IJST, Transactions of Electrical Engineering, Vol. 37, No. E2, pp 193-198
Printed in The Islamic Republic of Iran, 2013

"Research Note"

COMPARING EVOLUTIONARY ALGORITHMS ON TUNING THE PARAMETERS OF FUZZY WAVELET NEURAL NETWORK*

P. ARAB KHEDRI, M. EFTEKHARI** AND R. MAAZALLAHI

Dept. of Computer Engineering, Shahid Bahonar University of Kerman, Kerman, I. R. of Iran
Email: m.eftekhari@mail.uk.ac.ir

Abstract– In recent years Fuzzy Wavelet Neural Networks (FWNNs) have been used in many areas. Function approximation is an important application of FWNNs. One of the main problems in effective usage of FWNN is tuning of its parameters. In this paper several different evolutionary algorithms including Genetic Algorithm (GA), Gravitational Search Algorithm (GSA), Evolutionary Strategy (ES), Fast Evolutionary Strategy (FES) and variants of Differential Evolutionary algorithms (DE) are used for adjusting these parameters on five test functions. The obtained results are compared based on some measures by using multiple non-parametric statistical tests. The comparison reveals the superiority of some variants of DE in terms of convergence behavior and the ability of function approximation.

Keywords– Fuzzy wavelet neural networks, function approximation, evolutionary algorithms, nonparametric statistical test

1. INTRODUCTION

In recent years Fuzzy Wavelet Neural Network (FWNN) has been used as one of the most effective methods of soft computing. In fact, the structure of FWNN is a Neural Network that has been combined by fuzzy rules for dealing with complex problems which have ill-defined conditions and uncertain factors. Also, wavelet functions have been utilized in the consequent parts of fuzzy rules.

FWNN has been used in many different areas such as prediction, reinforcement learning and pattern recognition [1-4]. One of the main important applications of FWNN is function approximation [1-4]. In order to improve the function approximation accuracy and general capability of the FWNN system, the parameters of the FWNN must be adjusted. Several studies have been performed in which different variants of EAs have been applied for parameter optimization of FWNNs [1].

In this paper several evolutionary algorithms are used for adjusting the parameters of FWNN on some test functions and the results are compared.

The paper is organized as follows: Section 2 introduces FWNN and the evolutionary algorithms used. In Section 3 nonparametric statistical tests are briefly described. Experimental results are presented in Section 4. Finally, the conclusions are given in Section 5.

2. BASIC CONCEPTS

a) Fuzzy Wavelet Neural Network (FWNN)

In the following, the basic concept of fuzzy wavelet neural network is briefly introduced [1]. The structure of fuzzy wavelet neural network could be described as a set of M fuzzy rules. Rule R_j is defined as follows:

*Received by the editors December 12, 2012; Accepted October 20, 2013.
**Corresponding author

$$R_j: IF\ x_1\ is\ A_{1j}\ and\ \dots and\ x_n\ is\ A_{nj},\ \ Then\ y\ is\ \theta_j \tag{1}$$

where x_i , i=1...n, are the inputs and y is the output of FWNN. A_{ij} are linguistic terms characterized by fuzzy membership functions. In this paper, the Gaussian membership function is used which is defined as:

$$\mu_{A_{ij}}(x_i) = exp\left[-\frac{1}{2}(\frac{x_i - c_{ij}}{\sigma_{ij}})^2\right] \tag{2}$$

where c_{ij} and σ_{ij} stand for the center and width parameters, respectively. The output of R_j is calculated as follows:

$$\theta_j = \sum_{i=1}^{n} w_j \psi_{ij}(x_i) \tag{3}$$

where w_j are the weighting coefficients and $\psi_{ij}(\mathrm{x_i})$ stand for the family of wavelets obtained from the single Mexican Hat function, $\psi\ (x) = (1 - x^2)exp(-\frac{x^2}{2})$. Therefore $\psi_{ij}(\mathrm{x_i})$ is calculated as follows:

$$\psi_{ij}(x_i) = \frac{1}{\sqrt{|d_{ij}|}}(1 - z_{ij}^2)\exp\left(-\frac{z_{ij}^2}{2}\right) \tag{4}$$

where

$$z_{ij} = \frac{x_i - t_{ij}}{d_{ij}} \tag{5}$$

here, d_{ij} ,t_{ij} stand for the dilation and translation parameters, respectively. The output of FWNN is obtained as follows:

$$f(x) = \frac{\sum_{j=1}^{M} \theta_j \left[\prod_{i=1}^{n} \exp(-\frac{1}{2}(\frac{x_i - c_{ij}}{\sigma_{ij}})^2)\right]}{\sum_{j=1}^{M} \left[\prod_{i=1}^{n} \exp(-\frac{1}{2}(\frac{x_i - c_{ij}}{\sigma_{ij}})^2)\right]} \tag{6}$$

and

$$\theta_j = \sum_{i=1}^{n}\ w_j\psi_{ij}(x_i)\ =\ \sum_{i=1}^{n} w_j \frac{1}{\sqrt{|d_{ij}|}}\left(1 - (\frac{x_i - t_{ij}}{d_{ij}})^2\right) ex\,p\left(-\frac{1}{2}\left(\frac{x_i - t_{ij}}{d_{ij}}\right)^2\right) \tag{7}$$

So the parameters of FWNN that must be adjusted are $c_{ij}, \sigma_{ij}, t_{ij}, d_{ij} and\ w_j$. As we noted earlier, in this paper some evolutionary algorithms are utilized for the parameter tuning purpose. The structure of each chromosome, B, is as follows:

$$B = [c_{ij}\ \sigma_{ij}\ t_{ij}\ d_{ij}\ w_j]\ \ for\ i = 1, \dots, n\ and\ j = 1, \dots, M \tag{8}$$

where

$$c_{ij} =\ [c_{11} \dots c_{1M} \dots c_{n1} \dots c_{nM}]\ \ , \sigma_{ij} =\ [\sigma_{11} \dots \sigma_{1M} \dots \sigma_{n1} \dots \sigma_{nM}]\ , t_{ij} = [t_{11} \dots t_{1M} \dots t_{n1} \dots t_{nM}]$$
$$,\ \ d_{ij} =\ [d_{11} \dots d_{1M} \dots d_{n1} \dots d_{nM}]\ \text{and}\ w_j =\ [w_1 \dots w_M] \tag{9}$$

For evaluating each chromosome, the Mean Squared Error (MSE) is used as fitness function.

b) Evolutionary Algorithms (EAs)

A genetic algorithm (GA) is a search heuristic that mimics the process of natural evolution [1]. In GSA [5] chromosomes are considered as masses which attract each other by gravitational force. The chromosome with a heavier mass is a better solution and attracts others more. Therefore, the population moves toward the heaviest mass. Evolutionary strategy (ES) [6] is an optimization technique based on ideas of adaptation and evolution. It belongs to the general class of evolutionary computation or artificial evolution methodologies. In ES each chromosome consists of two parts: object-parameters and strategy-parameters. Fast Evolutionary Strategy (FES) [7] is a version of ES which uses Cauchy distribution function in mutation step instead of Gaussian distribution. The Differential Evolution algorithm (DE) [8, 9] is one of the most powerful stochastic real-parameter approaches to global numerical optimization which is reliable and fast. DE has different variants regarding the selection and recombination methods utilized in it. In this paper, nine different variants of DE are used, DE1 to DE9.

3. NONPARAMETRIC STATISTICAL TEST

In recent years the use of statistical tests has been increased to find out whether an algorithm has significant improvement over the others [10, 11]. The Friedman test is one of the N×N nonparametric statistical tests which has been greatly used for ranking the algorithms [10, 11]. Post-hoc procedures are then applied to determine whether the algorithms have significant difference with each other and whether the obtained ranking is reliable or not [11].

4. EXPERIMENTAL RESULTS

In this paper we have used five test functions to compare the performance of FWNN whose parameters are tuned by different evolutionary algorithms. Table 1 presents these test functions.

The parameters of different evolutionary algorithms (GA, GSA, ES, FES and nine variants of DE) are set as follows: 5000 generations, population size of 200, twenty independent runs for n=1 (number of inputs) and M=4 (number of fuzzy rules). The convergence diagrams of different EAs for the first benchmark function are depicted in Fig.1 based on the average of best so far solutions in each generation. The results show that variants of DE have better convergence rate and accuracy than others. This is because DE operates greedy and replaces best offspring with their parents only if they are better. The second reason is that DE has a lesser number of parameters to tune than other algorithms.

Table 1. Test functions

Name	Test function
Equal maxima	$F_1(x) = sin^6(5\pi x)$ $\quad 0 \le x \le 1$
Uneven decreasing maxima	$F_2(x) = \exp\left(-2\log(2).\left(\frac{x-0.08}{0.854}\right)^2\right).sin^6\left(5\pi\left(x^{\frac{3}{4}}-0.05\right)\right)$ $\quad 0 \le x \le 1$
Schwefel	$F_3(x) = 418.9829n + [-x\ \sin\left(\sqrt{\lvert x \rvert}\right)]$ $\quad -500 \le x \le 500$
Piecewise function 1	$F_4(x) = \begin{cases} -2.186x - 12.864, & -10 \le x < -2 \\ 4.246x, & -2 \le x < 0 \\ 10\exp(-.05x - 0.5)\, sin[(0.03x + 0.7)x], & 0 \le x \le 10 \end{cases}$
Piecewise function 2	$F_5(x) = \begin{cases} 3, & -8 \le x \le -4 \\ -4x - 13, & -4 < x \le -3 \\ x^2 + 6x + 8, & -3 < x \le 0 \end{cases}$

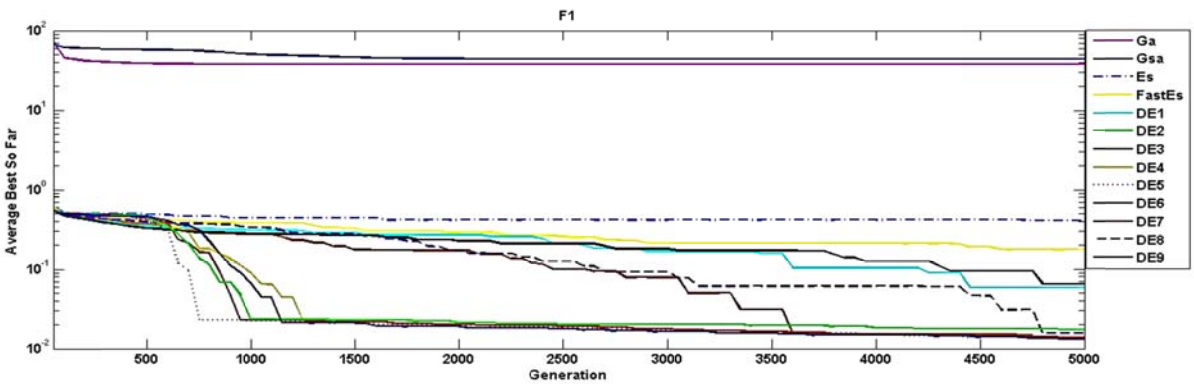

Fig. 1. Comparing the convergence of evolutionary algorithms used in tuning the FWNN for approximating F_1

In the following subsection, the results of comparing the evolutionary algorithms based on different criteria are presented. For each comparison, the multiple Friedman statistical tests (N×N) are used to rank the evolutionary algorithms. To make sure that the ranks obtained by Friedman test implicate a significant difference between each pair of algorithms, post-hoc procedures which determine an adjusted p-value (APV) for each hypothesis are utilized. The APV obtained by a post-hoc procedure is used for rejecting or accepting the hypothesis with a significant degree denoted by α. When the APV for a hypothesis is less than α, it is rejected with confidence level of (1-α), meaning the two algorithms specified by that hypothesis have significant difference. The significant degree of α=0.1 and two post-hoc procedures (Shaffer and Holm) [10-11] are applied in our multiple comparison tests.

a) Comparing evolutionary algorithms

Mean-Best-Fitness is the average of best fitness values obtained from different runs that indicates the performance of algorithms in terms of the accuracy of obtained results. Another important factor in comparing the algorithms is how fast they are. Therefore, the average number of fitness evaluations is calculated in 20 runs for each algorithm. For each test function, 20 independent runs are carried out and the standard deviation of fitness values for the best chromosomes (called STDEV-Best-Fitness) in different runs is calculated. The robustness of algorithms can be specified using this measure, which helps us to find out whether an algorithm yields almost the same results in multiple runs.

Figure 2 compares the EAs based on number of Mean-Best-Fitness, Mean-Fitness-Evaluations and STDEV-Best-Fitness. Three bars (from left to right) for each algorithm are the average rankings obtained by Friedman test considering Mean-Best-Fitness, Mean-Fitness-Evaluation and STDEV-Best-Fitness measures respectively.

As mentioned earlier, the post-hoc procedures are applied for a better understanding of the rankings obtained by Friedman test. The results are presented in Table 2 (Only some of the entire 78 hypotheses are shown due to the lack of space). We consider the results obtained by Shaffer and Holm post-hoc procedures. From Table 2, it is implied that DE6 has the best performance in terms of Mean-Best-Fitness among all the variants of DE, because it is rejected in four hypotheses relating to DE variants and has the lowest average ranking, as shown in the Fig. 2. Also, DE5 and DE7 have the worst performance as they are rejected in many hypotheses and have the highest average ranking regarding the Mean-Best-Fitness measure.

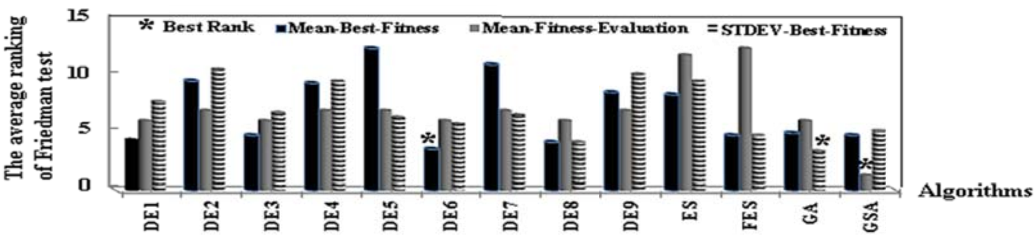

Fig. 2. The average rankings obtained by Friedman test considering Mean-Best-Fitness, Mean-Fitness- Evaluation and STDEV-Best-Fitness measures. (The lower rank means better performance of algorithm.)

Table 2. Adjusted p-values obtained by Post-hoc procedures for multiple comparisons among all algorithms based on different measures' rankings

	Mean-best-fitness measure			Mean-fitness-evaluation measure			STDEV-best-fitness measure		
i	Hypothesis	Holm	Shaffer	Hypothesis	Holm	Shaffer	Hypothesis	Holm	Shaffer
1	DE6 vs .DE5	0.0275	0.0275	FastEs vs .GSA	0.0005	0.0005	DE2 vs .GA	0.2702	0.2702
2	DE8 vs .DE5	0.0298	0.0292	ES vs .GSA	0.0005	0.0005	DE9 vs .GA	0.2791	0.2758
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
78	DE3 vs .FastEs	1	1	DE8 vs .GA	1	1	DE4 vs .Es	1	1

In Fig 2, GSA has the best ranking in terms of Mean-Fitness-Evaluations. Also, by looking at Table 2 which shows the result of post-hoc procedures, it can be seen that GSA rejects all the hypotheses. Thus, the GSA is surely the fastest algorithm. FES and ES have the worst ranking and are also rejected in all hypotheses. Therefore, they are the slowest algorithms. With the same explanation, it is concluded that the variants of DE and GA are faster than FES and ES but are slower than GSA. Based on the STDEV-Best-Fitness measure in Fig. 2, the GA has the best average ranking but it is not able to reject any hypotheses as implied from Table 2. Therefore, it is concluded that there is no significant difference between algorithms considering this measure.

5. CONCLUSION

In this paper, the structure of the FWNN model was introduced for function approximation from input–output pairs. It integrates the advantages of fuzzy concepts, wavelet functions, and neural networks. The parameters of FWNN must be adjusted properly before it can be used for function approximation. For this task, some evolutionary algorithms (GA, GSA, ES, FES and nine variants of DE) were used and their performance based on Mean-Best-Fitness, Mean-Fitness-Evaluation and STDEV-Best-Fitness factors were compared using Friedman statistical test. We also plotted the convergence diagram of each algorithm. The results show that variants of DE are the winner considering convergence rate and Mean-Best-Fitness factor. GSA runs faster due to the least average number of Fitness evaluations. The statistical tests implied that there is no significant difference between algorithms considering STDEV-Best-Fitness measure. So DE is recommended for approximation in situations where the accuracy is the most important factor.

REFERENCES

1. Shian-Tang, T. (2010). Design of fuzzy wavelet neural networks using the GA approach for function approximation and system identification. *Fuzzy Sets and Systems*, Vol. 161, pp. 2585-2596.
2. Zekri, M., Sadri, S. & Sheikholeslam, F. (2007). Approximation of feedback linearization control input based on fuzzy wavelet networks. *Iranian Journal of Science & Technology, Transaction B, Engineering*, Vol. 31, No. 6, pp. 651-662.
3. Ebadat, A., Noroozi, N., Safavi, A. A. & Mousavi, S. H. (2011). New fuzzy wavelet network for modeling and control: The modeling approach. *Communications in Nonlinear Science and Numerical Simulation*, Vol. 16, pp. 3385-3396.
4. Uma Shankar, B., Meher, S. K. & Ghosh, A. (2011). Wavelet-fuzzy hybridization: Feature-extraction and land-cover classification of remote sensing images. *Applied Soft Computing*, Vol. 11, pp. 2999-3011.
5. Rashedi, E., Nezamabadi-pour, H. & Saryazdi, S. (2009). GSA: A gravitational search algorithm. *Information Sciences*, Vol. 179, pp. 2232–2248.
6. Navale, R. L. & Nelson, R. M. (2012). Use of genetic algorithms and evolutionary strategies to develop an adaptive fuzzy logic controller for a cooling coil Comparison of the AFLC with a standard PID controller. *Energy and Buildings*, Vol. 45, pp. 169-180.
7. Yao, X. & Liu, Y. (1997). Fast evolution strategies. *Lecture Notes in Computer Science*, Vol. 1213, pp. 151-161.
8. Warren, L. T. (2010). Two hybrid differential evolution algorithms for engineering design optimization. *Applied Soft Computing*, Vol. 10, pp.1188-1199.
9. Pan, Q. K., Suganthan, P. N., Wang, L., Gao, L. & Mallipeddi, R. (2011). A differential evolution algorithm with self-adapting strategy and control parameters. *Computers & Operations Research*, Vol. 38, pp. 394-408.

10. García, S., Fernández, A., Luengo, J. & Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, Vol. 180, pp. 2044-2064.
11. Derrac, J., García, S., Molina, D. & Herrera, F. (2011). A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation*, Vol. 1, pp. 3-18.