A Novel Version of the Bacterial Memetic Algorithm with Modified Operator Execution Order

László Gál [1, 2], László T. Kóczy [1, 3] and Rita Lovassy [4]

[1] Inst. of Informatics, Electrical and Mechanical Eng., Széchenyi István University, Győr, Hungary

[2] Dept. of Technology, Informatics and Economy, University of West Hungary, Szombathely, Hungary, gallaci@ttmk.nyme.hu

[3] Dept. of Telecommunication and Media Informatics, Budapest University of Technology and Economics, Hungary, koczy@sze.hu, koczy@tmit.bme.hu

[4] Inst. of Microelectronics and Technology, Kandó Kálmán Faculty of Electrical Engineering, Óbuda University Budapest, Hungary, lovassy.rita@kvk.uni-obuda.hu

***Abstract**: The Three Step Bacterial Memetic Algorithm is proposed. This new version of the Bacterial Memetic Algorithm with Modified Operator Execution Order (BMAM) is applied in a practical problem, namely is proposed as the Fuzzy Neural Networks (FNN) training algorithm. This paper strove after the improvement of the function approximation capability of the FNNs by applying a combination of evolutionary and gradient based (global and local search) algorithms. The method interleaves the bacterial mutation (optimizes the rules in one bacterium) and a local seach method applied for each clone with the Levenberg - Marquardt method to reach the local optimum. In our novel algorithm various kinds of fast algorithm with less complexity, like Quasi-Newton, Conjugate Gradient, Gradient Descent furthemore Gradient Descent with Adaptive Learning Rate and Momentum are nested in the bacterial mutation. The benefits arising from the combination between various fast local search methods and memetic algorithm have been investigated in this paper.*

***Keywords**: Fuzzy Neural Network; Bacterial Memetic Algorithm with Modified Operator Execution Order*

1 Introduction

Genetic Algorithms (GA) represent the most widely used technique in the class of evolutionary computing. The original GA was developed by Holland [6] and was based on the process of evolution of biological organisms. These optimization techniques simultaneously examine and manipulate a set of possible solutions. In genetic algorithms the basic entity is called “chromosome”. A chromosome with high fitness value will reproduce more intensively than one with low fitness. During natural selection the fittest biological entity survives. However, GAs have the ability to converge to the global optimum but it is generally acceptable that convergence speed of GAs is quite slow, i.e. finding the global optimum with sufficient precision often takes a very long time. Nevertheless, genetic algorithms are generally able to find reasonable “good” solutions to many problems with high complexity.

The Pseudo-Bacterial Genetic Algorithm (PBGA) is a special kind of Genetic Algorithm [7]. Its core contains the bacterium which is able to carry a copy of a gene from a host cell and insert it into an infected cell. By this process, called bacterial mutation, the characteristics of a single bacterium can spread to the rest of the population, so this method mimics the process of microbial evolution.

Nawa and Furuhashi improved the PBGA [11] and completed the bacterial mutation with a gene transfer operation. The proposed new method was called Bacterial Evolutionary Algorithm (BEA).

Memetic Algorithms (MA) combine evolutionary and local search methods [10], they are also known as hybrid evolutionary algorithms.

The Bacterial Memetic Algorithm (BMA) is a recently developed technique [1]. The method interleaves the bacterial mutation (optimizes the rules in one bacterium) and the gene transfer operation (recombines the genetic information between the chromosomes) with the Levenberg - Marquardt method (LM) to reach the local optimum. The main steps of the algorithm are [3]: population initialization, bacterial mutation applied for each individual, Levenberg-Marquardt method applied for each individual and gene transfer operation applied infection times per generation. This procedure is repeated from the bacterial mutation step until a certain stopping criterion is satisfied. Memetic algorithm has been successfully applied in engineering fields ranging from microarray gene selection [15], aerodynamic design [13] to drug therapies design [12].

Hybrid evolutionary methods that combine genetic type algorithms with “classic” local search have been proposed to perform efficient global search. Bacterial Memetic Algorithm with Modified Operator Execution Order (BMAM) is an improved version of the BMA [2]. The algorithm consists of bacterial mutation including several LM cycles saving some potential clones that could be lost otherwise, followed by the LM method applied in this case for each individual, and finally the gene transfer operation made for a partial population. This particular merger of evolutionary and gradient based (global and local search) algorithms is used rather successful for global optimization approaches, in particular by optimizing parameter values, and to improve function approximation performance [5].

The outline of this paper is as follows. After the Introduction, Section 2 deals with the Bacterial Memetic Algorithm with Modified Operator Execution Order in general. In Section 3 we introduce a new version of the BMAM algorithm called the Three Step Bacterial Memetic Algorithm. In the bacterial mutation the LM algorithm is substituted with various kinds of local search algorithm, like Quasi-Newton, Conjugate Gradient, Gradient Descent, and Gradient Descent with Adaptive Learning Rate and Momentum. Section 4 presents the fuzzy J-K flip-flop neuron in general. The characteristic equations of the trigonometric type fuzzy J-K flip-flop neuron is given.

Chapter 5 shows how the bacterial memetic algorithm can be modified with various local search methods to achieve different results in the function approximation process, developing a new hybrid combination. In particular, the Three Step BMA algorithm is proposed for Fuzzy Neural Networks (FNN) training. The networks are built up from fuzzy flip-flop neurons [8]. Finally, the FNN function approximation performance comparison through simulation results is discussed, followed by brief Conclusions and References.

2 Bacterial Memetic Algorithm with Modified Operator Execution Order

The core of the Bacterial Memetic Algorithm with Modified Operator Execution Order contains the bacterium which is able to carry a copy of a gene from a host cell and insert it into an infected cell. It starts with the generation of the initial population when a random bacterial population is created, then the bacterial mutation is applied to each bacterium one by one. The algorithm consists of the following steps:

A. Bacterial mutation operation for each individual

- Each individual is selected one by one
- *M* clones are created from the selected individuals
- The same part or parts are selected randomly from the clones and mutated
- Some LM iterations are run after each bacterial mutation step
- The best clone is selected and transferred with its all parts to the other clones
- Choosing-mutation-LM-selection-transfer cycle is repeated until all the parts are mutated, improved and tested

- The best individual is remaining in the population all other clones are deleted
- This procedure is repeated until all the individuals are taking part in the modified bacterial mutation

B. Levenberg-Marquardt method for each individual

C. Gene transfer operation for a partial population.

The Levenberg-Marquardt algorithm in the optimization process often finds only the local minimum while the bacterial algorithm can avoid the local minima but finding a quasi-optimal result.

3 Three Step Bacterial Memetic Algorithm

In this section a novel version of the BMAM called Three Step BMA is presented. The LM method nested in the bacterial mutation is the kernel of the algorithm. In the next this gradient method is substituted with various kinds of fast local search methods with relatively high convergence speed. In this approach the three main steps of the algorithm are:

A. Bacterial mutation operation for each clone,

B. Local search method for each clone,

C. Levenberg-Marquardt method for each individual.

The bacterial memetic algorithm in combination with various kinds of gradient methods with different convergence speed has been proposed, to exploit their advantages as the less complexity and the ability to find the local minima. The next methods are proposed to be applied for each clone:

1. Quasi-Newton Algorithm
2. Conjugate Gradient Algorithm
3. Gradient Descent Algorithm with Adaptive Learning Rate and Momentum
4. Gradient Descent Algorithm

In the novel algorithm the gene transfer operation is completely excluded. Different applications require certain combinations in order to keep the advantages of the given procedure. In the next, the Three Step BMA algorithm is proposed for Fuzzy Neural Networks [4] training.

4 Fuzzy J-K Flip-Flop Based Neurons

The next state Q_{out} of a fuzzy J-K flip-flop is characterized as a function of both the present state Q and the two present inputs J and K. The so called fundamental equation of the J-K type fuzzy flip-flop [14] is

$$Q_{out} = (J \vee \neg K) \wedge (J \vee Q) \wedge (\neg K \vee \neg Q) \,, \tag{1}$$

where $\wedge, \vee, \neg$ denote fuzzy operations, in particular fuzzy conjunction, disjunction and negation, respectively (e.g. $\neg K = 1 - K$). In [8] the construction of a single input-single output unit from fuzzy J-K flip-flop is proposed where $\neg Q$ is fed back to K (K = 1 - Q) and (the old) Q is fixed. In the next, the t-norm is denoted by i (intersection), and the t-conorm by u (union). The characteristic equation of the fuzzy J-K neuron is

$$Q_{out} = (J \ u \ Q_{\text{fix}}) \ i \ (J \ u \ Q_{\text{fix}}) \ i \ \left(Q_{\text{fix}} \ u \ \left(1 - Q_{\text{fix}}\right)\right) \tag{2}$$

Obviously Q_{out} can be generated by a combinational circuit.

4.1 Trigonometric Operations Based Fuzzy J-K Flip-Flop Neuron

In our previous paper [5] we defined a pair of new fuzzy intersection and union, giving their expressions and enumerating their properties. The proposed connectives consist of simple combinations of trigonometric functions. The basic motivation for constructing new norms was to have fuzzy flip-flops with sigmoid transfer characteristics in some particular cases.

The trigonometric norms are

$$i_T\left(x, y\right) = \frac{2}{\pi} \cdot \arcsin\left(\sin\left(x\frac{\pi}{2}\right) \cdot \sin\left(y\frac{\pi}{2}\right)\right) \tag{3}$$

$$u_T\left(x, y\right) = \frac{2}{\pi} \cdot \arccos\left(\cos\left(x\frac{\pi}{2}\right) \cdot \cos\left(y\frac{\pi}{2}\right)\right) \tag{4}$$

For simplicity the subscript refers to the initial of the name of the norm: in case of trigonometric norms: $i_T\left(x, y\right) = x \ i_T \ y$ and $u_T\left(x, y\right) = x \ u_T \ y$.

The fundamental equation of the trigonometric type fuzzy J-K flip-flop neuron can be rewritten in the form

$$Q_{out} = (J \ u_T \ Q_{\text{fix}}) \ i_T \ (J \ u_T \ Q_{\text{fix}}) \ i_T \ \left(Q_{\text{fix}} \ u_T \ \left(1 - Q_{\text{fix}}\right)\right) \tag{5}$$

The neuro-fuzzy system proposed is based on two hidden layers constituted from fuzzy flip-flop neurons. The FNN is a supervised feedforward network, applied in order to approximate test function. The nonlinear characteristics exhibited by fuzzy neurons are represented by quasi sigmoid transfer functions. The proposed network activation function is the same at each hidden layer, from unit to unit [9].

5 Simulation Results

The FNN architecture is predefined, depending on the input function complexity. The networks approximate one and two dimensional trigonometric functions. The test functions are:

5.1 Two Sine Waves

A 1-4-3-1 fuzzy J-K flip-flop neuron based neural network is proposed to approximate a combination of two sine waves with different period lengths described by the equation

$$y_1 = \sin(c_1 \cdot x) \cdot \sin(c_2 \cdot x)/2 + 0.5 \tag{6}$$

The values of constants c_1 and c_2 were selected to produce a frequency proportion of the two components 1:0.35 to keep the wavelet in the unit interval. The input and output signals are distributed in the unit interval. The test function denoted by 1D is represented by 100 input/output data sets.

5.2 Two - Input Trigonometric Function

We used the following two dimensional test function

$$y_2 = \cos\left(\arctan^*\left(\frac{x_1}{x_2}\right)\right) \cdot e^{-\frac{r^2}{50}} \cdot \sin^3\left(\frac{r}{10}\right) \tag{7}$$

where $r = \sqrt{x_1^{\,2} + x_2^{\,2}}, \qquad x_1,\ x_2 \in [-20, 20]$

and arctan* is the four-quadrant inverse tangent.

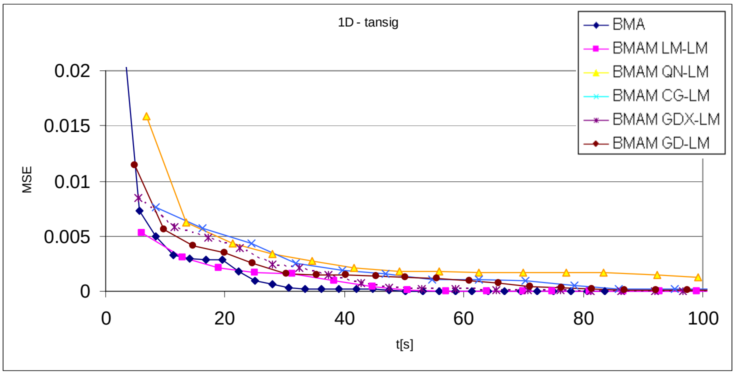
To approximate the two dimensional function denoted by 2D we proposed a 1-3-3-1 FNN size given by 1600 dates.

The FNNs are trained with the following methods: BMA, BMAM LM-LM, furthermore the Three Step BMA, including the combination of LM with Quasi-Newton (QN), Conjugate Gradient (CG), Gradient Descent with Adaptive Learning Rate and Momentum (GDX) and Gradient Descent (GD) algorithms.

In [5] by extensive simulations we found that the fuzzy J-K flip-flop neurons based on the trigonometric norms are suitable for constructing FNNs. In the next simulations we labeled them with TrigJKFF. The above mentioned types of FNN will be compared from the point of view of the respective fuzzy-neural network function approximation capability. The chosen target activation function was *tansig*, a Matlab built in sigmoid transfer function. In order to train the networks with the Three Step Bacterial Memetic Algorithm Matlab2007, one working thread under Windows 7 with AMD Athlon II X3 435 processor was used.

During simulations 30 generations of 5 individuals with 5 clones each were chosen to obtain the highest performance. 100 seconds and 200 seconds were fixed as secondary stopping criteria depending on the test function complexity. The LM, QN, CG, GDX and GD methods nested into the evolutionary algorithm were applied for 5 times for each clone. As the third step, the LM method was applied again for 7 times for each individual executing several cycles during the bacterial mutation after each mutation step. Figures 1 and 2 show the function approximation capability of the FNNs trained with the Three Step Bacterial Memetic Algorithm based on 5 various local search methods in comparison with the "original" Bacterial Memetic Algorithm [1]. During evaluation we compared the median of the training MSE values of the 8 runs average approximation goodness.

Comparing the function approximation capability of various FNNs we show that the networks using LM algorithms in the second and third steps outperform the other ones in every cases, followed by the combination of GDX with LM. The training methods based on CG, QN and GD local search algorithms provide FNNs with rather bad learning ability. It is very surprising that approximating the one dimensional test function (1D-TrigJKFF) applied for GDX-LM combination, the algorithm produced rather unexpected, very bad results. In order to find out whether these results are due to the inefficient implementation in Matlab, we intend to run our tests in another environment based on traditional programming languages such as C++. Comparing the simulation results we found that the proposed LM, QN, CG, GDX, GD algorithms produce remarkably small time delays between generations, however the GD and GDX methods work with far less time complexity than the LM method. That could be occurred by the Matlab simulation platform.



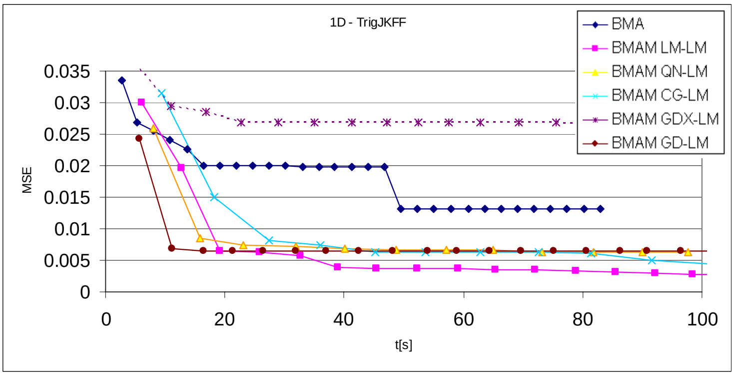
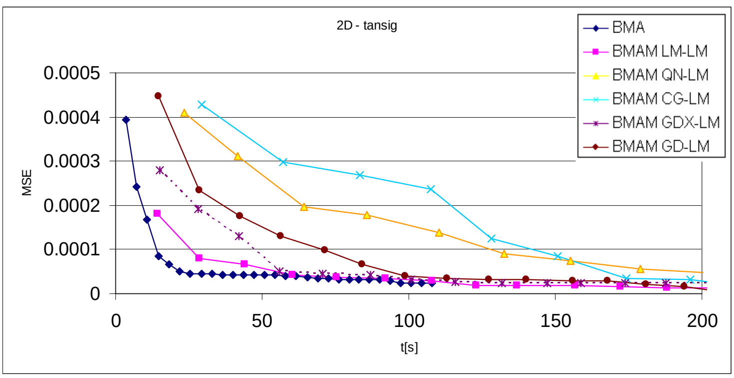


Figure 1

One dimensional test function. FNNs based on tansig and trigonometric norms trained with the Three Step BMA algorithm



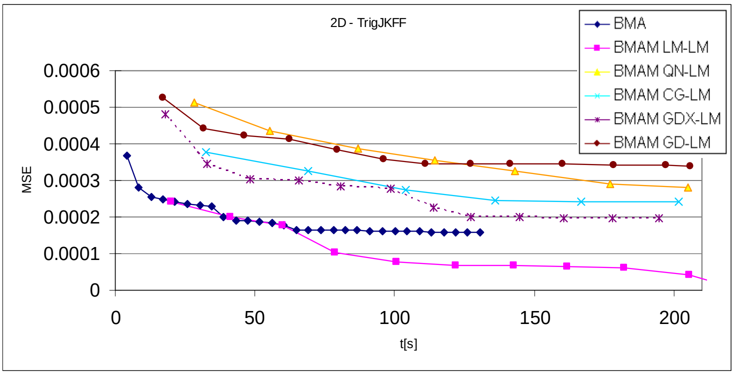


Figure 2
Two dimensional test function. FNNs based on tansig and trigonometric norms trained with the Three Step BMA algorithm

Conclusions

The benefits arising from the combination between various fast local search methods and memetic algorithm have been investigated in this paper. A hybrid merger of gradient based and evolutionary algorithms applied for FNNs training is proposed. In the original BMAM method with the LM method applied in two different steps achieves better function approximation results than the other combinations of local and global search methods. Future scope for the research lies in the use of the Three Step BMA algorithm in other simulation environments for various applications.

Acknowledgement

This paper was supported by the National Scientific Research Fund Grant OTKA K75711, further by Óbuda University Grants and Széchenyi István University Main Research Direction Grant.

References

[1] J. Botzheim, C. Cabrita, L. T. Kóczy, A. E. Ruano: Fuzzy Rule Extraction by Bacterial Memetic Algorithms, International Journal of Intelligent Systems, 24(3), 2009, pp. 312-339

[2] L. Gál, J. Botzheim, L. T. Kóczy: Improvements to the Bacterial Memetic Algorithm used for Fuzzy Rule Base Extraction, in Proc. of Computational Intelligence for Measurement Systems and Applications, CIMSA, Istanbul, Turkey, 2008, pp. 38-43

[3] L. Gál, L. T. Kóczy: Advanced Bacterial Memetic Algorithms, Acta Technica Jaurinensis Series Intelligentia Computatorica, 1(3), 2008, pp. 481-498

[4] L. Gál, L. T. Kóczy, R. Lovassy: Three Step Bacterial Memetic Algorithm, Proc. of 14th IEEE International Conference on Intelligent Engineering Systems, INES 2010, Las Palmas of Gran Canaria, Spain, pp. 31-36

[5] L. Gál, R. Lovassy, L. T. Kóczy, Function Approximation Performance of Fuzzy Neural Networks Based on Frequently Used Fuzzy Operations and a Pair of New Trigonometric Norms, IEEE World Congress on Computational Intelligence, WCCI 2010, Barcelona, Spain, pp. 1514-1521

[6] J. H. Holland: Adaptation in Natural and Artificial Systems, The University of Michigan Press, 1975

[7] J. H. Holland: Adaptation in Nature and Artificial Systems: An Introductory Analysis with Application to Biology, Control, and Artificial Intelligence, The MIT Press, Cambridge, MA 1992

[8] R. Lovassy, L. T. Kóczy, L. Gál: Multilayer Perceptron Implemented by Fuzzy Flip-Flops, in Proc. of IEEE World Congress on Computational Intelligence, WCCI 2008, Hong Kong, 2008, pp. 1683-1688

[9] R. Lovassy, L. T. Kóczy, L. Gál: Function Approximation Capability of a Novel Fuzzy Flip-Flop Based Neural Network, in Proc. of International Joint Conference on Neural Networks, IJCNN 2009, Atlanta, USA, 2009, pp. 1900-1907

[10] P. Moscato: On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms, Technical Report Caltech Concurrent Computational Program, Report 826, California Institute of Technology, Pasadena, USA, 1989

[11] N. E. Nawa, T. Furuhashi, T. Hashiyama, Y. Uchikawa: A Study on the Discovery of Relevant Fuzzy Rules Using Pseudo-Bacterial Genetic Algorithm, IEEE Transactions on Industrial Electronics, 46(6), 1999, pp. 1080-1089

[12] F. Neri, J. Toivanen, G. L. Cascella, Y. S. Ong: An Adaptive Multimeme Algorithm for Designing HIV Multidrug Therapies, IEEE/ACM Trans. Comput. Biol. Bioinformatics, 4, 2007, pp. 264-278

[13] Y. S. Ong, P. B. Nair, K. Y. Lum: Min-Max Surrogate-assisted Evolutionary Algorithm for Robust Aerodynamic Design, IEEE Trans., Evol., Comput., 10, 2006, pp. 392-404

[14] K. Ozawa, K. Hirota, L. T. Kóczy: Fuzzy Flip-Flop, In: M. J. Patyra, D. M. Mlynek, eds., Fuzzy Logic. Implementation and Applications, Wiley, Chichester, 1996, pp. 197-236

[15] Z. Zhu, Y. S. Ong, M. Dash: Wrapper-Filter Feature Selection Algorithm Using a Memetic Framework, IEEE Trans., Syst., Man, Cybern., B., 37, 2007, pp. 70-76