# Agent based Adaptive Firefly Back-propagation Neural Network Training Method for Dynamic Systems


Sudarshan Nandy
Department of Engineering & Technological Studies
University of Kalyani, Kalyani, Nadia,
West Bengal, India.
sudarshannandy@gmail.com

Manoj Karmakar
Department of Computer Science and Engineering
JIS college of Engineering,Kalyani, Nadia,
*West Bengal, India.*
manojk549@gmail.com

Partha Pratim Sarkar
Department of Engineering & Technological Studies,
University of kalyani, Kalyani, Nadia,
West Bengal,India.
parthabe91@yahoo.co.in

Achintya Das
Department of Eletronics & Tele. Comm. Engineering
Kalyani Govt. Engg. College, Kalyani, Nadia,
West Bengal, India.
achintya.das123@gmail.com

Ajith Abraham
Machine Intelligence Research Labs (MIR Labs)
Aubum, Washington 98071-2259, USA
IT For Innovations - Center of excellence,VSB -
Technical University of Ostrava,Czech Republic.
ajith.abraham@ieee.org

Diptarup Paul
Tech Mahindra Limited, Plot-1,Phase-III,
Rajiv Gandhi Infotech Park, Pune-411057.
diptarup.paul@gmail.com



***Abstract*— Nature Inspired meta-heuristic algorithms are one of the most efficient solution to many engineering optimization problems. The Firefly algorithm is one of the nature inspired solution. The objective of the proposed work is of two folds. In the first fold the firefly algorithm is applied to the back-propagation training phase to optimize the overall training process. One of the problem in this type of implementation is the adjustment of algorithmic parameters and number of firefly population, and for a dynamic system the manual modification of parameter is a troublesome matter . In the second fold, the proposed work is implemented a statistical hypothesis based agent which is adaptively control the various parameters and number of firefly populations in firefly algorithm based back-propagation method and this makes it more convenient for dynamic systems. The effectiveness of automatic parameter adjustment over the performance of algorithm is analyzed through correct classification rate and sum of squared error. The proposed method is tested over five bench mark non-linear standard data set and it is compared with genetic algorithm based back-propagation method. It is observed from the experiment that the agent automatically adjust the parameters and number of firefly populations in each iteration of the back-propagation optimization phase and it is finally converged within a minimum number of iteration.**



***Keywords-Firefly algorithm, Back-propagation training optimization, meta-heuristic algorithm, statistical hypothesis.***


## I. Introduction

Nature Inspired meta heuristic algorithms are simulates the biological or social phenomena of mother nature to find a solution for many NP-hard combinatorial optimization problem. One of the most important fact in this type of algorithm is that the merits of various natural life processes are used to find the solution. Numerical and computational technique are used in the algorithm for validating the philosophy of those inherited concept of natural life.

On the basis of types of different solution available at the end of iteration, the meta-heuristic algorithms are classified into two categories. In the first category, the algorithms are known as a deterministic and hill-climbing is one of the example of this kind. The Second category is classified as stochastic algorithm and this type of algorithm starts with same initial parameters but provides different solutions. The deterministic algorithms are initiated with same parameters and always produce a same set of output. So there is a small difference between deterministic and stochastic algorithms because in case of stochastic meta-heuristic algorithm, for same set of input the outputs are different. However this kind of algorithms are converged with the same solution. Particle swarm optimization(PSO)[9][10], ant colony optimization(ACO)[4][2], Firefly algorithm(FA)[6][7][8], genetic algorithm[19], differential evaluation algorithm[5] etc. are the examples of the stochastic meta-heuristic algorithm. The number of population is one of the key factor for this type of algorithm and each of this algorithms are initiated with a set of population. Different solution are available by increasing or decreasing the number of population. Apart from the number of population, other parameters are also responsible for the convergence. This kind of parameter adjustment is a problematic matter for dynamic system. The algorithmic performance of firefly may be the best in its kind but it also suffers from the problem of adjustment of its population and other related parameters[3].

The back-propagation method is one of the most widely implemented technique to optimize feed-forward neural network training procedure. In the back-propagation method

978-1-4673-5115-7  

the training of neural network is started with some initial parameter like learning rate , weight and bias and then those values are updated until the convergence of the back-propagation algorithm. The procedure is also known as a steepest descent back-propagation algorithm or SDBP[18]. The learning procedure of this algorithm follows the gradient and generally trapped into the local minimum. The issue of convergence in back-propagation is really important and to solve this problem various methods are developed. The development of the algorithmic procedure to optimize the back-propagation neural network training can be categorized into three part. In the first part the parameters along with the learning rates are updated in an ad-hoc manner[11][12][13]. Variable learning back-propagation(VLBP) and Momentum based back-propagation (MOBP) are the important procedure in this category. The numerical analysis techniques are categorized in the second part of development. In this phase of development the influences of first or second order derivative is observed. The levenberg-marquardt algorithm and conjugate algorithm is the most famous method to optimize the back-propagation training procedure. The conjugate back-propagation methods[14] is the second order derivative free algorithm. The convergence rate of this algorithm is better than that of the other algorithm in its kind. The levenberg-marquardt algorithm[15][17][18] is converted to the gauss-newton method when the algorithm is converged. This training optimization procedure takes a lot of memory for processing for which this procedure is slow but it is converged surely. The third part of the development is basically based on the hybrid optimization technique. In this phase of the development the optimization procedure of back-propagation method is embedded with other nature inspired solution. The genetic algorithmic procedure is use to update the weight and bias value of back-propagation optimization phase. This hybrid implementation is used over wide range of application and also suffer from slow convergence rate due to large search space. The selection of the parent depends on the performance of the procedure and the chance of selection as a parent is not equal for all the solution. The ant-colony optimization (ACO), particle swarm optimization(PSO)[20][21][22], artificial bee colony(ABC),genetic algorithm(GA)[16] are also used with the neural network to optimize the training procedure. The artificial bee colony algorithm is also used with the back-propagation optimization phase to optimize the training performance[23]. Hybrid firefly optimization procedure is proposed with cellular automata based learning technique[1].

The hybrid methods of the back-propagation training optimization suffers from the manual adjustment of initial population and the parameters related to the convergence of the algorithm. The proposed work uses an agent model to take the decision on the number of population of firefly and its related parameter for the convergence of the firefly back-propagation algorithm. The agent uses a statistical hypothesis based analysis and it is implemented in the critic part of the model to make a convenient adjustment to the parameters and number of firefly population. It is found from the experiment that the statistical hypothesis based agents decision on the parameter adjustment provides a better convergence speed for the back-propagation training process.

## II. Firefly Algorithm

One of the attractions for many researchers today has been various nature-inspired algorithms in solving tough mathematical optimization problems. Algorithms such as bat algorithm, firefly algorithm, ant colony optimization, particle swarm optimization have been accepted as having the needed potential to solve optimization problems easily.

Here the behavior of fireflies that has urged the researchers in incorporating the concepts in solving optimization problems is presented. Everyone is conversant with the flashing light of fireflies in the night sky. But there is a larger scenario behind the flashing of the light. There are two thousand firefly species and most of them are seen to be generating constant short and rhythmic flashes. And this pattern varies with each species differently. The process of bioluminescence is responsible for the production of such light. Although debatable but there exists some postulates which state two fundamental functions of such a signaling system. The main functions of the rhythmic flash are to attract members from the opposite sex for mating and to find potential prey.

The rhythmic flash brings both sexes together for mating. Females readily respond to the flashing pattern of their male counterparts in the same species. However there exists some species such as photuris in which the females can mimic the pattern displayed by the males to lure and eat them. The male fireflies in such instances make the mistake of deeming the flashing pattern of the females to be a potential invitation for mating.

It is known that the intensity of light is inversely proportional to the square of the distance from the source of light. In addition to that light is also absorbed by air and in this process the intensity becomes weaker and weaker with distance. In spite of that, at night the light is quite suitable for fireflies to communicate. Proper implementation of this behavior of fireflies can be achieved by associating the pattern of flashing of light to an objective function as the one to be optimized and this makes it possible to formulate the so-called firefly algorithm.

In implementing the firefly algorithm the following rules are believed to be the idealized ones:

1. Here the different species of fireflies is not considered and all the fireflies are thought to be unisex so that one firefly responds to the flashing pattern of the other fireflies regardless of its sex. It is favorable to consider all the fireflies to be a single species otherwise different algorithms would have to be devised for each of the different species which is quite a tedious job.

2. The firefly with the weaker intensity of flash is moved towards one with a brighter flash. The reason being attraction is directly proportional to the brightness of the flash. As a result when the distance between two fireflies increases the attraction between the two is readily decreased. If the brightness of any two fireflies are the same then they won't move towards each other but will continue to fly randomly. As the flash with the higher intensity is regarded as the potential signal for mating so,it is quite natural to put forward such a concept. Two fireflies with the same intensity

can not distinguish the feature and hence are not attracted to one another.

3. The brightness of the firefly is completely dependent upon and it is determined by the designed objective function. As the objective function for each of the optimization problems has to be designed keeping the respective problem at hand so the brightness will also be changed accordingly.

In the firefly algorithm there are two important things to consider: one is the determination of the light intensity and the other one is to formulate the concept of attractiveness. However since the attraction between two fireflies depends completely on the perception of light emanating from the other so the measure of attraction will vary with the distance between firefly $x_i$ and firefly $y_j$ . The measure of attraction can be calculated suing the following formula:

$$\wp = \wp_{0\,.}\, e^{\left(-\gamma\,.\,r^2\right)} \quad (1)$$

where,

$\wp$ = *attractivenes*
$\wp_0$ =*initial* attractivenes
$r$=*distance* between two fireflies
$\gamma$=*light* absorption coefficient

The distance between two fireflies can be calculated as follows:

$$r_{(i,k)} = \|x_i - y_i\| = \sqrt{\sum_{k=1}^{d} \left(x_{(i,k)} - y_{(i,k)}\right)} \quad (2)$$

where, d = dimension

Now, according to the firefly algorithm the movement of firefly $x_i$ to the firefly $y_j$ can be calculated as follows:

$$x_i = x_i + \wp_0 e^{\left(-\gamma r^2\right)} + \alpha\left(rand - \frac{1}{2}\right) \quad (3)$$

The firefly movement depends on the attractiveness and is also a key factor for the matting. The last part of the eq.3 is randomization parameters where,

$rand \in (0,1)$ and $\alpha \in (0,1)$

The value of initial attractiveness varies in each iteration and light absorption coefficient value is assigned to a fixed value. In the proposed method those parameters values are adjusted by the agent and hence the algorithm converge quickly. The number of population for the proposed method is always grater than to one.

## III. Proposed Methdology

The proposed method is implemented in two stage. In the first stage the conventional firefly algorithm is incorporated into the back-propagation training optimization phase and then in the second stage the agent module is attached with the proposed method. The agent module is responsible to modify the parameters for the convergence of this algorithm. The overall procedure is then automatically adjusted its own parameters and converged within a finite period. The multi layer feed-forward neural network considered for implementing the proposed method is depicted in the fig. 1.

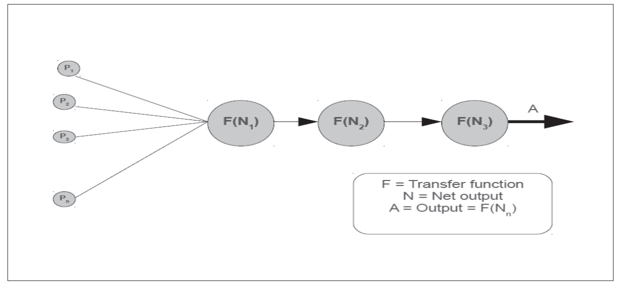


Figure 1. Feed-forward Neural network structure for simulating the back-propagation method.

In the initial stages the weight and bias of each node is considered randomly between (0,1). So,

$$Weight \in \{rand(0,1)\}$$

and

$$Bias \in \{rand(0,1)\}$$

Now, the weight and bias values are used to simulate the error for every input through back-propagation training procedure. The net output of each layer is considered as N and can be computed through the following formula:

$$N^{(r+1)}(m) = \sum_{n=1}^{k} \left(WL^{(r+1)}(m,n)\, a_{n-1} + BL^{(r+1)}\right) \quad (4)$$

Where, N is the net output, n is the number of layer, WL and BL is the weight and bias of a neuron.

Now the actual output $A_k$ of $k^{th}$ layer is calculated as follows:

$$A_k(m) = F_k\left(N_k(m)\right) \quad (5)$$

The proposed method considers the average sum of squared as the performance index and it is calculated as follows:

$$S_\mu(x) = \frac{\sum_{j=1}^{N} S_F(x)}{P_i} \quad (6)$$

where, $S_\mu$ is the average performance, $S_F$ is the performance index, P is the number of firefly population in $i^{th}$ iteration.

The performance index $S_f(x)$ is calculated by the following equation:

$$s_f(x) = \sum_{i=1}^{l} \left(t_i - a_i\right)^T . \left(t_i - a_i\right)$$

$$s_f(x) = \sum_{i=1}^{l} e^T . e \quad (7)$$

where,

$$t_i = i^{th}\ target,\ a_i = i^{th}\ output,\ e = Error$$

So, the $S_F(x)$ is calculated as:

$$S_F(x) = \sum_{k=1}^{q} s_f(x)^T . s_f(x) \quad (8)$$

The weight and bias are calculated according to the back-propagation method. The sensitivity of one layer is calculated from its previous one and the calculation of sensitivity starts from the last layer of the network and move backward.

Now, at the end of each iteration the list of average sum of squared error of $i^{th}$ iteration $SSE_i$ can be computed as:

$$SSE_i = \{S_\mu(x_1), S_\mu(x_2), S_\mu(x_3), \ldots, S_\mu(x_N)\} \quad (9)$$

The attractive firefly is replicate the minimum sum of squared error and it is found when all the inputs are processed for each population of the firefly. So the firefly $x_j$ is calculated as :

$$x_j = Min\{S_\mu(x_1), S_\mu(x_2), S_\mu(x_3), \ldots, S_\mu(x_N)\} \quad (10)$$

and rest of the average sum of squared is considered as other firefly. So, the movement of other firefly ($x_i$) towards $x_j$ can be derived from the eq.3 and it can be written as:

$$\Delta X_i = x_i + \varphi_0 e^{(-\gamma r^2)}(x_j - x_i) + \alpha(rand - \frac{1}{2}) \quad (11)$$

where , $\Delta X_i$ small movement of $x_i$ towards $x_j$. The weight and bias of the each layer then adjusted as :

$$WL^{(r+1)}_{(r,k)} = W^n_{(r,k)} - \Delta X_i \quad (12)$$

and

$$B^{(r+1)}_{(r,k)} = B^n_{(r,k)} - \Delta X_i \quad (13)$$

Now the proposed method introduce an agent to keep watch on the performance index of the method. Then at the end of a iteration it calculates the squared difference between new and old variance ($E(v_s)$ ) as follows:

$$E(v_s) = \sqrt{(v_{old} - v_{new})^2} \quad (14)$$

where, v is the variance and it is calculated as:

$$V = \frac{\sigma}{VS_\mu(x)} \quad (15)$$

The $VS_\mu(x)$ can be calculated as:

$$VS_\mu(X) = \frac{\sum_{i=1}^{N} S_\mu(X)}{N} \quad (16)$$

and the σ is calculated as :

$$\sigma = \frac{\sqrt{N(\sum_{i=1}^{N} S^2_\mu(x)) - (\sum_{i=1}^{N} S_\mu(x))^2}}{N} \quad (17)$$

Where N is the population size of firefly.

Now, the critic of the agent is actually identify the polarity of the performance as:

$$EE = f(E(v_s)) = \begin{Bmatrix} 1 \\ 0 \end{Bmatrix}$$

If EE is 1 then counter is decreased by one and otherwise it increases by one. If counter, c > 2 then the agent adjusts the parameters and this process continue up-to the convergence of the proposed method. In the experiment it is observed that the number of population is increased when the performance index is high and then it is stable on a fixed number of iteration.

## IV. Evalution

The performance index of the proposed method is the average sum of squared error. The correct classification rate is also considered as a another criteria of convergence. The proposed method is tested with three of the benchmark data set.

### A. Experimental setup

The agent based firefly back-propagation method is developed using the python programming language and python-matplotlib is used to plot the analyzed result. The simulation of this method is performed with the Intel P4 core2Duo machine with 1.66GHz processor and 1GB of memory. Python numerical library and scientific library are used for the numerical processing.

The data sets are non-linear and consists of different classes. In order to analyses the performance of the algorithm following data set are processed:

1. Iris [24];
2. Wine[25];
3. Glass[26].
4. Liver Disorder[27]
5. Soybean[28]

### B. Experimental Result Analysis

The proposed method is tested over the glass, iris and wine data set. The convergence criteria is the correct classification rate and sum of squared error. The maximum cycle number for the proposed method is 50. In the fig.2 the analyzed result of glass data set training is depicted. The proposed method is achieved to the 97.66% of correct classification accuracy. The effect of the agent based decision is depicted in the 4th sub-plot of fig.2. It is observed from the graph that the training methods required 8 number of firefly to achieve the 97.66% of correct classification rate. The iris data set are training analysis is depicted in the fig.3. The agent based adaptive firefly back-propagation neural network training shows that the correct classification rate for the iris data set is 97.8%. The number of firefly population which is decided by the proposed method is 9 and this number of population is the minimum requirement to achieve the 97.8% correct classification rate.

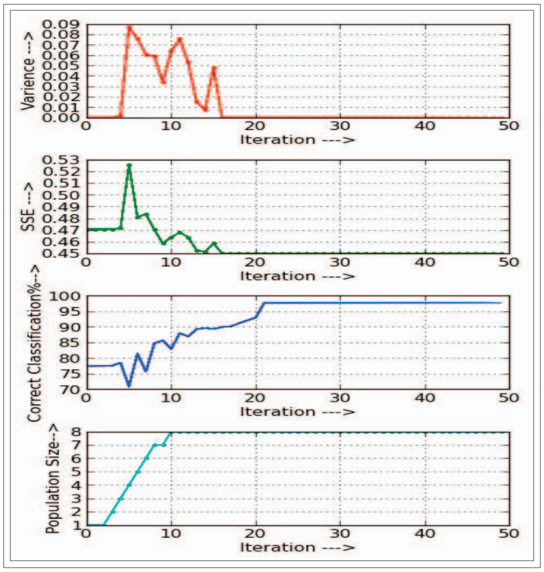


Figure 2. Glass Data set training analysis.

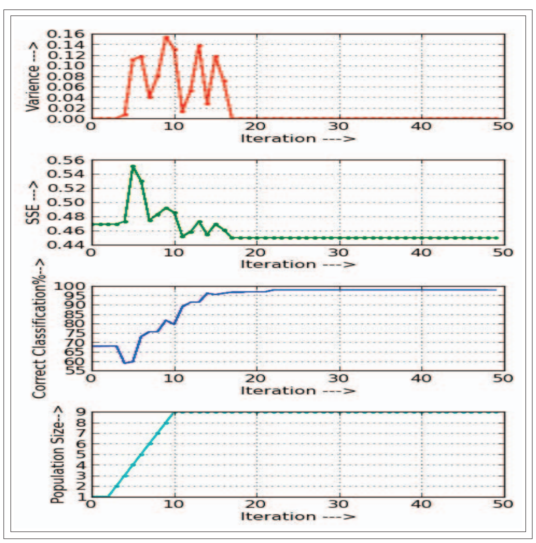


Figure 3. Iris Data set are trained with proposed method.

TABLE I. ANALYSIS PROPOSED METHOD(AFFBPNN) OVER STANDARD DATA SET

| Sl. No. | Data Set | Correct Classification (%) | SSE | No. of Firefly (automaticall y adjusted) |
|---|---|---|---|---|
| 1 | Glass | 97.66 | 0.45 | 8 |
| 2 | Iris | 97.8 | 0.46 | 9 |
| 3 | Wine | 98.1 | 0.44 | 5 |
| 4 | Liver Disorder | 99.42 | 0.3 | 4 |
| 5 | Soybean | 91.48 | 0.73 | 13 |

TABLE II. COMPARATIVE ANALYSIS OF PROPOSED METHOD (AFFBPNN) WITH GENETIC ALGORITHM BASED BACK-PROPAGATION TRAINING(GABPNN)

| Data set | Algorithm | Correct Classification (%) | SSE |
|---|---|---|---|
| Glass | GABPNN | 97.2 | 0.51 |
| | AFFBPNN | 97.66 | 0.45 |
| Iris | GABPNN | 96.6 | 8.29 |
| | AFFBPNN | 97.8 | 0.46 |
| Wine | GABPNN | 97.4 | 0.51 |
| | AFFBPNN | 98.1 | 0.44 |
| Liver Disorder | GABPNN | 99.13 | 10.14 |
| | AFFBPNN | 99.42 | 0.3 |
| Soybean | GABPNN | 86.8 | 28.08 |
| | AFFBPNN | 91.48 | 0.73 |

The analysis of the wine data set is depicted in the fig.4. In case of the wine data set the proposed method

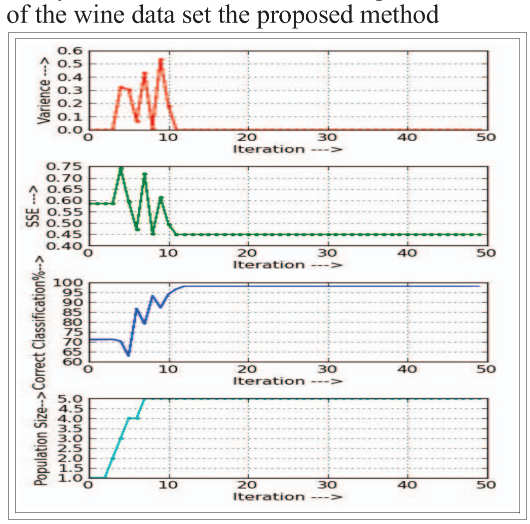


Figure 4. Wine Data set are trained with proposed method.

automatically adjusted the number of firefly to 5 and gradually reaches o the 98.1% of correct classification accuracy. The correct classification rate and SSE of liver and soybean data set are also observed from the Table I. The correct classification rate is 91.48% and 9.42 for soybean and liver data set and this is achieved with the 4 and 13 number of automatically adjusted firefly population.

The agent based firefly based back-propagation (AFFBPNN) is compared with the genetic algorithm based back-propagation neural network training method. The comparative analysis of proposed method and GABPNN is given in the Table II. The performance of proposed method(AFFBPNN) in terms of correct classification rate and SSE is improved than the compared one.

## V. Conclusion

The agent based firefly back-propagation neural network method is implemented the firefly algorithm to back-propagation method for optimization of training procedure and then implement a statistical hypothesis based agent model to adjust its parameters and number of firefly population which makes it applicable to the real time systems. The average correct classification rate for the five benchmark data set is 96.89%. The method achieve its correct classification rate when the population number is stable. It is observed from the experiment that if the number of population adjusted manually then it causes the memory of computer. In the proposed method the number of firefly population is less and corresponding parameters is modified on the basis of optimization performance. Thus proposed method can be implemented to any dynamic system. The agent based firefly back-propagation algorithm is compared with GABPNN and the performance is analyzed on the basis of 1. sum of squared error or solution quality 2. classification error percentage. It is observed that the algorithm outperform the compared one over all of the tested problems and hence the efficiency of the proposed method gets established to be improved.

## References


[1] Hassanzadeh, Tahereh,(2012) “A new hybrid algorithm based on Firefly Algorithm and cellular learning automata”,20th Iranian Conference on Electrical Engineering (ICEE), pp. 628 - 633.

[2] Dorigo M. & Stützle T.,(2004) “Aant Colony optimization”, MIT Press, Cambridge.

[3] Gaige Wang, Lihong Guo, Hong Duan, Luo Liu and Heqi Wang, (2012),“A Modified Firefly Algorithm for UCAV Path Planning”, International Journal of Hybrid Information Technology, Vol. 5, No. 3, July, 2012, pp. 123-144.

[4] Guang-Feng Deng & Woo-Tsong lin, (2011) “ Ant colony optimization-based algorithm for airline crew scheduling problem”, International Journal of Expert System with Applications, 38, pp. 5787-5793.

[5] K.V. Price, R.M. Storn, J.A. Lampinen (Eds.),(2005) “Differential Evolution: A Practical Approach to Global Optimization”, Springer Natural Computing Series.

[6] X.-S. Yang, (2010) "Firefly Algorithm, Stochastic Test Functions and Design Optimisation", Int. J. Bio-Inspired Computation, Vol. 2, No. 2, pp.78–84.

[7] X.-S. Yang, (2009)“Firefly algorithms for multimodal optimiza-tion”, in: Stochastic Algorithms: Foundations and Applications, SAGA 2009, Lecture Notes in Computer Sciences, Vol. 5792, pp. 169-178 .

[8] X.-S. Yang, (2010) “Firefly algorithm, L´evy flights and global op-timization”, in: Research and Development in Intelligent Systems XXVI (Eds M. Bramer, R. Ellis, M. Petridis), Springer London, pp. 209-218 .

[9] J. Kennedy, R.C. Eberhart, (1995) “Particle swarm optimization”, Proceedings of the IEEE International Conference on Neural Networks, vol. 4, pp. 1942–1948.

[10] Y. Fukuyama, S. Takayama, Y. Nakanishi, H. Yoshida, (1999)“A particle swarm optimization for reactive power and voltage control in electric power systems”, Genetic and Evolutionary Computation Conference,pp. 1523–1528.

[11] Samad, T., 1990, “Back-propagation improvements based on heuristic arguments”, Proceedings of International Joint Conference on Neural Networks, Washington, 1, pp. 565-568.

[12] Sperduti, A. & Starita, A. , 1993, “Speed up learning and network optimization with Extended Back-propagation”, Neural Networks, 6, pp. 365-383.

[13] Van Ooten A. , Nienhuis B, 1992, “Improving the convergence of the back-propagation algorithm”, Neural Networks, 5, pp. 465-471.

[14] C. Charalambous, 1992, “Conjugate gradient algorithm for efficent training of neural networks”, IEEE Procedings-G , vol. 139, 3.

[15] Levenberg, K., 1944, “A method for the solution of certain problem in least squares”, Quart. Appl. Math., 2, pp.164-168.

[16] David J. Montana and Lawrence Davis, (1989) "Training a Feed-forward Neural Networks using Genetic Algorithms", Journal of Machine Learning, pp. 762-767.

[17] Marquardt , D. , 1963, “ An algorithm for least sqare estimation of nonlinear parameters”, SIAM J. Appl. Math., 11, pp. 431-441.

[18] T. Hagan and M. B. Menhaj, 1994, “Training feedforward networks with the Marquardt algorithm,” IEEE Trans. Neural Netw., vol. 5, no. 6, pp.989–993.

[19] J.H. Holland,( 1975) “Adaptation in Natural and Artificial Systems”, University of Michigan Press, Ann Arbor, MI.

[20] Celal Ozturk & Devis Karaboga, (2011) “ Hybrid artificial bee colony algorithm for neural network training”, IEEE Congress on Evolutionary Computing, pp. 84-88.

[21] P. Y. Kumbhar & S. Krishnan, (2011), “Use of artificial bee colony algorithm in artificial neural network synthesis”, International Journal of Advanced Engineering Sciences and Technologies, Vol. No. 11, 1, pp. 162-171.

[22] D.Karaboga,B.Akay and C.Ozturk(2007),“Artificial Bee Colony (ABC) Optimization Algorithm for Training Feed-Forward Neural Networks”,MDAI '07 Proceedings of the 4th international conference on Modeling Decisions for Artificial Intelligence , pp.318-329

[23] J. Zhang, J. Zhang, T. Lok & M.Lyu, (2007) “ A hybrid particle swarm optimization backpropagation algorithm for feed forward neural network training” , Applied Mathematics and Computation, Vol. 185, pp. 1026-1037.

[24] UCI Machine Learning Repository : Iris Data Set - http://archive.ics.uci.edu/ml/datasets/Iris

[25] UCI Machine Learning Repository : Wine Data Set - http://archive.ics.uci.edu/ml/datasets/Wine

[26] UCI Machine Learning Repository : Glass Identification Data Set http://archive.ics.uci.edu/ml/datasets/Glass+Identification.

[27] UCI Machine Learning Repository : Liver Disorder Data Set http://archive.ics.uci.edu/ml/datasets/Liver+Disorders.

[28] UCI Machine Learning Repository : Soybean Data Set http://archive.ics.uci.edu/ml/datasets/Soybean+%28Small %29.