–251, 2004.
 17. Akdagli, A. and K. Guney, “Null steering of linear antenna arrays by phase perturbations using modified tabu search algorithm,” *J. Communications Technology and Electronics*, Vol. 49, 37–42, 2004.
 18. Khodier, M. M. and C. G. Christodoulou, “Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization,” *IEEE Trans. Antennas Propagat.*, Vol. 53, 2674–2679, 2005.
 19. Yang, S. W., Y. B. Gan, and A. Y. Qing, “Antenna-array pattern nulling using a differential evolution algorithm,” *Int. J. RF and Microwave Computer Aided Eng.*, Vol. 14, 57–63, 2004.
 20. Chung, Y. C. and R. L. Haupt, “Amplitude and phase adaptive nulling with a genetic algorithm,” *Journal of Electromagnetic Waves and Applications*, Vol. 14, 631–649, 2000.
 21. Lu, Y. and B. K. Yeo, “Adaptive wide null steering for digital beamforming array with the complex coded genetic algorithm,” *IEEE Int. Conf. Phased Array Systems and Technology*, 557–560, Dana Point CA, USA, 2000.
 22. Mouhamadou, M., P. Vaudon, and M. Rammal, “Smart antenna array patterns synthesis: Null steering and multi-user beamforming by phase control,” *Progress In Electromagnetics Research*, PIER 60, 95–106, 2006.

23. Babayigit, B., A. Akdagli, and K. Guney, "A clonal selection algorithm for null synthesizing of linear antenna arrays by amplitude control," *Journal of Electromagnetic Waves and Applications*, Vol. 20, 1007–1020, 2006.
24. Mouhamadou, M., P. Armand, P. Vaudon, and M. Rammal, "Interference suppression of the linear antenna arrays controlled by phase with use of SQP algorithm," *Progress In Electromagnetics Research*, PIER 59, 251–265, 2006.
25. Guney, K. and M. Onay, "Amplitude-only pattern nulling of linear antenna arrays with the use of bees algorithm," *Progress In Electromagnetics Research*, PIER 70, 21–36, 2007.
26. Guney, K. and M. Onay, "Bees algorithm for null synthesizing of linear antenna arrays by controlling only the element positions," *Neural Network World*, Vol. 16, 153–169, 2007.
27. Yan, K. K. and Y. L. Lu, "Sidelobe reduction in array-pattern synthesis using genetic algorithm," *IEEE Trans. Antennas Propagat.*, Vol. 45, 1117–1122, 1997.
28. Baskar, S., A. Alphones, and P. N. Suganthan, "Genetic-algorithm-based design of a reconfigurable antenna array with discrete phase shifters," *Microwave and Optical Technology Letters*, Vol. 45, 461–465, 2005.
29. Mitilineos, S. A., S. C. A. Thomopoulos, and C. N. Capsalis, "Genetic design of dual-band, switched-beam dipole arrays, with elements failure correction, retaining constant excitation coefficients," *J. Electromagnetic Waves and Applications*, Vol. 20, 1925–1942, 2006.
30. Ayestarán, R. G., J. Laviada, and F. Las-Heras, "Synthesis of passive-dipole arrays with a genetic-neural hybrid method," *J. Electromagnetic Waves and Applications*, Vol. 20, 2123–2135, 2006.
31. Jin, J., H. L. Wang, W. M. Zhu, and Y. Z. Liu, "Array patterns synthesizing using genetic algorithm," *Progress In Electromagnetics Research Symposium*, 64–68, Cambridge, USA, March 2006.
32. Mahanti, G. K., S. Das, and A. Chakraborty, "Design of phase-differentiated reconfigurable array antennas with minimum dynamic range ratio," *IEEE Antennas and Wireless Propagat. Letters*, Vol. 5, 262–264, 2006.
33. Tonn, D. A. and R. Bansal, "Reduction of sidelobe levels in interrupted phased array antennas by means of a genetic algorithm," *Int. J. RF and Microwave Computer Aided Eng.*, Vol. 17, 134–141, 2007.

34. Mahanti, G. K., N. Pathak, and P. Mahanti, "Synthesis of thinned linear antenna arrays with fixed sidelobe level using real-coded genetic algorithm," *Progress In Electromagnetics Research*, PIER 75, 319–328, 2007.
35. Mahanti, G. K., A. Chakrabarty, and S. Das, "Phase-only and amplitude-phase synthesis of dual-pattern linear antenna arrays using floating-point genetic algorithms," *Progress In Electromagnetics Research*, PIER 68, 247–259, 2007.
36. Mahanti, G. K., A. Chakrabarty, and S. Das, "Design of fully digital controlled reconfigurable array antennas with fixed dynamic range ratio," *Journal of Electromagnetic Waves and Applications*, Vol. 21, 97–106, 2007.
37. Xu, Z., H. Li, Q. Z. Liu, and J. Y. Li, "Pattern synthesis of conformal antenna array by the hybrid genetic algorithm," *Progress In Electromagnetics Research*, PIER 79, 75–90, 2008.
38. Akdagli, A., K. Guney, and D. Karaboga, "Touring ant colony optimization algorithm for shaped-beam pattern synthesis of linear antenna arrays," *Electromagnetics*, Vol. 26, 615–628, 2006.
39. Hosseini, S. A. and Z. Atlasbaf, "Optimization of side lobe level and fixing quasi-nulls in both of the sum and difference patterns by using continuous ant colony optimization (ACO) method," *Progress In Electromagnetics Research*, PIER 79, 321–337, 2008.
40. Gies, D. and Y. Rahmat-Samii, "Particle swarm optimization for reconfigurable phase-differentiated array design," *Microwave and Optical Technology Letters*, Vol. 38, 168–175, 2003.
41. Boeringer, D. W. and D. H. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," *IEEE Trans. Antennas Propagat.*, Vol. 52, 771–779, 2004.
42. Lee, K. C. and J. Y. Jhang, "Application of particle swarm algorithm to the optimization of unequally spaced antenna arrays," *Journal of Electromagnetic Waves and Applications*, Vol. 20, 2001–2012, 2006.
43. Chen, T. B., Y. L. Dong, Y. C. Jiao, and F. S. Zhang, "Synthesis of circular antenna array using crossed particle swarm optimization algorithm," *Journal of Electromagnetic Waves and Applications*, Vol. 20, 1785–1795, 2006.
44. Mahmoud, K. R., M. El-Adawy, S. M. M. Ibrahim, R. Bansal, and S. H. Zainud-Deen, "A comparison between circular and hexagonal array geometries for smart antenna systems using particle swarm optimization algorithm," *Progress In Electromagnetics Research*, PIER 72, 75–90, 2007.

45. Jin, N. B. and Y. Rahmat-Samii, "Advances in particle swarm optimization for antenna designs: Real-number, binary, single-objective and multiobjective implementations," *IEEE Trans. Antennas Propagat.*, Vol. 55, 556–567, 2007.
46. Mikki, S. M. and A. A. Kishk, "Physical theory for particle swarm optimization," *Progress In Electromagnetics Research*, PIER 75, 171–207, 2007.
47. Akdagli, A. and K. Guney, "Shaped-beam pattern synthesis of equally and unequally spaced linear antenna arrays using a modified tabu search algorithm," *Microwave and Optical Technology Letters*, Vol. 36, 16–20, 2003.
48. Yang, S. W., Y. B. Gan, and A. Y. Qing, "Sideband suppression in time-modulated linear arrays by the differential evolution algorithm," *IEEE Antennas and Wireless Propagation Lett.*, Vol. 1, 173–175, 2002.
49. Kurup, D. G., M. Himdi, and A. Rydberg, "Synthesis of uniform amplitude unequally spaced antenna arrays using the differential evolution algorithm," *IEEE Trans. Antennas Propagat.*, Vol. 51, 2210–2217, 2003.
50. Yang, S. W., Z. P. Nie, and Y. J. Wu, "A practical array pattern synthesis approach including mutual coupling effects," *Electromagnetics*, Vol. 27, 53–63, 2007.
51. Guney, K., A. Akdagli, and B. Babayigit, "Shaped-beam pattern synthesis of linear antenna arrays with the use of a clonal selection algorithm," *Neural Network World*, Vol. 16, 489–501, 2006.
52. Akdagli, A., K. Guney, and B. Babayigit, "Clonal selection algorithm for design of reconfigurable antenna array with discrete phase shifters," *Journal of Electromagnetic Waves and Applications*, Vol. 21, 215–227, 2007.
53. Passino, K. M., "Biomimicry of bacterial foraging," *IEEE Control Systems Magazine*, Vol. 22, 52–67, 2002.
54. Mishra, S., "Hybrid least-square adaptive bacterial foraging strategy for harmonic estimation," *IEE Proc. - Generation Transmission and Distribution*, Vol. 152, 379–389, 2005.
55. Mishra, S., "A hybrid least square-fuzzy bacterial foraging strategy for harmonic estimation," *IEEE Trans. Evolutionary Computation*, Vol. 9, 61–73, 2005.
56. Lin, W. and P. X. Liu, "Hammerstein model identification based on bacterial foraging," *Electronics Letters*, Vol. 42, 1332–1334, 2006.

57. Kim, D. H. and J. H. Cho, "A biologically inspired intelligent PID controller tuning for AVR systems," *Int. J. Control Automation and Systems*, Vol. 4, 624–636, 2006.
58. Niu, B., Y. Zhu, X. He, and X. Zeng, "Optimum design of PID controllers using only a germ of intelligence," *6th World Congress on Intelligent Control and Automation*, 3584–3588, Dalian, China, June 2006.
59. Tang, W. J., M. S. Li, S. He, Q. H. Wu, and J. R. Saunders, "Optimal power flow with dynamic loads using bacterial foraging algorithm," *2006 Int. Conf. on Power System Technology*, 1–5, Chongqing, China, October 2006.
60. Majhi, B. and G. Panda, "Recovery of digital information using bacterial foraging optimization based nonlinear channel equalizers," *First Int. Conf. on Digital Information Management*, 367–372, Bangalore, India, Dec. 2006.
61. Ulagammai, M., P. Venkatesh, P. S. Kannan, and N. P. Padhy, "Application of bacterial foraging technique trained artificial and wavelet neural networks in load forecasting," *Neurocomputing*, Vol. 70, 2659–2667, 2007.
62. Kim, D. H., A. Abraham, and J. H. Cho, "A hybrid genetic algorithm and bacterial foraging approach for global optimization," *Information Sciences*, Vol. 177, 3918–3937, 2007.
63. Mishra, S. and C. N. Bhende, "Bacterial foraging technique-based optimized active power filter for load compensation," *IEEE Trans. Power Delivery*, Vol. 22, 457–465, 2007.