

A COMPARATIVE STUDY ON NEURAL NET CLASSIFIER OPTIMIZATIONS

Subhendu Sekhar Behera¹, Sangeeta Bhanja Chaudhuri², and Subhagata
Chattopadhyay³

¹BTech & ²Asst. Professor, Department of Computer Science & Engineering, National
Institute of Science & Technology, Berhampur, India

³Professor, Department of Computer Science & Engineering, Bankura Unnayani Institute of
Engineering, Bankura, India

ABSTRACT

This paper studies the performances of three optimization algorithms, such as (a) Back Propagation (BP) that works by the principle of gradient descent, (b) Simulated Annealing (SA), which is a probabilistic search and (c) a Binary-Coded Genetic Algorithm (BCGA) that is an evolutionary technique for training of a Feed-Forward Neural Net Classifier (NNC). The objective is to compare the performances of BP, SA, and BCGA in terms of yielding accurate outputs in less time. Four standard datasets, e.g. Iris, Diabetes, Glass and Teaching Assistant Evaluation (TAE) are used for their implementations and testing. It has been observed that BP loses its merit as a good NNC optimizer due to its tendency of being trapped into local minima. SA, on the other hand, consumes more time for convergence, but yields better classification accuracy, compared to BP. BCGA, being a global search algorithm, is able to find better solution from all available solutions and thus yields the best accuracy, but with slow convergence rate. Based on these observations, the paper suggests that SA might be a reasonable choice for NNC optimization to start with, when both accuracy and convergence speed are considered.

KEYWORDS: *Neural Network, Back Propagation, Simulated Annealing, Binary Coded Genetic Algorithm, Classification & Optimization*

I. INTRODUCTION

Feed-Forward Neural Net Classifier (NNC) is an artificial neural network structure with input signals constrained to propagate in the forward direction [1, 2]. It finds frequent applications in the fields like Estimation of Software Effort [3], Software Reliability Modelling [4], Dynamic System Identification and Control [5], Voice Recognition [6], and many more. So, it urges for the optimization of feed-forward NNC with the best accuracy and convergence. It is not a simple task and requires comprehensive parametric studies on various datasets. This leads us to the implementation of Back Propagation (BP), Simulated Annealing (SA) algorithms and Binary-Coded Genetic Algorithm (BCGA), which are three popular optimizers to train and test an NNC.

Among several algorithms to facilitate NNC's supervised training process, back propagation (BP) is the most popularly used one. It has been used in trade prediction [7], energy saving in wireless sensor networks [8], analysis of LHM elemental basic structure [9], weather forecasting [10,11], identification of seismic arrival types [12], microstrip filters [13], medical diagnoses [14,15] etc. Its goal is to optimize the weight set associated with NNC through iterations [16]. Nevertheless, its nature of getting stuck at local minima limits the possibility of obtaining the optimal solution and sometimes fails to produce better accuracy though with faster convergence.

Simulated annealing (SA), on the other hand, is a probabilistic global search algorithm. It states the convergence of the physical process of crystallization that involves stabilization of molecules with

slow decrement in temperature [17]. Similar to BP, optimal weight set is the key focus of SA, which is accomplished with the controlled decrement of the temperature. SA also finds applications in the field of Chromatographic Separation [18], Water Distribution System [19], Structural Optimization [20], Design of Low Pass Filters [21], and so on. Being a global search it may lead to the optimal solution with considerable convergence rate, yet it is probabilistic in nature.

Genetic Algorithm (GA) is an analogy to the process of natural evolution that is used for optimization [22]. The different types of GAs are Messy GA [23], Adaptive GA [24], Parallel GA [25], Independent Sampling GA [26], Real-Coded GA [27], and many more. Here BCGA is implemented where the individuals in population are encoded with binary digits [25]. Some of its important contributions include: antenna design [28, 29], pattern classification [30], pipeline flow optimization [25], medical diagnosis [31, 32, 33] and so forth. One of its most important characteristic is it can find better solution using the information available in current solutions with the operations such as cross-over and mutation whereas the convergence rate is not up to the mark.

There have been a few comparative studies performed between BCGA and BP [34, 35] and SA and BP [36] on data classification using neural network on the basis of accuracy. But in classification task, convergence rate also plays a major role along with accuracy in real-world problems. So the analysis in our study includes: parametric optimization of NNC training, classification accuracy check and estimating the convergence rate.

The objective of this work is to compare these three different classes of optimization algorithms on the basis of accuracy and convergence rate while training an NNC. The accuracy reflects the minimum testing error and convergence rate is based upon the time taken for completion of training. The above three algorithms are operated on *Iris*, *Diabetes*, *Glass*, and *Teaching Assistant Evaluation* (TAE) datasets [37].

Rest of the paper is organized as follows. Section II describes the methodology used, section III explains results, and the paper is concluded in section IV.

II. METHODOLOGY

This section briefly describes the data, tools, and techniques used in the study.

2.1. Data

Four standard datasets, such as **Iris** (150 x 4; Class: 1-3), Teaching Assistant Evaluation (**TAE**: 149 x 5; Class: 1-3), **Glass** (214 x 9; Class: 1-6) and **Diabetes** (150 x 8; Class: 1-2), are considered for the study [37].

2.2. Tools Used

Dev C++ is used for the development and implementation of the NNC, BP, SA and BCGA algorithms. *MATLAB* 9 is used for plotting graphs. Intel Core 2 Duo 2 GHZ with 1 GB RAM processor is used for simulation of all the programs.

2.3. Techniques Applied

This section briefly describes the complete methodology adopted in our work.

2.3.1 Feed-Forward NNC Construction

Feed-Forward NNC is developed for each dataset using the structure feature available in C language. The structure of each net is based on the notation $(n - \lceil n/2 \rceil - 1 :: \text{Input-Hidden-Output})$ with 1 hidden layer where n is the number of neurons/nodes in the input layer. Table 1 describes the structure of each network developed.

Table 1. Description of NNCs, Developed

Network Description	Datasets			
	TAE	Iris	Diabetes	Glass
No. of layers	3	3	3	3
No. of hidden layers	1	1	1	1
No. of neurons in input layer	5	4	8	9
No. of neurons in output layer	1	1	1	1
No. of neurons in hidden layer	3	2	4	5

Log Sigmoid function mentioned in (1) is used as the transfer function to keep the output of each hidden/output neuron between 0 to 1.

$$f(x) = (1 + e^{-x})^{-1} \quad (1)$$

The cost function here used is the Mean Square Error (MSE) calculated by the difference between the calculated output (CO) and the target output (TO). MSE cost function is given in (2).

$$MSE = \frac{1}{N} \sum_{i=1}^N (TO - CO)^2 \quad (2)$$

Where 'N' denotes the number of total instances

As we are also investigating the run time as another objective function we have used TIME (see (3)) as the cost function.

$$TIME = \frac{1}{N} \sum_{i=1}^N t_i \quad (3)$$

2.3.2 Normalization

The datasets constituting features and class labels are pre-processed to have values from 0 to 1 before feeding into NNC to facilitate simpler calculation and use of log sigmoid function as transfer function.

2.3.3 BP Algorithm

BP, a well-directed gradient based search, works in the following phases: (1) passing of input signals in forward direction and calculation of output at output layer, (2) Calculation of error using cost function, and (3) back propagation of error using *generalized Δ rule* using (4) and (5) [16]. Here the basic idea is to reduce the MSE through iterations consequently increasing the prediction accuracy using the principle of gradient descent.

$$\Delta w_{i,j}^t = -\lambda \cdot \frac{\partial MSE}{\partial w_{i,j}^t} + \alpha \cdot w_{i,j}^t \quad (4)$$

$$w_{i,j}^{t+1} = \Delta w_{i,j}^t + w_{i,j}^t \quad (5)$$

Where 'w' denotes the weights between the connectors 'i' and 'j' and 't' is the state of iteration. To optimize the learning process rigorous parametric study (PS) [38] is included. In PS, the Learning Rate (λ) is optimized by checking the minimum MSE associated λ . Table 2 lists the parameters involved in training NNC with BP algorithm.

Table 2. Parameter List for BP

Parameters	Corresponding Values
Learning Rate (λ)	0.1 to 1.0
Momentum Constant(α)	0.9
Epoch Size	100

2.3.4. SA Algorithm

SA is an adaption of the physical process where a solid is gradually cooled until its structure attains the “frozen” state at the minimum energy level [39]. Likewise, the weight set in the neural network has to go through a number of configurations during the annealing process until it obtains the global minima.

The process starts at a high temperature; with gradual decrement of temperature and the weight configuration changes attaining a “thermal equilibrium” state at each temperature [17].

At each temperature, we randomize the weights. We accept the new set of weights as the newly optimized set of weights if the error with this weight set is lower than the error with the previous set of weights or with a probability that the current set of weights will lead to the global minima. Here, the PS includes the weight range to be optimized. Table 3 shows the list of parameters involved.

Table 3. Parameter List for SA

Parameters	Corresponding Values
Initial Temperature	10° C
Final Temperature	1° C
Weight Range	(-1,+1), (-2,+2), (-4,+4)
Equillibrium State	No. of changes in the weight set is more than 10 or No. of iterations is more than 15000

2.3.5. Binary Coded GA

GA, based on natural evolutionary process, is a global optimization algorithm. It constitutes of continual processes, such as (1) initialization of population, (2) evaluation of fitness of each individual in population (3) selection of best-fit individuals for cross-over process (4) cross-over and mutation of selected individuals to breed off-springs, and (5) evaluation of off-springs and their replacement with least-fit old individuals [22].

The type of GA used here is BCGA where each individual solution is encoded with binary digits. As BCGA is applied to a neural net, here the weight links connected between nodes are coded into strings of binary. Figure 1 shows the same.

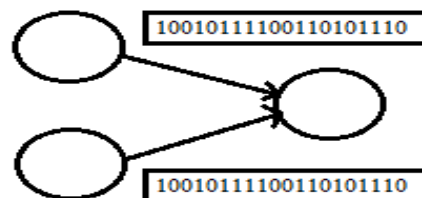


Figure 1. Weights of NN with Binary Encoding

Cross over operation in BCGA is done in the following way: (1) Select two individuals i.e. two weight sets $[W]_1$ and $[W]_2$ out of all individuals, (2) Mate each weight links of $[W]_1$ with the corresponding weight link of $[W]_2$ (i.e. $[W]_{1ij}$ with $[W]_{2ij}$ where W_{ij} is the weight link between i and j neurons) to produce two new weight sets. Figure 2 demonstrates the above operation.

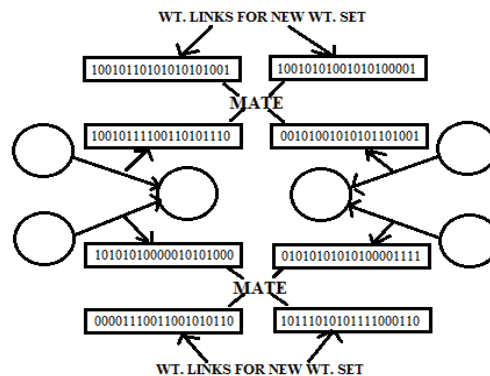


Figure 2. Cross Over Operation of two Weight Sets

In PS, the parameters are: weight range, type of cross over, initial population size. Epoch size is 500 as convergence rate of BCGA is slow. Table 4 shows the parameter list

Table 4. Parameter List for BCGA

Parameters	Corresponding Values
Epoch Size	500
Cross Over Type	Single-point, Two-point
Weight Range	(-1,+1), (-2,+2), (-4,+4)
Initial Population Size	100 to 1000
No. of Bits for a Wt. Link	20 for wt. range(-1,+1);21 for wt. range(-2,+2); 22 for wt. range(-4,+4)
Mutation	4 bits of each wt. link of offspring are randomly changed using XOR bitwise operator after cross over operation

III. RESULTS AND DISCUSSION

The result section constitutes two sub-sections. First describes the results of PS, while in the second, comparison is made.

3.1. Parametric Study Results

3.1.1. BP: Tuning Learning Rate

All possible λ are tested while training the NNC. The λ with minimum MSE during testing is selected for comparison with SA and BCGA. Figure 3 depicts the λ optimization for all the datasets. From Figure 3, it is clear that for Iris data, the minimum MSE is 0.0062 at $\lambda = 0.2$. For Glass data, minimum MSE is 0.0253 at $\lambda = 0.3$; while for the Diabetes data, minimum MSE is 0.0980 at $\lambda = 0.7$. Finally, for TAE, minimum MSE is 0.0558 at $\lambda = 0.2$.

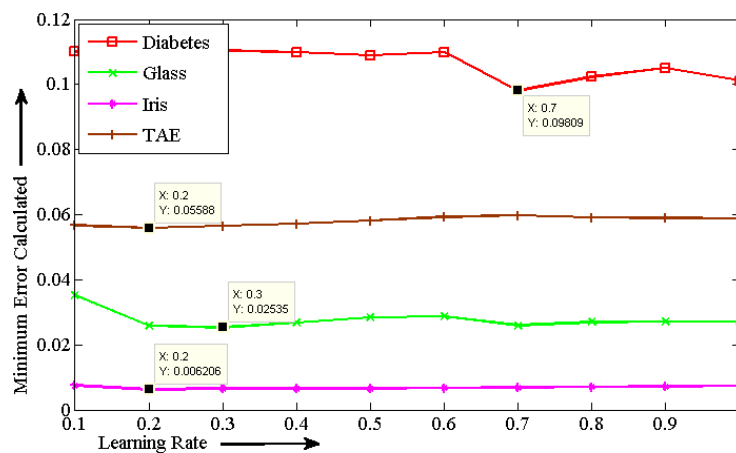


Figure 3. Learning Rate Optimization for all datasets

3.1.1. SA: Optimization of Weight Range

As mentioned earlier, different weight ranges i.e. the search spaces are tried during the training of NNC. The search space which results the minimum MSE during testing of the NNC is selected for comparison with BP. Table 5 suggests, for Iris data the minimum MSE is 0.0043; while for TAE it is 0.0549. For Glass and Diabetes, the minimum MSE values are 0.0268 and 0.0940 respectively. All the minimum MSEs fall in the weight range (-1, +1).

Table 5. SA Parametric Study Results

Weight Optimization	Minimum MSE (testing) of datasets corresponding to weight range			
	Iris	TAE	Glass	Diabetes
(-1,+1)	0.0043	0.0549	0.0268	0.0940

Weight Optimization	Minimum MSE (testing) of datasets corresponding to weight range			
	Iris	TAE	Glass	Diabetes
(-2,+2)	0.0044	0.0574	0.0322	0.1072
(-4,+4)	0.0046	0.0581	0.0385	0.1056

3.1.2. BCGA

Different parameters mentioned in Table 4 are optimized through PS. Figure 4 depicts the same.

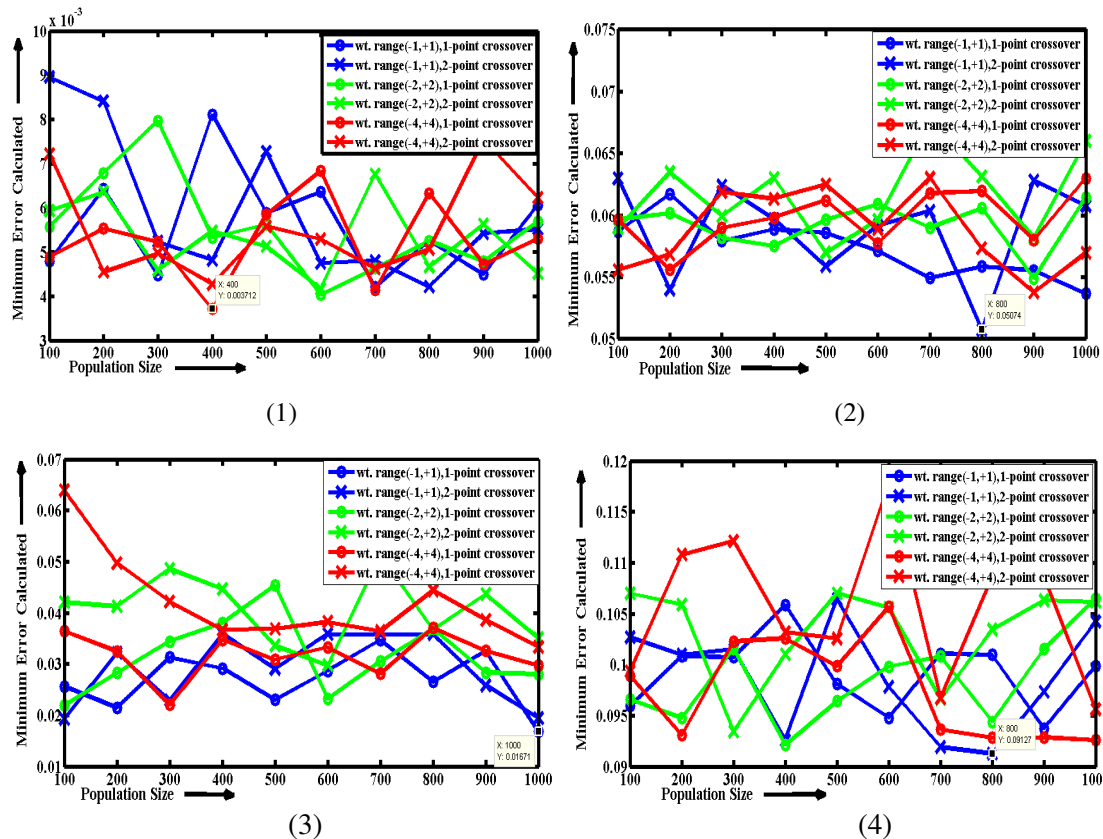


Figure 4. PS Results for (1) IRIS, (2) TAE, (3) GLASS and (4) DIABETES data

Results show, the minimum MSE 0.0037 for IRIS occurs with weight range (-4, +4), single-point crossover, and population size 400. For TAE, the minimum MSE 0.0507 occurs with weight range (-1, +1), two-point crossover, and population size 800. With weight range (-1, +1), single-point crossover, and population size 1000 minimum MSE 0.0167 is attained for Glass data. For Diabetes, the minimum MSE 0.0912 is obtained with weight range (-1, +1), two-point crossover and 800 population size.

3.2. Comparative Study of BP, SA and BCGA

This section includes the comparative study results in two sections based upon accuracy and convergence rate of NNC training.

3.2.1. Accuracy

Here accuracy is considered as the minimum MSE during testing. Table 6 & 7 demonstrate the advantage of BCGA over SA and BP when NNC deals with accuracy.

Table 6. Performance of BP, SA and BCGA during Training

Training Analysis	Minimum MSE (training) for all the datasets			
	Iris	TAE	Glass	Diabetes
BP	0.0071	0.0634	0.0133	0.0748

Training Analysis	Minimum MSE (training) for all the datasets			
	Iris	TAE	Glass	Diabetes
SA	0.0048	0.0541	0.0041	0.0672
BCGA	0.0066	0.0597	0.0120	0.0860

Table 6 suggests that SA acquires the pole position during training of NNC for all the datasets. BCGA acquires the 2nd position for all datasets except Diabetes. BP secures 2nd position for Diabetes; while 3rd for the rest.

Table 7. Performance of BP, SA and BCGA during Testing

Testing Analysis	Minimum MSE (testing) for all the datasets			
	Iris	TAE	Glass	Diabetes
BP	0.0062	0.0634	0.0253	0.0980
SA	0.0043	0.0549	0.0268	0.0940
BCGA	0.0037	0.0507	0.0167	0.0912

Results shown in Table 7 suggest BCGA outperforms SA and BP during testing for all the data. SA secures 3rd position for Glass and 2nd for the rest. BP secures 2nd position only for Glass data and 3rd for the rest. Here we try to emphasize, BP's tendency of getting stuck at local minima doesn't produce considerable accuracy. BCGA is able to find the best solutions from the available solutions for all datasets.

3.2.1. Convergence Rate

Table 8 shows the time consumed by BP, SA, and BCGA during training for the optimized convergence for each dataset.

As the system used for simulation is Intel Core 2 Duo 2 GHZ with 1 GB RAM the time consumed for convergence can be further optimized with a system with better processing capacity. Table 8 suggests, Convergence for BCGA is much slower than BP and SA. BP has got the highest convergence rate.

Table 8. Convergence Rate Comparison of BP, SA and BCGA

Convergence Rate Analysis	TIME taken for Convergence (in seconds)			
	Iris	TAE	Glass	Diabetes
BP	10	14	28	25
SA	190	90	52	124
BCGA	293	1500	4560	2554

IV. CONCLUSION AND FUTURE WORK

In this work we tried to solve a bi-objective optimization problem in terms of accuracy and speed. It concludes that SA takes more time (Iris: 190; TAE: 90; Glass: 52; Diabetes: 124) to converge when compared to BP (Iris: 10; TAE: 14; Glass: 28; Diabetes: 25), but much less time compared to BCGA (Iris: 293; TAE: 1500; Glass: 4560; Diabetes: 2554). On the other hand, BCGA provides better accuracy during testing (Iris: 0.0037; TAE: 0.0507; Glass: 0.0167; Diabetes: 0.0912) when compared to SA (Iris: 0.0043; TAE: 0.0549; Glass: 0.0268; Diabetes: 0.0940) and BP (Iris: 0.0062; TAE: 0.0634; Glass: 0.0253; Diabetes: 0.0980). Hence, we suggest that SA could be one reasonably useful technique for NNC optimization. However, it requires more study with different real-life datasets, which is the future work of the authors.

REFERENCES

- [1] G.K. Jha, "Artificial Neural Network", IARI, 2006
- [2] A. Abraham. Chapter 129: "Artificial Neural Network" in Handbook of Measuring System Design, by Sydenham P. H. and Thorn R. (eds.) John Wiley & Sons, 2005 [http://www.softcomputing.net/ann_chapter.pdf] last accessed on 04/05/2012]
- [3] V.S. Dave and K. Dutta, "Application of Feed-Forward Neural Network in Estimation of Software Effort". IJCA Proceedings on International Symposium on Devices MEMS, Intelligent Systems &

- Communication (ISDMISC) (5):5-9, 2011. Published by Foundation of Computer Science, New York, USA.
- [4] K.Y. Cai, L. Cai, W.D. Wang, Z.Y. Yu, D. Zhang., "On the neural network approach in software reliability modeling", *Journal of Systems and Software*, v.58 n.1, p.47-62, August 2001
- [5] J.G. Kuschewski, S. Hui, S.H. Zak, "Application of feedforward neural networks to dynamical system identification and control", *IEEE Transactions on Control Systems Technology*, vol.1, no.1, pp.37-49, Mar 1993, doi: 10.1109/87.221350
- [6] P. Melin, J. Urias, D. Solano, M. Soto, M. Lopez, O. Castillo, "Voice Recognition with Neural Networks, Type-2 Fuzzy Logic and Genetic Algorithms", *Engineering Letters*, Vol 13, Issue 2, EL_13_2_9 (Online; last accessed on 04/05/2012)
- [7] A. Skabar, I. Cloete, "Neural Networks, Financial Trading and the Efficient Market Hypothesis", ACSC, 2002.
- [8] J. Lin, W. Guo, G. Chen, H. Gao, X. Fang, "A PSO-BPNN-BASED model for energy saving in Wireless Sensor Networks", *International Conference on Machine Learning and Cybernetics*, Baoding, 12-15 July, 2009.
- [9] J. Yu, X. Hong, X. Liu, W. Teng, "Application of BPNN in analyzing LHM elemental basic structure", *Applied Mechanics and Materials* vols. 16-19, pp 460-470, 2009.
- [10] H. Wang, X. Kuang, J. Liu, "Application of back propagation neural network in paleoclimate", *International Conference on Information Science and Technology*, 2011.
- [11] I. Maqsood, M.R. Khan, A. Abraham, "Neural network ensemble method for weather forecasting", *Neural Computing and Application*, Vol 13, pp 112-122, 2004.
- [12] H. Dai, C. Macbeth, "Application of back-propagation neural networks to identification of seismic arrival types", *Physics of the Earth and Planetary Interiors*, vol. 101, issues 3-4, pp. 177-188, 1997.
- [13] B.S. Nayak., S.S. Behera, A. Shah, "Study and Realization of Defected Ground Structures in the Perspective of Microstrip Filters and Optimization through Neural Network". *International Journal of Advances In Engineering & Technology*, vol. 2, Issue 1, pp. 323-330, 2012.
- [14] S. Chattopadhyay, P. Kaur, F.A. Rabhi, U.R. Acharya, "Neural Network Approaches to Grade Adult Depression", *Journal of Medical Systems*, 2011 (in press) DOI: 10.1007/s10916-011-9759-1
- [15] S. Chattopadhyay, P. Kaur, F.A. Rabhi, U.R. Acharya, "An Automated System to Diagnose the Severity of Adult Depression", in the *Proceedings of EAIT-2012* (Eds. Jana D., Pal P.), pp.121-124.
- [16] I. Jung, W. Wang, "Pattern Classification of Back-Propagation Algorithm Using Exclusive Connecting Network", *World Academy of Science, Engineering and Technology*, 36, 2007.
- [17] D.M. Bailer-Jones, C.A.L. Bailer-Jones, "Modelling Data: Analogies in neural networks, simulated annealing and genetic algorithms" in *Model-Based Reasoning: Science, Technology, Values*, L. Magnani and N. Nersessian (eds.), New York: Kluwer Academic/Plenum Publishers, 2002.
- [18] K. Kaczmarski, D. Antos, "Use of Simulated Annealing for optimization of chromatographic separations", *Acta Chromatographica*, No. 17, pp. 20-45, 2006.
- [19] F.E. Goldman, L.W. Mays, Chapter 5: "Water distribution system operation: application of simulated annealing" In: Mays LW (ed) *Water Resources Systems Management Tools*. McGraw-Hill, New York, 2005.
- [20] F. Kolahan, M.H. Abolbashari, S. Mohitzadeh, "Simulated Annealing Application for Structural Optimization", *World Academy of Science, Engineering and Technology*. 2007, pp. 35.
- [21] B.S. Nayak, S.S. Behera, R.K. Mishra, "High Performance LPF Design using Neural Network and Simulated Annealing Techniques". *PIERS Proceedings*, 968-972, March 27-30, Kuala Lumpur, Malaysia 2012
- [22] D.E. Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning", Reading, MA: Addison-Wesley, 1989.
- [23] D.E. Goldberg, B. Korb, K. Deb, "Messy Genetic Algorithms: Motivation, Analysis, and First Results", *Complex Systems* vol. 3, no. 5, pp. 493-530, 1989.
- [24] A. E. Eiben, J. K. van der Hauw, and J. I. van Hemert, "Graph coloring with adaptive evolutionary algorithms," *J. Heuristics*, vol. 4, no. 1, pp.25-46, 1998.
- [25] S.G. Akl, "The Design and Analysis of Parallel Algorithms", Englewood Cliffs, NJ: Prentice-Hall, 1991
- [26] C.F. Huang, "Independent sampling genetic algorithms", *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 367-374, San Francisco, CA, 2001.
- [27] S. Vukovic, L. Sopta, "Binary-coded and real-coded genetic algorithm in pipeline flow optimization", *Mathematical Communications* vol. 4, pp. 35-42, 1999
- [28] A. A. L. Neyestanak, F. H. Kashani, and K. Barkeshli, "E-Shaped Patch Antenna Design Based on Genetic Algorithm Using Decision Fuzzy Rules", *Iranian Journal of Electrical and Computer Engineering*, vol. 4, no.1, 2005

- [29] E.E. Altshuler, D.S. Linden, , "Wire-antenna designs using genetic algorithms," *Antennas and Propagation Magazine, IEEE* , vol.39, no.2, pp.33-43, 1997
- [30] S. Bandyopadhyay, C.A. Murthy, S.K. Pal, "Pattern Classification using Genetic Algorithms: Determination of H", *Pattern Recognition Letters*, vol. 19, issue 13, pp. 1171-1181, 1998.
- [31] S. Chattopadhyay, D.K. Pratihar, S.C. De Sarkar, "Statistical Modeling of Psychoses Data", *Computer Methods and Programs in Biomedicine*, vol. 100, issue 3, pp. 222-236.
- [32] S. Chattopadhyay, D.K. Pratihar, S.C. De Sarkar, "Fuzzy Logic-based Screening and Prediction of Adult Psychoses: A Novel Approach", *IEEE Transactions on Systems, man and Cybernetics – Part A: Systems and Humans*, vol. 39, no. 2, pp. 381-387.
- [33] S. Chattopadhyay, D.K. Pratihar, S.C. De Sarkar, "Developing Fuzzy Classifiers to Predict the Chance of Occurrence of Adult Psychoses", *Knowledge Based Systems*, vol. 21, issue 6, pp. 479-497.
- [34] J.N.D. Gupta, R.S.Sexton, "Comparing genetic algorithm fo neural network training", *Omega*, volume 27, issue 6, December 1999, Pages 679-684.
- [35] H.H.Örkcü, H.Bal, "Comparing performances of backpropagation and genetic algorithms in the data classification", *Expert Systems with Applications*, Volume 38, Issue 4, April 2011, Pages 3703–3709.
- [36] R.S.Sexton, R.E.Dorsey, J.D.Johnson, "Beyond Backpropagation: Using Simulated Annealing for Training Neural Networks", *Journal of Organizational and End User Computing*, Volume 11, Issue 3, 1999.
- [37] <http://www.ics.uci.edu> [last accessed on 01/05/2012]
- [38] C. K. Lee, C.H. Chung, "The effect of learning rate to an unsupervised neural network for inspection process", *Transactions on Information and Communication Technologies*, WIT Press, vol. 19, 1997
- [39] D. Bertsimas and J. Tsitsiklis, "Simulated Annealing", *Statistical Science*, vol. 8, No. 1, pp. 10–15, 1993.

Authors:

Subhendu Sekhar Behera is currently pursuing his Bachelor of Technology in the Dept. of Computer Science & Engineering at National Institute of Science and Technology, Berhampur, Orissa. His research interests lie in the field of Soft Computing and Algorithm Design.



Sangeeta Bhanja Chaudhuri earned her B.Sc (Bio Science) from Gurudas College, Calcutta University, Kolkata India, then M.C.A degree in Computer Applications from The Oxford College of Science, Bangalore University and an M.Phil degree in Computer Science from Vinayaka Missions University. Data mining and Soft Computing are her key research fields. She has published about 3 research papers in International Journals and 3 in Conference proceedings. Currently she is an assistant professor of National Institute of Science & Technology, Berhampur India in the School of Computer Science & Engineering.



Subhagata Chattopadhyay earned his PhD degree in Information Technology from Indian Institute of Technology, Kharagpur India. He pursued postdoctoral fellowship at UNSW, Sydney Australia. Currently he is the Principal of Bankura Unnayani Institute of Engineering, India and a full professor in the Dept. of CSE in the same institute. He has published over 90 research papers in several refereed international journals, books and conferences.

