Progress In Electromagnetics Research, PIER 83, 185–198, 2008

WEIGHTS OPTIMIZATION OF NEURAL NETWORK VIA IMPROVED BCO APPROACH

Y. Zhang and L. Wu

School of Information Science and Engineering
Southeast University
China

Abstract—Feed forward neural Network (FNN) has been widely applied to many fields because of its ability to closely approximate unknown function to any degree of desired accuracy. Back Propagation (BP) is the most general learning algorithms, but is subject to local optimal convergence and poor performance even on simple problems when forecasting out of samples. Thus, we proposed an improved Bacterial Chemotaxis Optimization (BCO) approach as a possible alternative to the problematic BP algorithm, along with a novel adaptive search strategy to improve the efficiency of the traditional BCO. Taking the classical XOR problem and *sinc* function approximation as examples, comparisons were implemented. The results demonstrate that our algorithm is obviously superior in convergence rate and precision compared with other training algorithms, such as Genetic Algorithm (GA) and Taboo Search (TS).

1. INTRODUCTION

Multi-layer Feed forward Neural Network (FNN) is the most popular and widely applied Neural Network (NN) due to is superior ability of non-linearity to approximate unknown function to any degree of desired accuracy, which has been widely applied to many fields, such as pattern recognition, image processing, financial prediction, and signal and information processing, especially in the filed of the Electromagnetics: Mohamed used RBF to optimize and characterize the electromagnetical coupled patch antennas [1]. Geney used neural networks (NN) to calculate the characteristic impedance of air-suspended trapezoidal and rectangular-shaped microshield lines [2]. Ayestar chose NN and source reconstruction to transform the near field to far filed [3] and synthesize non uniform-antenna array [4]. Kizilay adopt NN to identify and classify the cylindrical targets [5]. Zainud-Deen applied RBF to estimate the direction of arrival and the state of polarization [6]. Panda applied FNN to simulate a multiple cavity model of 2D phased array [7].

In order to hasten the convergence of NN, most applications of FNNs use some variation of the gradient technique, such as Back Propagation (BP) to optimize the coefficients of NN [8]. However, Lawrence et al. [9] pointed out that, when the training of a BP tends to be difficult due to the noise of data, then the networks fall into a naïve solution such as always predicting the most common output. Miao et al. [10] indicated that the BP solutions are usually forced to the local minimum due to the gradient descent algorithm used to get weights of connections. Engoziner et al. [11] presented that BP use some variation of the gradient technique, which is essentially a local optimizing method and thus has some inevitable drawbacks, such as easily trapping into the local optimal and dissatisfying generalization capability. Sexton et al. [12] proposed the fact that the gradient descent algorithm may perform poorly even on simple problems when predicting the holdout data. And document [13] suggested that, in interest of mitigating the above limitation, weighted values and thresholds of neurons in BP are optimized by global search algorithms. Artificial Intelligence (AI) is a powerful global search algorithm, which had been widely used in electromagnetics. Chiu employed genetic algorithm (GA) to reduce path loss in urban area [14]. Tian employed GA on ultraconveniently finding multiple solutions of complex transcendental [15]. Lu betaked GA to optimize the broadband top-load antenna [16]. Chen betaked GA to image 3D buried objects [17]. From papers above, it is obvious that AI is a mighty tool for optimization.

Consequently, newly research tends to hybridize several artificial intelligence (AI) techniques to improve the performance. Tsaih et al. [18] integrated the rule-based technique and ANNs. Kohara et al. [19] incorporated prior knowledge, Gao [20] incorporated Niche Genetic Algorithm (NGA), and Chen et al. [21] united Immune Programming (IP) and Gene Expression Programming (GEP) to improve the learning process of the conventional ANN. Miao et al. [10] adopted Bacterial Colony Radial Basis Function Neural Network (RBFNN) and Majhi et al. [22] utilized Bacterial Foraging Optimization (BFQ) to ameliorate the performance of traditional ANN. He et al. [23] presented an adaptive Tabu Search (TS) approach as a possible alternative to the BP algorithm. Ayestaran [24] bestowed genetic-neural hybrid method (GN) to synthesize the passive-dipole arrays.

In recent years, the popularity of Bacterial Chemo-taxis

Optimization (BCO) has grown significantly as a new global search technique and it has achieved widespread success in solving practical optimization problems in different domains [25–27]. This article extended the line of the research by applying an improved BCO (IBCO for short) strategy to the difficult problem of NN optimization. Taking the classical XOR problem and *sinc* function approximation as tests. The results show that our BCO algorithm is obviously superior in convergence, precision, and generalization compared with the traditional BP algorithms.

2. THE ERROR BACK-PROPAGATION NEURAL NETWORK

BP, one of the most popular techniques in the filed of NN, is a kind of supervised learning neural network, the principle behind which involves using the steepest gradient descent method to reach any small approximations. A general model of the BP has a structure depicted in Figure 1.

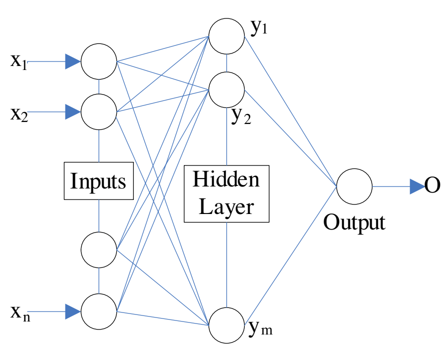


Figure 1. The architecture of BP.

Here we can see there are three layers contained in BP: input layer, hidden layer, and output layer. Two nodes of each adjacent layer are directly connected, which is called a link. Each link has a weighted value, which presents the relational degree between two nodes [28]. Assume that there are n input neurons, m hidden neurons, and 1 output neuron, from which we can infer the training process described by the following equations to update these weighted values, which can be divided into two steps:

I) Hidden layer stage: The outputs of all neurons in the hidden layer are calculated by following steps:

$$net_j = \sum_{i=0}^{n} v_{ij} x_i \quad j = 1, 2, \cdots, m \tag{1}$$

$$y_j = f_H(net_j) \quad j = 1, 2, \cdots, m \tag{2}$$

Here net_j is the activation value of the jth node, y_j is the output of the hidden layer, and f_H is called the activation function of a node, usually a sigmoid function as follows:

$$f_H(x) = \frac{1}{1 + \exp(-x)} \tag{3}$$

II) Output Stage: The outputs of all neurons in the output layer are given as follows:

$$O = f_O\left(\sum_{j=0}^{m} \omega_{jk} y_j\right) \tag{4}$$

Here f_O is the activation function, usually a line function. All weights are assigned with random values initially, and are modified by the delta rule according to the learning samples traditionally.

3. CONVENTIONAL BACTERIAL CHEMOTAXIS OPTIMIZATION

The optimization based on Bacterial Chemotaxis [25, 26] (BC) was inspired from bacterial foraging behavior, pioneered by Bremermann [25] and his coworkers, and proposed by analogy to the way bacteria react to chemo-attractants in concentration gradients. Sibylle D Muller [26] extracted the BC algorithm from the newest production found in biology field, which was testified to excel other optimization algorithms.

3.1. Description of the 2-D Model

Dahlquist et al. [26, 27] model the motion of a single bacterium in two dimensions by making the following assumptions.

1) The path of a bacterium is a sequence of straight-line trajectories.
2) All trajectories have the same constant speed.
3) When a bacterium turns, its choice of the new direction, the angle between two successive trajectories, and the duration of a trajectory are all regulated by a probability distribution.
4) The probability distributions for both the angle and the duration are independent of parameters of the previous trajectory.

3.2. Algorithm Steps of BCO

The processing of BCO is presented as follows:

STEP1: Compute the velocity of a bacterium v, which is assumed to be a scalar constant value 1.

STEP2: Compute the duration of the trajectory τ, the distribution of which satisfies the exponential probability density function (PDF)

$$P(X=\tau)=\frac{1}{T}\exp\left(-\frac{\tau}{T}\right) \tag{5}$$

where the expectation value $E(X)=T$ and the variance $Var(X)=T^2$. The time T is given by

$$T=\begin{cases} T_0, & \text{for } \dfrac{f_{pr}}{l_{pr}}\geq 0 \\ T_0\left(1+b\left|\dfrac{f_{pr}}{l_{pr}}\right|\right), & \text{for } \dfrac{f_{pr}}{l_{pr}}<0 \end{cases} \tag{6}$$

where T_0 presents the minimal mean time; f_{pr} presents the difference between the actual and the previous function value; l_{pr} presents the vector connecting the previous and the actual position in the parameter space; and b is assumed as the dimensionless parameter.

STEP3: Compute the new direction. The PDF of the angle α between the previous and the new direction is Gaussian and read, for turning right or left, respectively as follows:

$$P(X=\alpha, v=\pm\mu)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{(\alpha-v)^2}{2\sigma^2}\right] \tag{7}$$

where, the expectation value $\mu=E(X)$ and variance $\sigma=\sqrt{Var(X)}$ are given by:

If $\frac{f_{pr}}{l_{pr}}<0$, and then

$$\mu=62^\circ(1-\cos\theta);\quad \sigma=26^\circ(1-\cos\theta);\quad \cos\theta=\exp(-\tau_c\tau_{pr}) \tag{8}$$

Else $\frac{f_{pr}}{l_{pr}}\geq 0$, and then $\mu=62^\circ$, $\sigma=26^\circ$.

Where τ_c presents the correlation time, and τ_{pr} presents the duration of the previous step. The choice of a right or left direction as referring to the previous trajectory is determined using a uniform PDF, thereby yielding a PDF for the angle α

$$P(X=\alpha)=\frac{1}{2}\left[P(X=\alpha, v=\mu)+P(X=\alpha, v=-\mu)\right] \tag{9}$$

STEP4: Compute the new position,

$$x_{new} = x_{old} + n_u l \tag{10}$$

Here, x_{new} presents the new position of the bacteria; x_{old} presents its previous position; n_u presents the normalized new direction vector; and l presents the length of the new trajectory.

In summary, the algorithm contains the following parameters to be computed in advance: T_0, τ_c and b. Document [26] gives their detailed formula

$$T_0 = \varepsilon^{0.30} 10^{-1.73};\ b = T_0 \left(T_0^{-1.54} 10^{0.60}\right);\ \tau_c = \left(\frac{b}{T_0}\right)^{0.31} 10^{1.16} \tag{11}$$

4. AN IMPROVED BCO STRATEGY

4.1. Mechanism of the Improvement

In the optimization of BCO, every bacterium imparts information each other for ameliorating the foraging environment, nevertheless it will beget the whole colony to fall into local nutrients [29] and debilitated to cross over noxious substance. As to the algorithm, it means the searching will be trapped into local extrama.

No life-form is living single in realism. Although bacteria are microscopic and primeval, there are by all means correlations between different individuals and colonies. Investigation showed that Escherichia coli aggregate at the foraging process, and disparate colonies exchange the food information while keep certain distance from each other [30]. They are more capable of surviving because of the enhancement of comprehension on the settings around.

4.2. Detailed Improvement measures

As stated above, improvements are present here, encompassing the following features which can be separated as two stages, as shown in Figure 2.

I) Infra-colony Phase [31]

Since every bacterium seizes limited intelligence, and has the capability to regulate its locomotion by information perceived from approximate bacteria, aggregate several bacteria into one colony, which abide following pattern due to the description on the biome community distribute behavior:

I-I) Before every new motion, bacteria should perceive surroundings to validate the existence of more nutrient areas. If there are, bacteria are more likely to transfer to the centroid of these areas.

$$\begin{aligned} Center(x_{i,k}) =& Aver\left(x_{j,k} \middle| f(x_{j,k}) < f(x_{i,k}) \right. \\ & \left. \text{and } dis(x_{j,k}, x_{i,k}) < SenseLimit\right) \\ Aver(x_1, x_2, \cdots x_n) =& \left(\sum_{i=1}^{n} x_i\right) / n \end{aligned} \tag{12}$$

where k presents motion step, i presents the bacterium index, $dis(x_{j,k}, x_{i,k})$ presents the distance between bacteria i and j.

I-II) If a bacterium trends to shift to the center of its accompanies around, the length of the new trajectory is $rand() \bullet dis(x_{i,k}, center(x_{i,k}))$, where $rand()$ uniformly distribute at the range of (0, 2).

While bacteria perceive the chemo-attractants vary little for a session, they will migrate by diverse forms in pursuit of more nutrient nourishment [30]. So after continuous n_e steps with absolute differences of functional values are less than given threshold ε_e, the bacteria colony will migrate to a new place. Through the migration, the multiformity of the bacteria colony can be preserved, and the ability of bouncing out of local minima can be fortified.

II) Inter-colony Phase

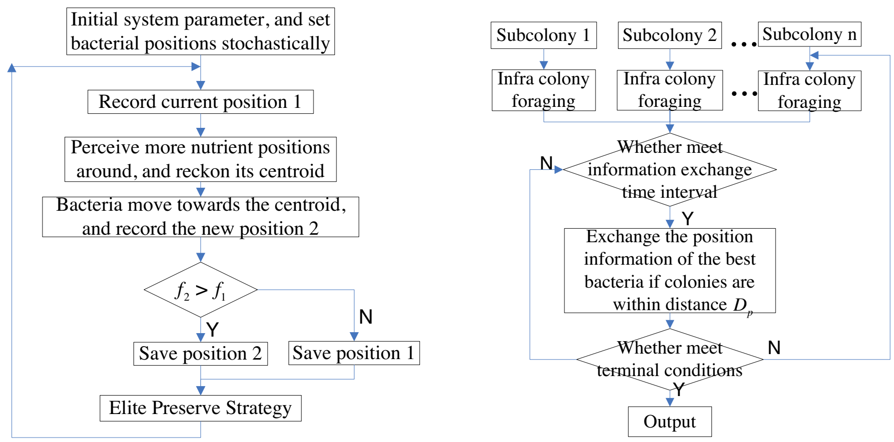
Divide the whole bacteria colony into several sub-colonies, and each one cast about for food independently. If two sub-colonies encounter each other with mutual distance less than given threshold L ($L >> SenseLimit$), the sub-colony with poor performance should be punished to migrate. In this way it can be assured that only one sub-colony exists at the round with diameter L, which prevents the whole colony from congregating at one point, and maintains its diversity of augmenting the global search ability of origin algorithm.

5. THE IBCO FOR TRAINING OF FNNS

MSE (mean squared error) is selected as the search objective of our approach, and it is detailed as follow:

$$MSE = \sum_{i=1}^{N} (true - output)^2 \tag{13}$$

Here *true* presents the authentic values which are already known to users, *output* is the output values of the BP after BCO training, and N presents the number of samples. Our goal is to minimize the *MSE* through BCO.



(a) intra-colony foraging (b) inter-colony foraging

Figure 2. Two stages of the flow chart of IBCO.

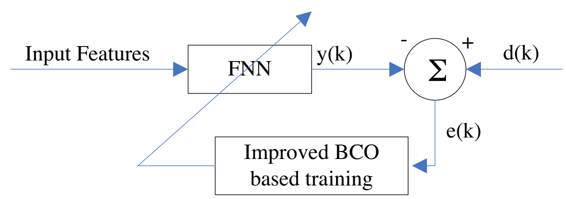


Figure 3. The structure of IBCO for optimization of FNN weights.

6. EXPERIMENTS AND DISCUSSIONS

6.1. XOR Problem

In order to examine the feasibility and validity of IBCO technique, XOR problem was firstly taken as a test. The FNN was set to 2-2-1. The former two layers used the sigmoid transfer function *logsig* and the output layer used the linear transfer function *purelin*. Error function was set to *MSE*. The initial weights were all set to real values randomly picked out from the range $[-1, +1]$.

Some parameters in BCO was set to several different combinations, detailed in Table 1, where NB denotes the number of bacteria; SC denotes the size of a colony; MIS denotes the maximum iterative steps; EGP denotes the error goal precision.

The experiment had been performed for 100 runs for each group, whose results were presented in Table 2, where the data of Simple BP

Table 1. The combination of some parameters in IBCO.

Group	NB	SC	MIS	EGP
1	20	5	2000	1e-6
2	25	5	2000	1e-6
3	30	5	2000	1e-6

Table 2. Comparison of results for XOR between IBCO and other algorithms.

Learning algorithm	Simple BP	Steepest BP	Super Linear BP	Adaptive TS	BCO	IBCO
Maximum learning steps	8000	8000	8000	2000	2000	2000
Error precision	1e-3	1e-3	1e-3	1e-6	1e-6	1e-6
Convergence rate	89%	86%	82%	94%	92%	98%

algorithm, Steepest algorithm, Super Linear BP algorithm, and TS algorithm were extracted from document [23]

6.2. *sinc* Function Approximation

In interest of examining the generalization capability of our IBCO technique, a comparison to BP algorithm with momentum was implemented for sinc function approximation ($f(x) = \sin(x)/x$). In our experiments, both the BP algorithm and BCO or IBCO based BP network architecture were set to 1-20-1; the transfer function for the hidden layer and the output layer was *tansig* and *purelin*, respectively; error function was set to MSE; the maximum iterative steps were set to 1000; the error goal precision was set to 10^{-6}; and the initial weights are all picked out randomly. The approximation comparison is shown in Figure 4. It is evident that the approximation effect of IBCO is better than that of other algorithms.

6.3. Multi-Dimensional Function Approximation

Our last experiment is to verity the performance of our proposed algorithm on the multi-dimension function approximation. Here we choose the following functions as test

$$f_1(x) = \sum\nolimits_{i=1}^{10} x_i^2, \quad x_i \in [-1, 1] \tag{14}$$

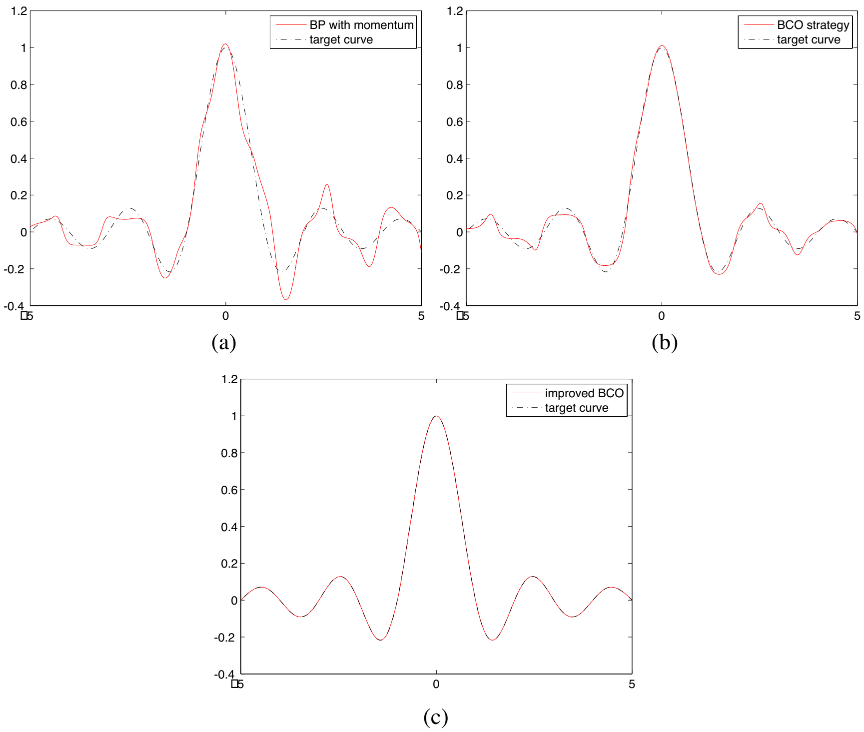


Figure 4. Comparison of *sinc* function approximation. (a) BP with momentum; mse = 3.268e − 3, (b) BCO strategy; mse = 2.909e − 4, (c) improved BCO; mse = 9.93846e − 7.

$$f_2(x) = \sum_{i=1}^{10} \left[(x_i^2 - x_{i+1})^2 + (1 - x_i)^2\right], \quad x_i \in [-5\ 5] \tag{15}$$

All network architecture were set to 10-100-1; the transfer function for the hidden layer and the output layer was *tansig* and *purelin*, respectively; error function was set to *MSE*; the maximum iterative steps were set to 1000; the error goal precision was set to 10^{-6}; and the initial weights are all picked out randomly. The training data was obtained by 500 uniformly distributed samples.

Table 3 shows the comparison of multi dimensional function approximation by BP with momentum, BCO based BP, and IBCO based BP.

It is obvious from Table 3 that concerning our proposed algorithm overmatches other two algorithms on the function $f_1(x)$ and $f_2(x)$. Since there were not enough steps, IBCO based BP did not converge at given precision, however it still performed best at three of all.

Table 3. Comparison of multi-dimensional function approximation.

	BP with momentum	BCO based BP	IBCO based BP
$f_1(x)$	0.452059	4.22362e-5	1.04658e-6
$f_2(x)$	9.14633	1.10718e-3	3.03194e-4

7. CONCLUSION

It can be deduced from the experiments that our IBCO strategy is obviously superior in convergence rate and precision, and better generalization capability compared with other training algorithms.

This article also illustrates the importance of using global search techniques for optimizing neural networks, and it also demonstrates that the IBCO should be regarded as a possible solution to the hard optimization problems.

REFERENCES

1. Mohamed, M. A., E. A. Soliman, and M. A. El-Gamal, "Optimization and characterization of electromagnetically coupled patch antennas using RBF neural networks," *J. of Electromagn. Waves and Appl.*, Vol. 20, No. 8, 1101–1114, 2006.
2. Guney, K., C. Yildiz, S. Kaya, and M. Turkmen, "Artificial neural networks for calculating the characteristic impedance of air-suspended trapezoidal and rectangular-shaped microshield lines," *J. of Electromagn. Waves and Appl.*, Vol. 20, No. 9, 1161–1174, 2006.
3. Ayestar, R. G. and F. Las-Heras, "Near filed to far field transformation using neural networks and source reconstruction," *J. of Electromagn. Waves and Appl.*, Vol. 20, No. 15, 2201–2213, 2006.
4. Ayestar, R. G., F. Las-Heras, and J. A. Martinez, "Non uniform-antenna array synthesis using neural networks," *J. of Electromagn. Waves and Appl.*, Vol. 21, No. 8, 1001–1011, 2007.
5. Kizilay, A. and S. Makal, "A neural network solution for identification and classification of cylindrical targets above perfectly conducting flat surfaces," *J. of Electromagn. Waves and Appl.*, Vol. 21, No. 14, 2147–2156, 2007.
6. Zainud-Deen, S. H., H. A. Malhat, K. H. Awadalla, and E. S. El-Hadad, "Direction of arrival and state of polarization estimation

using radial basis function neural network (RBFNN)," *Progress In Electromagnetics Research B*, Vol. 2, 137–150, 2008.

7. Panda, D. K. and A. Chakrabarty, "Multiple cavity modeling of a feed network for two dimensional phased array application," *Progress In Electromagnetics Research Letters*, Vol. 2, 135–140, 2008.
8. Engoziner, S. and E. Tomsen, "An accelerated learning algorithm for multiplayer perception: Optimization layer by layer," *IEEE Trans. on Neural Network*, Vol. 6, 31–42, 1995.
9. Lawrence, S., A. C. Tsoi, and C. L. Giles, "Noisy time series prediction using symbolic representation and recurrent neural network grammatical inference," Technical report UMIACS-TR-96-27 and CS-TR-3625, Institute for Advanced Computer Studies, University of Maryland, 1996
10. Miao, K., F. Chen, and Z. G. Zhao, "Stock price forecast based on bacterial colony RBF neural network," *Journal of QingDao University*, Vol. 20, 50–54, 2007.
11. Engoziner, S. and E. Tomesn, "An accelerated learning algorithm for multiplayer perception: Optimization layer by layer," *IEEE Trans. on Neural Network*, Vol. 6, 31–42, 1995.
12. Sexton, R. S., B. Alidaee, R. E. Dorsey, and J. D. Johnson, "Global optimization for artificial neural networks: A tabu search application," *European Journal of Operational Research*, Vol. 106, 570–584, 1998.
13. Sexton, R. S., R. E. Dorsey, and J. D. Johnson, "Toward global optimization of neural networks: A comparison of the genetic algorithm and back-propagation," *Decision Support System*, Vol. 22, 171–185, 1998.
14. Chiu, C.-C. and T.-C. Tu, "Path loss reduction in an urban area by genetic algorithms," *J. of Electromagn. Waves and Appl.*, Vol. 20, No. 3, 319–330, 2006.
15. Tian, Y.-B. and J. Qian, "Ultraconveniently finding multiple solutions of complex transcendental equations based on genetic algorithm," *J. of Electromagn. Waves and Appl.*, Vol. 20, No. 4, 475–488, 2006.
16. Lu, Y.-Q. and J.-Y. Li, "Optimization of broadband top-load antenna using micro-genetic algorithm," *J. of Electromagn. Waves and Appl.*, Vol. 20, No. 6, 793–801, 2006.
17. Chen, X., D. Liang, and K. Huang, "Microwave imaging 3-D buried objects using parallel genetic algorithm with FDTD technique," *J. of Electromagn. Waves and Appl.*, Vol. 20, No. 13,

1761–1774, 2006.

18. Tsaih, R., Y. Hsu, and C. C. Lai, "Forecasting S&P 500 stock index futures with a hybrid AI system," *Decision Support System*, Vol. 23, 161–174, 1998.
19. Kohara, K., T. Ishikawa, Y. Fukuhara, and Y. Nakamura, "Stock price prediction using prior knowledge and neural networks," *Int. J. Intelligence System Accounting Finance Manage*, Vol. 6, 11–12, 1997.
20. Gao, X. J., "Artificial neural network based on niche genetic algorithm applies to stock price prediction," *Market Modernization*, Vol. 500, 35–35, 2007.
21. Chen, F., Y. H. Chen, and J. Z. Zhang, "Stock index modeling using IP based gene expression programming," *Computer Engineering and Applications*, Vol. 43, 227–229, 2007.
22. Majhi, R. and G. Panda, "Stock market prediction of S&P 500 and DJIA using bacterial foraging optimization technique," *IEEE Congress on Evolutionary Computation*, 2007.
23. He, Y., Y. H. Qiu, G. Liu, and K. Y. Lei, "Optimizing weights of neural network using an adaptive tabu search approach," *Proc. ISNN 2005*, LNCS 3496, 672–676, 2005.
24. Ayestaran, R. G., J. Laviada-Martinez, and F. Las-Heras, "Synthesis of passive-dipole arrays with a genetic-neural hybrid method," *J. of Electromagn. Waves and Appl.*, Vol. 20, No. 15, 2123–2135, 2006..
25. Bremermann, H. J., "Chemotaxis and optimization," *J. Franklin Inst.*, Vol. 297, 397–404, 1974.
26. Sibylle, D. M., M. Jarno, A. Stefane, et al., "Optimization based on bacterial chemotaxis," *IEEE Trans. on Evolutionary Computation*, Vol. 6, 17–19, 2002.
27. Dahlquist, F. W., R. A. Elwell, and P. S. Lovely, "Studies of bacterial chemotaxis in defined concentration gradients — A model for chemotaxis toward l-serine," *J. Supramolecular Structure*, Vol. 4, 329–342, 1976.
28. Basma, A. A. and N. Kallas, "Modeling soil collapse by artificial neural networks," *Geotechnical and Geological Engineering*, Vol. 22, 427–438, 2004.
29. Dong, H. K., A. Ajith, and H. C. Jae, "A hybrid genetic algorithm and bacterial foraging approach for global optimization," *Information Sciences*, Vol. 177, 3918–3937, 2007.
30. Passino, K. M., "Biomimicry of bacterial foraging for distributed optimization and control," *IEEE Control System Magazine*, Vol. 6,

52–67, 2007.

31. Li, W. W., H. Wang, Z. J. Zou, and J. X. Qian, "Function optimization method based on bacterial colony chemotaxis," *Journal of Circuits and Systems*, Vol. 10, 58–63, 2005.