International Journal of Computer Information Systems and Industrial Management Applications.
ISSN 2150-7988 Volume 9 (2017) pp. 249-256

Shuffled Frog-Leaping Algorithm trained RBFNN Equalizer

Pradyumna Mohapatra[1], Padma C. Sahu[2], K. Parvathi[3] , Siba Prasada Panigrahi[4]

[1]OEC, Bhubaneswar,[2]KIT, Berhampur,KIIT University [3]VSSUT, Burla, Odisha, India, siba_panigrahy15@rediffmail.com

Abstract. Ability of Artificial Neural Networks (ANN) in mapping between the variables attracts its application in channel equalization. Single hidden layer of Radial Basis function Neural Networks (RBFNN) makes it most popular equalizers to mitigate the channel distortions. Most challenging problem associated with design of RBFNN Equalizer is the traditional hit and trial method. Ability of evolutionary algorithms in solving complex problems in finding global optimal solutions attracted this paper for training of RBFNN equalizer using a recently proposed population based optimization, Shuffled Frog-Leaping Algorithm (SFLA) and three of its modified forms. It is found from the simulation results that performances of different forms of SFLA for the training of RBFNN equalizers are superior as compared to existing equalizers.

Keywords: Radial Basis Function Neural Network, Channel Equalization, Shuffled Frog-Leaping Algorithm.

1. Introduction

The advent of cellular communications has changed the way people communicate. Use of mobile phones has changed from a luxury item to a necessity of day to day life. Today it is also expected that social requirements, rather than technology or location, it is the ready people those decide how fast they can reach. Present day voice communication fulfills their basic requirement to communicate, and mobile phone connections continuing to improve even better and became the luxury requirement of day to day life. People also developed an appetite for services at affordable cost; audio-visual connection can be a bright example. The ever growing internet services have become a common source of information in day to day life, and ease and versatile mobile in access to this data will be taken for granted.

In early 1980s, the first generation cellular and cordless phone systems were introduced where analog FM technology is implemented to carry voice services only. Then, the second generation digital cellular radio networks were introduced to improve the spectral efficiency and voice quality in early 1990s. Basically, the cellular networks on air can be divided into two major categories given as Time Division Multiple Access (TDMA) and Code Division Multiple Access (CDMA). The European standard Global System for Mobile Communication (GSM), which is the world leader of second generation communication systems, is designed based on TDMA concept operating at 900 MHz, 1800 MHz and 1900 MHz. In addition, the digital PCS IS-136 which is the extension of IS-54 in United States, and Personal Digital Cellular (PDC) in Japan are also based on TDMA. Since the introduction of digital cellular radios networks, the service providers were facing the exponential growth of subscriber numbers in wireless communication systems.

Predicted from the trend of growth, the evolution of second-generation cellular systems is necessary. Based on the 2-G background, the third generation wireless systems were introduced that allowing the mobile users to have larger bandwidth for new features such as web browsing, video, image and other multimedia services.

Third generation (3G) cellular networks became popular because of the cost effective optimization of network capacity and quality of service (QOS). This is achieved with improvements in transmission methods, careful network planning and operation and advances in receiver design techniques. Intend to provide quality of service in all types of requirements for future wireless communications, extensive researches are in progress for fifth and later generation of communication systems.

With the deployment of Fourth Generation (4G) that increases the data throughput by around 4 times as compared to 3G. In addition, 4G systems provides services for Wimax 2 and LTE Advance. This appears as a high speed communication system and likely to capture the market soon.

The researches to standardize the Fifth generation (5G) network that will achieve higher system capacity, spectral efficiency, energy efficiency and data rate are still undergoing [1, 2]. With the Internet of Things (IoT) being served by a 5G system, the bulk of transmissions will show similar characteristics as today. So, 5G will potentially require a transmission mode with very low air interface [2].

Present day research on filter design and channel equalization focuses around use of swarm and evolutionary algorithms. However, use of artificial neural network (ANN) is common and popular in a wide range of engineering problems like those of the problem equalization. A detailed review on channel equalization using Multi layer Perceptron (MLP), functional-link artificial NN (FLANN) and neuro-fuzzy systems is provided in [1, 2]. Recent literature on channel equalization shows a pointer for use of neural networks [3, 4]. As discussed above, ability to sove complex problems for global minima makes it a right choice for use of evolutionary algorithms in equalization. ANNs trained with evolutionary algorithms is also a recent trend for engineering applications. ANN trained with a hybrid algorithm has also bee used in [5]. But ANNs have associated with large complexity and also fail because of over-fitting and local optima. However, in [6] it was demonstrated that the RBFNN has identical structure to the optimal Bayesian equalizer, and can be used to implement it RBFNN also finds global minima [7]. In addition to lesser complexity, RBFNN performance is also better than ANN as proved in the literature. These merits of RBFNN in channel equalization became an active area of research [7-10].

Once it is decided to use RBFNN for channel equalization, the problem consists in setting RBFNN parameters (centers and widths) properly, which is the focus of this paper. In work of Chen et al. [6], the RBFNN parameters was determined through simple classical methods as the k-means clustering and the LMS algorithm. To avoid time consuming process of classical methods, Barreto et.al [11] used Genetic Algorithm (GA) and Feng [12] used Particle Swarm Optimization (PSO). These are used to decide these key and bias parameters. Minimization of the Mean Square Error (MSE) between the desired and actual outputs is actual criteria in the designs formulated in [11, 12]. Still, in PSO there is a limited to a finite search space and hence falls to the local minima [13].

The Shuffled Frog Leaping Algorithm (SFLA) has some advantages such as simple steps, a few parameters, fast speed and easy realization [14]. These merits of SFLA and its extensive applications [15, 16] promoted use of SFLA to train ANN and use in channel equalization [17]. To avoid the limitations of ANN, as discussed above, SFLA has been used to train RBFNN based equalizer. In addition, exploration of the advancements and improvements in SFLA, this paper introduced three of its modified forms of SFLA for training of RBFNN equalizer. Merits of this paper can be seen as use of RBFNN that is advantageous over MLP for the problem and use of improved forms of SFLA, which is also proved in the simulations.

2 SFLA and Modified Forms

2.1 SFLA

The SFLA is gaining importance in engineering optimization problems because of advantages like simplicity in its structure & realization and also because of necessity of lesser number of control parameters. Each frog is a solution to the problem in SFLA. A set of frogs with similar structure constitute the population. The population divided into subsets, sub-memeplexes. Motivations that led some of researchers to propose modified forms of SFLA are discussed in this section. The SFLA [16, 17] is a meta-heuristic optimization algorithm. The algorithm mimics to the food searching behavior of frog groups. They search a position where they will get maximum food. First, the frog group is divided and then undergoes memetic evolution. Frog group is basis for memetic vectors in the algorithm. They host the memes. These vectors are identical in structure but capacity for adaptation is different. These memes are improved because of communication among the frogs in the group.

Implementing SFLA into an optimization problem, we generate a population of frogs. The population then divided into a number of sets. These sets are termed as memeplexes. A particular memeplex is that which represent a particular type of meme. In the search-space, the memeplexes moves randomly and independently in all possible directions. Local exploration of search-space gives rise to memetic evolution by the frogs in each memeplex.

A pre-defined strategy is meant for evolution. In this phase memes are transferred inside the same individuals of the group. Adaptation capacity of one frog is its fitness, which is the feasible solution for the optimization problem. The number of evolutions are also pre-defined. The next process is shuffling In this stage exchange of information takes place among the memeplexes of the same group. A bias-free cultural improvement through a cultural evolution leads to the desired location.

A pre-defined stopping criteria meant for convergence stops the process of shuffling and local exploration. The terms used in SFLA are:

- Memetic vector: A N number of frogs in a group hosting the memesforms the memetic vector.
- Memeplex: A set of n identical frogs with a particular meme type is memeplex. The number of memeplexes are, $m = N / n$.
- Submemeplex: memeplex can be divided into a set of submemeplexes (with q n number of frogs with better fitness) to avoid failing to local minima.
- Population: The solution space is a collection of memeplexes and termed as population.
- Fitness: Adaptation capacity of a frog is its fitness denoted by f i for ith frog.
- Memetic evolution: movement of frogs in local exploration is memetic evollution.
- If, S is step size for frog leaping, $S_{\max}$ is maximum allowed distance in one jump and r is a random number between (0, 1), then the update equations used in SFLA can be: $S = r\left(X_b - X_W\right)$ and $X_W^{'} = X_W + S$
- X_W leaps toward X_b, during the evolution, as per above equations. If it gives rise to a better solution, then it replaces the X_W . Else otherwise, X_W replaces X_b . If the resulting leap is not towards a better one, then randomly generate the new position of the X_W

The algorithm steps are:

- Population initialization

The population is generated by random functions and hence is not uniform. Hence, diversity of the population diversity and ability to search is less.

- Population division
- Local search

  Only worst solution updated in SFLA. There is no updating method and principle for the best solution. Hence, convergence is slower. Also precision is not considered while updating the worst solution.
- Hybrid operation

  The frogs with the worst and the best fitness are represented respectively as X_W and X_b . If, S is size of leap step, $S_{\max}$ is maximum jump distance allowed and r is a random number in (0, 1), then the update equations are:

$$S = r(X_b - X_W) \quad (1)$$

$$X'_W = X_W + S, (S < S_{\max}) \quad (2)$$

2.2 Modified Forms

A. JSFLA

Modified forms of SFLA as proposed by Jiang et.al is termed here in this paper as JSFLA.

To meet the limitations of the SFLA, Jiang et.al [14] proposed a modification. Population initialization is improved by dividing the solution space into *p* parts consisting of randomly generated individuals. In this way the population is both random and uniform. In this modified form the population is segregated into searching population and competing population. The best and the second best solution are updated after evolution operation; the simplex method is used to update the second best solution. The convergence speed is improved by adding a Mutation operator to act on the individuals having lower convergence. To improve adaptability, updating rule of equation (1) is changed as:

$$S = r(X_b - X_w) * w_i \quad i = 1,2,\cdots,N \quad (3)$$

Here, N is population size and w_i is the compressibility factor [13] defined as:

$$w_i = w_l + (w_h - w_l) * X_i \quad (4)$$

Here, w_h and w_l respectively represent highest and lowest compressibility factor available in the population of frogs. Here, X_i is a factor called relative fitness of a particular frog and defined as:

$$X_i = \frac{X_w - X_{i,n}}{X_w - X_b} \quad (5)$$

$X_{i,n}$ is the fitness of frog 'I' in nth generation.

B. ZSFLA

Modified forms of SFLA as proposed by Zhang et.al is termed here in this paper as ZSFLA.

Zhang et.al [19] proposed a post heuristic SFLA algorithm. This algorithm found to improve the convergent velocity. Here, global information exchange and local depth search is used to find find global solution. The paper made it possible using the "follow" property of Artificial Fish-Swarm Algorithm (AFSA) [20-21] in the stage of global information interchange.

AFSA is based on swarm of fish for maximum food location.

Fish in order to search for maximum food location observes two of its properties, "follow: property and "swarm" property. The first, "follow" property of fish helps a faster advancement towards location of maximum food. In AFSA, this property helps in improvement in the speed of convergence and avoids oscillation around local optima. The second property, "swarm" property, avoids local optima and converges to the global optima. If these two properties fail, the prey behavior becomes a random movement.

In ZSFLA, these two properties AFSA can be inserted in SFLA in the stage of global information exchange. This improves convergence characteristics of SFLA and avoids falling to local optima. At the same time, since the prey behavior is random walk Gaussian variation is used to comprehend the prey behavior.

The variation equation is:

Mutation $(x) = x\left(1 + \frac{N(0,1)}{2}\right)$

Here, N(0, 1) is normal distribution random vector having expectation is 0 and standard deviation is 1.

Main steps of ZSFLA are as follows:

1. Initialize the group of frogs as population with size is N individuals of dimension m and M numbers of sub-populations. The number of iterations of local search in sub-group is cyc , the iterations of global interchange is $all\ cyc$, cognitive distance is $Visual$, moving step length is $Step$ and crowded degree factor is $\delta(0 < \delta < 1)$, the distance between Artificial Fish individuals is $d_{i,j} = \|X_i - X_j\|$.
2. Fitness of each of the individuals and mean fitness of all individuals is to be calculated. For the individuals having leower fitness than mean, "follow" property iis to be adopted. This helps in choosing a X from the neighborhood ($d_{i,j} < Visual$) out of the best with Y_j . If $Y_j / nf > Y . \delta$, it becomes X and steps towards X , $X_{new} = X_i + rand(\ ) \times Step \times (X_j - X_i) / \|X_j - X_i\|$, otherwise apply fitness equation to generate X_{new}. If $f(X_{new}) > f(X_j)$, replace individuals before follow with ones after follow. Otherwise, they remain unchanged.

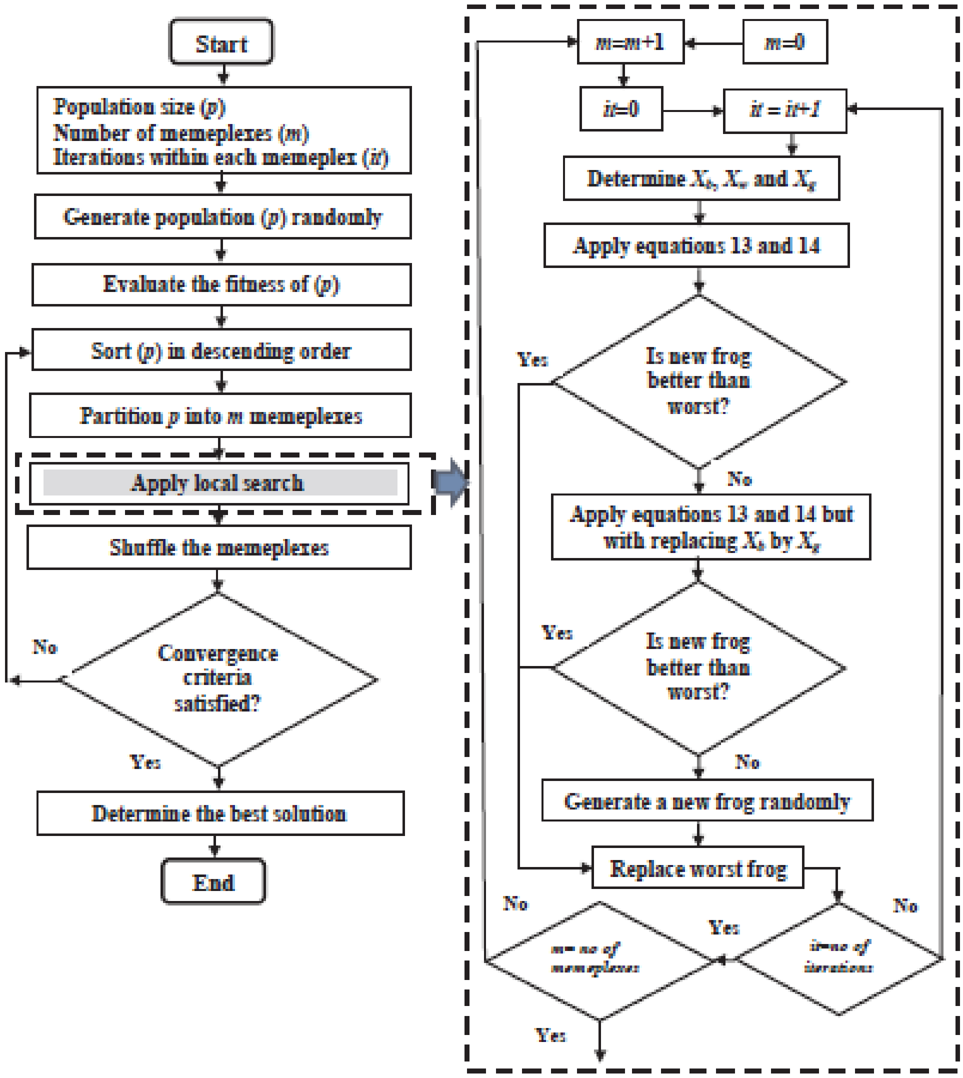

Fig. 1: Flow chart for SFLA [18]

3. Individuals are arranged with a decreasing order of fitness values and then divided into sub-groups.
4. Worst individuals of the sub-population is to be updated using "Swarm" property in all individuals of current sub-population, that is as for individual X_i, search the partner number nf and central location X_C during the current neighborhood ($d_{i,j} < Visual$), if $Y_C / nf > Y.\delta$, it is denoted that X_j has lower surrounding crowded degree, resulting in stepping forward towards the central location of partner, $X_{new} = X_i + rand(\ ) \times Step \times (X_C - X_i) / \|X_C - X_i\|$, other-wise yield Xnew using fitness equation). If $f(X_{new}) > f(X_j)$, individuals before Swarm are replaced by ones after, or otherwise, they remain unchanged. Jump to step (4) until the number of iterations satisfies cyc which has been given beforehand.
5. If all sub-populations are updated and if current optimal individual meets the convergence criteria, then the evolution stage concludes successfully with a pointer to go to step (6). Else, if global interchange iteration is not all cyc, go to step (2).
6. Record the output global optimal solution.

C. KSFLA

Modified forms of SFLA as proposed by Kavousifard et.al is termed here in this paper as KSFLA.

Kavousifard et.al in their work [22] proposed to select a modification by updating of the worst frog using a new vector. This new vector considers three factors; the best frog from the population, best individuals in each memeplex and a randomly generated control factor, β.

In SFLA, the information with a frog is improved increasingly in a virtual population without change of the physical uniqueness of the frogs.

With this exchange of idea among the frogs and hence influence each other and thereby their memes improved a heuristic search is completed.

Each solution frog $\bar{X}$ is defined as:

$$\bar{X} = [w_1, w_2, \cdots\cdots, w_{Nw}, b_1, b_2, \cdots\cdots, b_{Nb}]$$

Here, w_i and b_i are the weighting and biasing factors. N_w and N_b are the number of weighting and biasing factors.

The KSFLA algorithm follows following steps.

1. An initial population of n frogs randomly is defined.
2. Fitness of the initial population calculated. Frogs are sorted in descending order based on fitness.
3. Population of frogs is divided into m memplexes where each memeplex have p frogs and hence, $n = m \times p$.
4. Worst frog of each memplex is improved. In the j^{th} memplex the frog with the best and worst fitness are denoted by $\overline{X}_B$ and $\overline{X}_w$ respectively . The best frog in the entire population denoted as $\overline{X}_G$

$$\left.\begin{aligned} \overline{X}_{B,j} &= \left[x_{Bj.1}, x_{Bj.2} \cdots\cdots, x_{Bj.N}\right] \\ \overline{X}_{w,j} &= \left[x_{wj.1}, x_{wj.2} \cdots\cdots, x_{wj.N}\right] \\ \overline{X}_{G,j} &= \left[x_{Gj.1}, x_{Gj.2} \cdots\cdots, x_{Gj.N}\right] \end{aligned}\right\} \quad (6)$$

Here, N is the number of memotypes.

With aim to improve the worst frog of each memplex, the following modification is introduced in the KSFLA:

At first a frog ($\overline{X}_{Br}$) is selected from the best individuals in each memplex $\overline{X}_{B.j}$ ($j = 1,2,\cdots p$)in a way that $\overline{X}_{Br} \neq \overline{X}_G$. Now by the use of following equation a new vector ($\overline{X}_Q$) is generated.

$$\left.\begin{aligned} \overline{X}_Q &= \overline{X}_G + \beta\left(\overline{X}_{Br} - \overline{X}_G\right) \\ \overline{X}_{Q.j} &= \left[x_{Qj.1}, x_{Qj.2}, \cdots x_{Qj.N}\right] \\ \overline{X}_{Br} &= \left[x_{Br.1}, x_{Br.2}, \cdots x_{Br.N}\right] \end{aligned}\right\} \quad (7)$$

Here, β is a random value generated in the range of (0.1,1.3).

Now the with the use of this new modified individual the worst frog position is modified as:

$$X_{Wj.i}^{new} = \begin{cases} x_{Bj.i} & \lambda_1 \leq \lambda_2 \\ x_{Qj.i} & otherwise \end{cases} \quad (8)$$

Here, λ_1 and λ_2 are random values in the range of (0,1) and $i = 1,2,\cdots,N$.

If the newly produced frog is better than the previous worst frog, then it replaces the previous one else the same process as (7) is repeated for $\overline{X}_Q$ and $\overline{X}_G$. Again if the new frog is better than the worst frog, then it replaces the worst else the position of $\overline{X}_{Wj}$ will be generated randomly.

5. Continue step 3 to a specific iterations.
6. Reshuffle all frogs and sort them again.
7. If the termination criterion is not met return to step 2, else stop the algorithm.

3. Proposed Training

SFLA and its modified forms find optimal number of RBFs, centres etc. This is basis for training of RBFNN. Each of these parameters are optimized using the proposed training stepsas follows:

i. Initialize population of memes. Each meme corresponds to a network. Allowed iterations as MaxIteration. Start the first iteration.
ii. Each meme to be decoded to one ANN, calculate the weights using pseudo–inverse method. Calculate the meme's fitness.
iii. Position to be updated running SFLA .
iv. Go to next iteration;
v. Go back to step ii until reaching the maximum number of iterations.

In this work, RBFNN trained with original SFLA is termed as SRBF, while trained with modified forms proposed by Jiang et.al [13], Zhang et.al [17] and Kavousifard et.al [20] are referred as SRBF-J, SRBF-Z and SRBF-K respectively for convenience of the reader.

4 RBFNN Equalizer

The system considered in this paper is shown in figure 1.

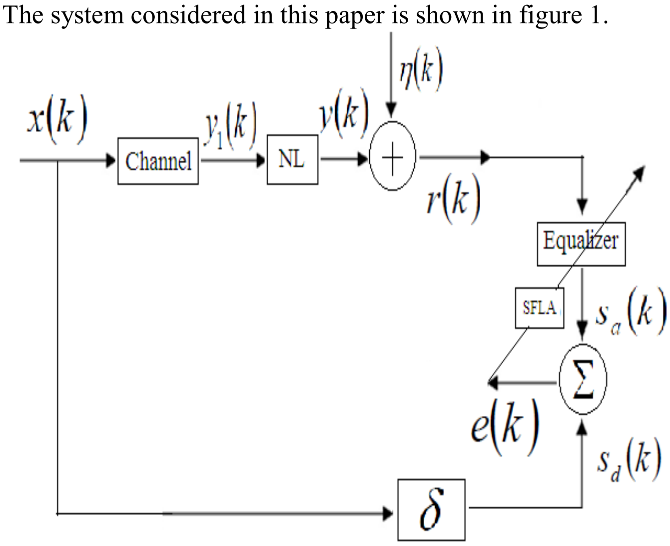

Fig. 2. Communication system with Equalizer.

A popular linear channel model is FIR model. In this model, transmitted sequence is binary, represented as $x(k)$ at k^{th} time instance and corresponding output at the same instant is, $y_1(k)$ as:

$$y_1(k) = \sum_{i=0}^{N-1} h_i x(k-i) \quad (9$$

Here, $h_i(i = 0,1,\cdots N-1)$ and N respectively are the channel taps and N is the channel length. The block 'NL' is nonlinearity inserted in the channel. Most popular form of nonlinear function is:

$$y(k) = F(y_1(k)) = y_1(k) + b[y_1(k)]^3 \quad (10)$$

Here, b is a constant. The output from this block is:

$$y(k) = \left(\sum_{i=0}^{N-1} h_i x(k-i)\right) + b\left(\sum_{i=0}^{N-1} h_i x(k-i)\right)^3 \quad (11)$$

The channel output $y(k)$ is added with noise, $\eta(k)$ inserted in channel. The signal at the receiver is $r(k)$ and given by:

$$r(k) = y(k) + \eta(k) \quad (12)$$

Equalizer is used to recover the transmitted symbol, $x(k-\delta)$, from a-priory knowledge on the samples received, ' δ ' being the associated transmission delay.

The desired signal can be represented as $d(k)$ and can be defined as:

$$d(k) = x(k-\delta) \tag{13}$$

Equalization can be treated as if a problem of classification [6-9], and the equalizer makes partition in the input space $x(k) = [x(k), x(k-1), \cdots x(k-N+1)]^T$ into two distinct regions.

The Bays theory [23] provides the optimal solution for this where the decision function is:

$$f_{bay}(x(k)) = \sum_{j=1}^{n} \beta_j \exp\left(\frac{-\left\|x(k)-c_j\right\|}{2\sigma^2}\right) \tag{14}$$

Since the sequence transmitted is binary, hence:

$$\beta_j = \begin{cases} +1 & c_j \in C_d^{(+1)} \\ -1 & c_j \in C_d^{(-1)} \end{cases} \tag{15}$$

Here, $C_d^{(+1)} / C_d^{(-1)}$ and c_j respectively represent transmitted symbol, $x(k-\delta) = +1/-1$ and σ^2 is the noise variance.

In figure 1, the block "Equalizer" is ANN. SFLA and its modified forms are used to optimize the number of layers and neurons in each layer. In the input layer there are N number of neurons, same as number of taps.

The equalizer output is:

$$f_{RBF}(x(k)) = \sum_{j=1}^{n} w_j \exp\left(\frac{-\left\|x(k)-t_j\right\|^2}{\alpha_j}\right) \tag{16}$$

Here, t_j and α_j represent the centers and the spreads of the neurons in hidden layers. The vector w_j contains the connecting weights.The output of the ANN equalizer of equation (6) implements the nonlinear function of equation (7), For optimal weights, the condition is t_j is equals to c_j.

The decision at equalizer output, is:

$$\hat{x}(k-\delta) = \begin{cases} +1 & f_{ANN}(x(k)) \geq 0 \\ -1 & elsewhere \end{cases} \tag{17}$$

Hence, the difference (i.e, $s_a(k) = \hat{x}(k-\delta)$) or (i.e, $s_d(k) = x(k-\delta)$) is the error, $e(k)$, and updates the weights.

For l is the number of samples then, Mean Square Error (MSE):

$$MSE = \frac{1}{l} E\left[e^2(k)\right] \tag{18}$$

Bit error rate (BER) is the ratio of error bits to transmitted bits: In this paper, MSE & BER are chosen as performance index.

.

5 SRBF Equalizer

The RBFNN equalizer is made of using channel output states instead of the channel parameter to avoid complexity in modeling the channel. The channel output states set is the data set for the centers. This relation exists on both linear channel and nonlinear channel with one to one mapping between channel input and channel output. On the other words, if the channel output states are known, the RBF equalizer is designed. The objective of the problem can be changed to find the optimal data set (channel output states). The problem of equalization is now a optimization problem with objective of reaching at Bayesian likelihood. But the relation between the output states of the channel and the Baye's likelihood is a complex or non-realizable formulation, when the structure of nonlinear channel is unknown, we apply genetic algorithm to solve this complex optimal problem with local minima.

Initial population of N ($m \times n$) memes is the basis for the SRBF equalizer, the channel states is initially chosen from. Each state constitutes the number of memeplexes and each meme represents one symbol that is assigned. As said in the introduction, objective of this paper is to develop SRBF based equalizer as shown in figure 2. Different forms of SFLA is used here as training algorithm for RBFNN..

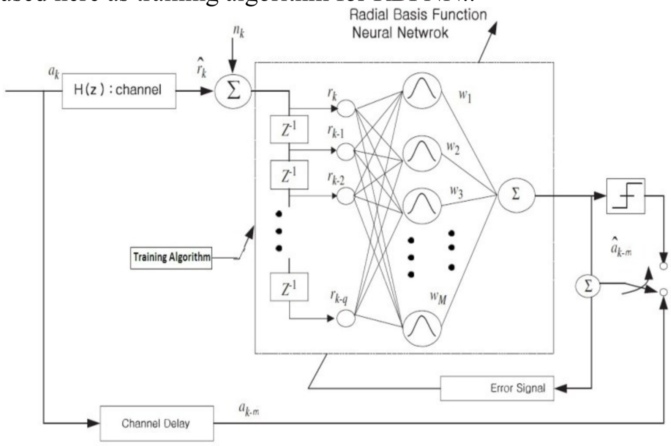

Fig. 3: SRBF Equalizer.

Table 1. Simulation Parameters.

GA		PSO		SFLA	
Parameter	Value	Parameter	Value	Parameter	Value
Number of iteration	1000	Number of iteration	1000	Number of iterations	1000
Number of individuals	50	Number of particles	50	Population	50
Mutation ratio	0.03	Coefficient C1	0.7	Number of memeplexe	10
Crossover ratio	0.9	Coefficient C2	0.7	Number of Memes in Memeplexes	10
Mutation type	Uniform			Number of Memes in Sub-Memeplexes	08
Crossover type	Single point			Number of Memetic evolutions in each submemeplex	10

6 Simulation Results

For evaluation of performance of proposed SRBF equalizers, results of contemporary RBFNN GA trained RBFNN (GRBF) [7] and PSO trained RBFNN (PRBF) [8] based equalizers are reproduced for the purpose of comparison. Parameters selected for the simulations illustrated in table 1.

Simulations were conducted for the most popular distorted channel with transfer function:

$$H(z) = 0.26 + 0.93z^{-1} + 0.26z^{-2} \qquad (20)$$

The equalizer performance is affected by channel nonlinearity. This effect studied in this paper introducing the nonlinearity:

$$y(n) = \tanh[x(n)] \qquad (21)$$

MSE was evaluated with a fixed SNR of 10dB.

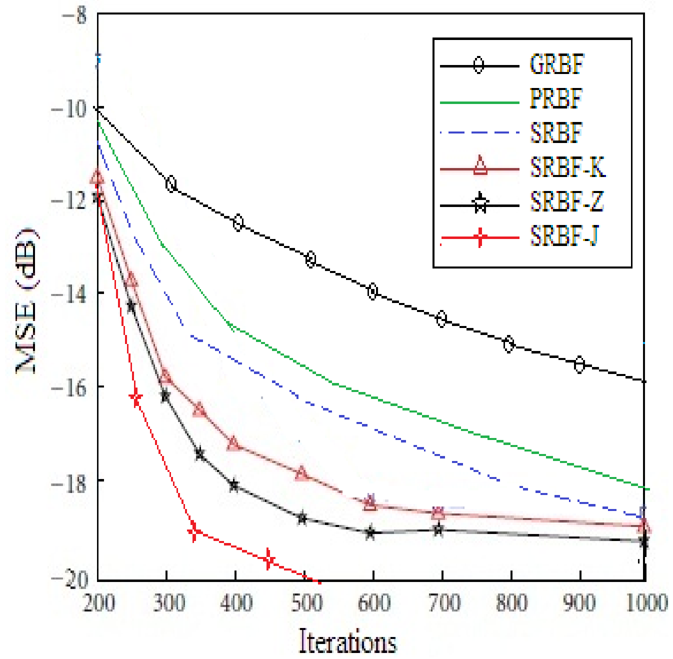

Fig. 4: MSE of RBFNN equalizers.

Figure 4 depicts the convergence of different equalizers at 10dB. It is observed from the figure that, proposed SRBF-J outperforms other equalizers. It is also seen that, RBFNN trained with SRBF and its modified forms are better than as trained with other nature inspired algorithms like GA and PSO. It is also observed that SRBF-J is a better method for training of RBFNN equalizer as compared to original as well as other modified forms of SFLA. It was observed that, SRBF-J requires only 500 iterations to converge while other equalizers fail to converge within 1000 iterations.

BER comparison among RBFNN based equalizers is depicted in figure 5.

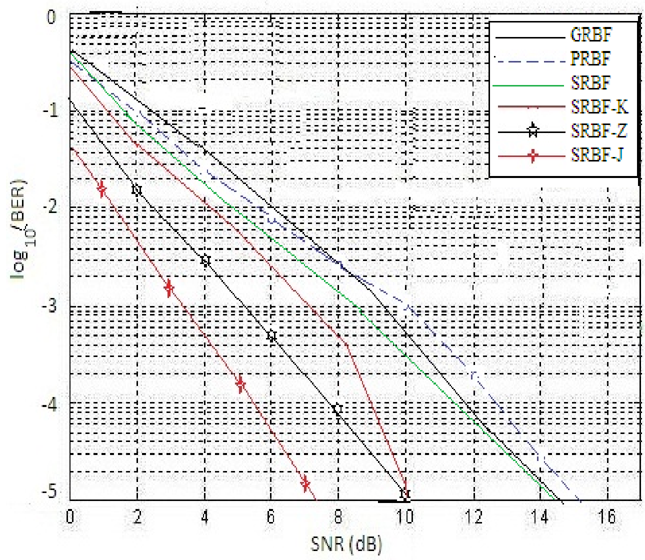

Fig. 5. BER of RBFNN equalizers.

It is seen from figure 4 that, performance of GRBF and PRBF are comparable to each other. SRBF equalizers perform better than GRBF and PRBF. Once again SRBF-J performs better than SRBF, SRBF-Z and SRBF-K equalizers with while SRBF-Z performs better than SRBF-K. It was observed that, SRBF-J achieves tolerable error rate for SNR more than 8dB. The same is achieved by SRBF-Z and SRBF-K with an SNR of 10dB.

7 Conclusion

This paper introduced novel strategy for RBFNN training using SFLA and its modified forms. This paper also proposed some efficient methods for equalization as proved in simulations. Major contributions by this paper are, RBFNN training using SFLA and its modified forms, use of SRBF in channel equalization and comparison among SFLA and its modified forms while training RBFNN. Significance of the works carried out in this paper as compared to existing RBF based equalizers is that of a better learning and generalization of the RBF network. Performance of SRBF based equalizer also better than the existing equalizers as seen from the simulations.

It is also found from the results that SRBF-J performs better than other forms of SFLA. Extension of the work with other kinds of population based algorithms and also other variants of SFLA may yield better results and a pointer for the future research.

During the course of publication of this paper, some more kinds of modified and improved forms of SFLA came to the literature and not discussed in this paper. The same can be seen as apointer for future ressearch.

References

[1] Burse K., Yadav R. N., and Shrivastava S. C., 2010, Channel Equalization Using Neural Networks: A Review, IEEE Trans. On Systems, man and cybernetics-Part C: Applications and Reviews, Vol. 40, No. 3, pp.352-357.

[2] Subramanian K, Savitha R, Suresh M, 2014, A complex-valued neuro-fuzzy inference system and its learning mechanism, Neurocomputing, Vol. 123, pp. 110-120.

[3] Ruan X, Zhang Y, 2014, Blind sequence estimation of MPSK signals using dynamically driven recurrent Neural Networks, Neurocomputing, Vol. 129, pp. 421-427

[4] Cui M, Liu H, Li Z, Tang Y and Guan X, 2014, Identification of Hammerstein model using functional link artificial_Neural Networks, Neurocomputing, Vol. 142, pp. 419-428

[5] Yaghini M, Khoshraftar MM, Fallahi M, 2013, A hybrid algorithm for artificial neural network training, Engineering Applications of Artificial Intelligence, Vol. 26, Issue 1, pp. 293-301.

[6] Chen S, Mulgrew B, Grant P. M, 1993, A Clustering Technique for Digital Communications Channel Equalization Using Radial Basis Function Networks, IEEE Transactions on Neural Networks, Vol 4, pp, 570-579

[7] Gan, M., Peng, H. and Chen, L. 2012, Global–local Optimization Approach to Parameter Estimation of RBF-type Models. Information Sciences. Vol.197, pp.144-160.

[8] Çiviciοğlu P., Alç M and Beşdok E, 2005, Using an Exact Radial Basis Function Artificial Neural Network for

Impulsive Noise Suppression from Highly Distorted Image Databases, Lecture Notes in Computer Science, Vol. 3261, pp 383-391

[9] Schilling RJ, Carroll JJ Jr and Al-Ajlouni AF, 2001, Approximation of Nonlinear Systems with Radial Basis Function Neural Networks, IEEE Transactions on Neural Network, Vol. 12, No. 1, pp.1-15.

[10] Yavuz, O. and Yildirim, T., 2008, Design of digital filters with bilinear transform using neural networks," 16th IEEE Conference on Signal Processing, Communication and Applications, pp.1-4.

[11] Barreto AMS, Barbosa, H.J.C. and Ebecken, N.F.F., 2002, Growing Compact RBF Networks Using A Genetic Algorithm, *In Proceedings of the VII Brazilian Symposium on Neural Networks*, pp. 61-66.

[12] Feng H.M., 2006, Self-generating RBFNs Using Evolutional PSO Learning, *Neurocomputing*, Vol. 70, pp. 241-251.

[13] V. Bergh, A.P. Engelbrecht, A new locally convergent particle swarm optimizer, Proceedings of IEEE International Conference on Systems, Man, and Cybernetics, (2002) 96-101

[14] Jiang J, Su Q, Li M, Liu M and Zhang L, 2013, An Improved Shuffled Frog Leaping Algorithm, Journal of Information & Computational Science Vol. 10: No.14 pp.4619–4626

[15] Eusuff, M.M. and Lansey KE, 2003, Optimization of water distribution network design using the shuffled frog leaping algorithm. J. Water Resources Planning Management. Vol. 129, No 3, pp.210–225.

[16] Kumar JV and Kumar DM, 2014, Generation bidding strategy in a pool based electricity market using Shuffled Frog Leaping Algorithm, Applied Soft Computing, Vol. 21, pp. 407-414

[17] Panda S, Panigrahi SP, A New Training Strategy for Neural Network Using Shuffled Frog-Leaping Algorithm and Application to Channel Equalization, Elsevier AEÜ-International Journal of Electronics and Communications, in press.

[18] Fattahi H, Bazdar H, Applying improved artificial neural network models to evaluate drilling rate index, Tunnelling and Underground Space Technology 70 (2017) 114–124

[19] Zhang X, Zhang Y, Shi Y, Zhaoa L and Cairong Zou, 2012, Power control algorithm in cognitive radio system based on modified Shuffled Frog Leaping Algorithm, , Vol. 66 pp.448– 454

[20] Li X-l, Feng S-h, Qian J-x. An optimizing method based on autonomous animates: fish-swarm algorithm. Systems Engineering: Theory & Practice 2002;22 (11):112–5.

[21] Li X-l. A new intelligent optimization method—Artificial Fish School Algorithm. Ph.D. dissertation. Hangzhou: Zhejiang University; 2003.

[22] Kavousifard A and Samet H, 2011, A Novel Method Based on Modified Shuffled Frog Leaping Algorithm and Artificial Neural Network for Power System Load Prediction, in D. Ryzko et al.: Emerging Intelligent Technologies in Industry, SCI 369, pp. 35–46. springerlink.com.

[23] Bayes, T. 1764. "An Essay Toward Solving a Problem in the Doctrine of Chances",*Philosophical Transactions of the Royal Society of London* **53**, 370-418.