Channel Equalization in Wireless Communication Systems using Metaheuristic Algorithms and Machine Learning Based Approaches

Submitted in partial fulfilment of the requirements for the award of the degree of

Doctor of Philosophy

by

Kishor Kisan Ingle

(**Reg No. 716144**)

Supervisor

Dr. Ravi Kumar Jatoth

Associate Professor,

Dept. of ECE

Department of Electronics & Communication Engineering

NATIONAL INSTITUTE OF TECHNOLOGY

Warangal-506004, Telangana State, India.

September 2021

APPROVAL SHEET

This thesis entitled "**Channel Equalization in Wireless Communication Systems using Metaheuristic Algorithms and Machine Learning Based Approaches**" by Mr. **Kishor Kisan Ingle** is approved for the degree of **Doctor of Philosophy**.

Examiners

____

____

Supervisor

____

Dr. Ravi Kumar Jatoth

Associate Professor, Electronics and Communication Engineering Department,

NIT WARANGAL

Chairman

____

Prof. L. Anjaneyulu

Head, Electronics and Communication Engineering Department,

NIT WARANGAL

Date:

Place: Warangal

DECLARATION

This is to certify that the work presented in the thesis entitled “Channel Equalization in Wireless Communication Systems using Metaheuristic Algorithms and Machine Learning Based Approaches” is a bonafide work done by me under the supervision of Dr. Ravi Kumar Jatoth, Department of Electronics and Communication Engineering, National Institute of Technology Warangal, and was not submitted elsewhere for the award of any degree.

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/date/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Kishor Kisan Ingle
Roll No: 716144

Date:
Place: Warangal

Department of Electronics and Communication Engineering

National Institute of Technology

Warangal – 506 004, Telangana, India

CERTIFICATE

This is to certify that the dissertation work entitled **“Channel Equalization in Wireless Communication Systems using Metaheuristic Algorithms and Machine Learning Based Approaches”**, which is being submitted by Mr. Kishor Kisan Ingle (Roll No.716144), is a bonafide work submitted to National Institute of Technology Warangal in partial fulfilment of the requirement for the award of the degree of **Doctor of Philosophy in Electronics and Communication Engineering**.

To the best of our knowledge, the work incorporated in this thesis has not been submitted elsewhere for the award of any degree.

Dr. Ravi Kumar Jatoth
Research Supervisor
Associate Professor
Department of ECE
National Institute of Technology
Warangal – 506004

Dedicated to My Mother

Contents

Acknowledgements

First of all, I am very much greatful to my research supervisor Dr. Ravi Kumar Jatoth for his support during my research work. His cooperation, encouragment and support helped me to complete my thesis within the stipulated time.

I am also grateful to Prof. L. Anjaneyulu, Head of the Department, Dept. of Electronics and Communication Engineering, for his support in my reseach period.

I take this privilege to thank all my Doctoral Scrutiny Committee members, Prof. T. Kishore Kumar, Professor, Department of Electronics and Communication Engineering, Dr. V. V. Mani Associate Professor, Department of Electronics and Communication Engineering, Dr. P. Muthu, Associate Professor, Department of Mathematics, Dr. M. Udaya Bhasker, Associate Professor, Department of Electrical Engineering, for their constructive suggestions during the progress of this research work.

I am indebted to the former Heads of the ECE department Prof. N. Bheema Rao and Prof. T. Kishore Kumar for their support and cooperation. I would also like to aknowledge the support from teaching, non-teaching members of Dept. of E.C.E. of N.I.T. Warangal.

I would like to thank my close friends for their support during my PhD. Thanks to Dr. N Hathiram, Dr. Sudeep Surendran, Dr. Uday Bagale, Ahadith, Dr. Jailsingh Bhookya, Prathap Soma, Mushahid Majid, D Srikar, Santanu Sasidharan, Sachin Wankhade, Uday Bagale, Abhay Lingayat and Ganesh Gawale for their motivation and support throughout my work.

I would like to thank my family members for giving me mental support and inspiration. They have motivated and helped me to complete my thesis work successfully.

Finally, I am thankful to god for giving me strength to completet this research work

Kishor Kisan Ingle

ABSTRACT

In the last few years, the field of wireless communication has seen a tremendous development in various technologies. The extensive use of internet applications resulting in increased demand for higher data rate. Eventually, the higher data transmission rates are making the problem of inter-symbol interfernce (ISI) more severe. Thus, to mitigate the effect of inter-symbol interference (ISI) in multipath wireless channels, designing a channel equalizer is becoming more demanding. Furthermore, the transmitted signal also subjected to noise and non-linear distortion from the channel and signal processing devices. Therefore, designing of efficient non-linear channel equalizers is a need of time. In the literature, several non-linear channel equalizers are have been developed. However, it still requires further investigation to improve equalization performance in terms of bit error rate (BER) by improving the efficiency of the training algorithms used for channel equalizers. Therefore, this thesis investigates the various metaheuristic algorithms and machine learning based approaches for non-linear channel equalization.

In this thesis, a new training scheme using cuckoo search algorithm (CSA) has been proposed to train the neural network based channel equalizers. The performance of the proposed scheme also has been analysed over different non-linear wireless communication channels. The robustness of a scheme is shown by considering a burst error scenario. An efficient JAYA algorithm with Levy flight has been proposed for non-linear channel equalization and the performance of the proposed algorithm is examined under wireless communication channels with different eigenvalue (EVR) ratio and non-linerarities. Furthermore, a modified grasshopper optimization algorithm has been porposed and non-linear channel equalization in wirelss environments have been carried out using a proposed modified grasshopper optimization algorithm. Finally, a neural network based approach is proposed for joint channel estimation and data detection in universal filtered multi-carrier (UFMC) system and the efficiency of the proposed approach have been confirmed over wireless channel.

List of Figures

List of Tables

Nomenclature

AWGN	Additive White Gaussian Noise
BER	Bit Error Rate
SNR	Signal-to-Noise Ratio
ISI	Inter-symbol interference
NN	Neural Network
FIR	Finite Impulse Response
CSA	Cuckoo Search Algorithm
MSE	Mean Squared Error
MLP	Multilayer perceptron
CP	Cyclic Prefix
BP	Back-Propagation
PSO	Particle Swarm Optimization
DE	Differential evolution
GWO	Grey Wolf Optimizer
LF	Levy flight
CPU	Central Processing Unit
NL	Non-linearity
WNN	Wavelet Neural Network
EVR	Eigen Value Ratio
WOA	Whale Optimization Algorithm
LS	Least Square
MMSE	Minimum Mean Square Error

Chapter 1

Introduction

1.1 Background

In the last decade, widespread use of internet has resulted in a massive rise in the data rate of a wireless communication system. Eventually, to mitigate the effect of inter-symbol interference (ISI) in multipath wireless channels, designing a channel equalizer is becoming more demanding. Generally, in wireless communication, the information is broadcasted via band-limited channels [1]. Thus, if the transmitted signals have higher bandwidth than the channel coherence bandwidth, it results in amplitude and phase distortion of the signals, which causes ISI [2–4]. Moreover, multi-path effects in the wireless environment and the band-limited nature of channels are the key factors which lead to ISI [5]. The noise introduced by the system during transmission of data and the non-linear distortion arising from the use of amplifiers also needs to be alleviated [6,7]. Hence, an equalizer is needed at the receiver to combat the distortion owing to ISI, noise and nonlinearity [2–4].

1.2 Motivation

In recent years, the problem of channel equalization has been solved by many researchers. Initially, communication researchers employed adaptive filter based linear equalizers for channel equalization in wireless environment. However, linear equalizers does not provides better BER performance for non-linear wireless channels [1] [8–10]. The neural network (NN) based equalizers provides superior performance owing to their non-linear structure [1] [8–10]. In general, gradient-decent algorithms are utilized to train neural network (NN) for non-linear channel equalization in wireless environment [1,11,12]. However, the use of gradient based algorithms such as back-propagation (BP) for training the NN based non-linear equalizer results in the performance degradation because of certain key factors like BP has a problem of stagnation in local minima [13–15], learning rate parameter and initial coefficients affect the convergence, and slower convergence rate of BP [16]. These limitations motivated the researchers to employ nature-inspired metaheuristic algorithms to train NN based channel equalizers [17–19]. The capability to deal with the problems which are nonlinear, non-differentiable and complex is present in metaheuristic algorithms whereas gradient-descent based approaches need differentiable and continuous fitness function [15]. Owing to the stochastic nature of population based metaheuristic algorithms, they are capable of escaping from local optima [16] and provides faster convergence [20]. However, it still requires further investigation to improve equalization performance in terms of bit error rate (BER) by improving the efficiency of the training algorithms used for channel equalizers. This motivated us to propose new training schemes for neural network based non-linear channel equalizers using population based metaheuristic algorithms.

1.3 Research Objectives

The key research objectives of the research work carried out in this thesis are as follows:

- ❖ To propose a new training scheme for Neural Network based Non-linear Channel Equalizers using Cuckoo Search Algorithm
- ❖ To develop an efficient JAYA Algorithm with Levy flight for Non-linear Channel Equalization
- ❖ To develop a Modified Grasshopper Optimization Algorithm for Non-linear Channel Equalization
- ❖ To develop a Neural Network based approach for joint channel estimation and data detection in multi-carrier system.

1.4 Contributions of the Thesis

The key contributions of this have been briefly explained as follows:

1.4.1 A New Training Scheme for Neural Network based Non-linear Channel Equalizers using Cuckoo Search Algorithm

This contribution deals with development of a new training scheme using Cuckoo Search Algorithm (CSA) to train the neural network based non-linear channel equalizer. To overcome the limitations of existing algorithms, this work proposes a training scheme using Cuckoo Search Algorithm (CSA) for neural network based channel equalizers. The proposed training scheme has a better ability to escape from local minima, higher exploitation and exploration capabilities. To choose the optimum values of the parameters, the sensitivity analysis of the proposed approach is performed with its key parameters. Furthermore, three non-linear channels have been simulated to demonstrate the equalization performance of the CSA based training scheme and the results have been compared with recent and well-established algorithms. The simulations confirm that the proposed training scheme performs substantially better than existing metaheuristic algorithms in terms of BER and MSE performance. To show the robustness of the proposed method, the burst error scenario has been considered and results proved that the method is more successful in handling such scenarios when compared to other methods. The performance of the proposed scheme has been validated for a wide range of signal-to-noise ratio through simulation studies and it is observed that the scheme outperforms the other algorithms in poor SNR conditions as well. Also, to examine the statistical significance of the results provided by the proposed scheme, the Wilcoxon test is performed and the test reveals that the obtained results are statistically significant

1.4.2 An Efficient JAYA Algorithm with Levy flight for Non-linear Channel Equalization

This contribution involves the development of an efficient JAYA algorithm with Levy flight for Non-linear Channel Equalization. JAYA is an effective and simple population based metaheuristic algorithm. Despite being an efficient and simple algorithm, JAYA gets trapped into local optima owing to its weak exploration competence and inadequate solution diversity. To mitigate these issues, in this contribution Lévy flight (LF) concept and greedy selection scheme has been incorporated into the basic JAYA. The LF concept enhances the population diversity and thus avoids the state of stagnation. The greedy selection scheme is employed to improve the exploitation ability without loss of population diversity. Furthermore, the exploitation and exploration capabilities of the algorithm have been balanced by proposing an adaptive Lévy index using a linear control parameter strategy. The sensitivity analysis of proposed method called JAYA algorithm with Lévy flight (JAYALF) with its key parameters is carried out to select the optimal values for these parameters. In order to validate the local optima avoidance ability, exploitation and convergence rate of the proposed JAYALF algorithm, it is tested on unimodal and multimodal benchmark functions and to verify the effectiveness of the

JAYALF for non-linear channel equalization problem, three non-linear wireless communication channels have been considered for simulations. In addition, the non-parametric pairwise Wilcoxon rank-sum test has been employed to test the statistical validity of the results obtained from JAYALF. The results of experiments and statistical test demonstrate that the proposed algorithm significantly outperforms existing algorithms in terms of convergence rate and accuracy. Furthermore, simulations show that proposed JAYALF algorithm provides faster convergence without being trapped into local optima and has a better exploration ability.

1.4.3 Non-linear Channel Equalization using a Modified Grasshopper Optimization Algorithm

In this contribution, a modified grasshopper optimization algorithm (MGOA) is proposed for equalization of non-linear wireless channels. The proposed algorithm overcomes the limitations of grasshopper optimization algorithm and existing metaheuristic algorithms. The superiority of the proposed MGOA based equalizer is illustrated over other equalizers optimized by the other metaheuristic algorithms. The simulation results on four non-linear communication channels demonstrate the efficiency of the proposed MGOA algorithm in terms of MSE and BER performance. In order to test the statistical validity of the results obtained from MGOA, a non-parametric pairwise Wilcoxon rank-sum test has been employed and the test reveals that the obtained results are statistically significant.

1.4.4 Joint Channel Estimation and Data Detection for Multi-carrier System: A Neural Network based approach

In conventional multicarrier system to recover the transmitted symbols, channel estimation and data detection are carried out as a two different processes. However, the approach involves the use of pilots for detection of transmitted data. Furthermore, in the pilot based channel estimation approach it requires to explicitly model the channel using the available channel observations which may not be accurate always. To overcome these drawbacks, this work proposes a joint channel estimation and data detection approach for universal filtered multi-carrier Systems using neural network. The proposed approach provides better BER performance when compared to conventional channel estimation approaches.

1.5 Thesis Organization

This thesis has been organized into seven key chapters and the chapters are summarized in following section as

Chapter 1: This chapter provides the introduction to the channel equalization in a wireless communication system, motivation, and contributions of the thesis.

Chapter 2: In this chapter basic system model of non-linear channel equalization is provided along with necessary mathematical equations. Furthermore, introduction about machine learning is given by explaining the basic concepts related to the neural networks. This chapter also gives overview of metaheuristic algorithms. A comprehensive review of the literature available in the area of channel equalization is also have been discussed.

Chapter 3: In this chapter a training scheme using cuckoo search algorithm is proposed to train the neural network based non-linear channel equalizers. The performance of the proposed scheme also has been analysed in the burst error scenario.

Chapter 4: This chapter proposes an efficient JAYA algorithm with Levy for non-linear channel equalization. The peformance of the algorithm is examined in terms of MSE and BER performance.

Chapter 5: A modified grasshopper optimization algorithm is proposed in this chapter and the perfromance of the proposed algorithm has been evaluated in terms of BER and MSE performance

Chapter 6: This chapter proposes an neural network based approach for joint channel estimation and data detection of multicarrier system.

Chapter 7:Chapter 7 provides the conclusion of the research work and future scope to take this research to the higher level.

Chapter 2

Preliminaries and Literature Survey

2.1 Introduction

In this chapter basic system model of non-linear channel equalization is provided along with necessary mathematical equations. Furthermore, introduction of machine learning is given by explaining the basic concepts related to the neural networks. This chapter also presents the overview of metaheuristic algorithms. Finally, a comprehensive review of the literature available in the area of channel equalization is also have been discussed.

2.2 Non-linear Channel Equalization

A discrete-time model of non-linear channel equalization in the wireless communication system is demonstrated in Fig. 2.1. The transmitted symbols are considered as independent and equiprobable for all $s(n)$ and are taken as the random binary symbols in the form of {+1,-1}. The wireless channel block in this figure consists of the transmission medium along with the transmitter-side filter. Generally, the FIR model is utilized to characterize the linear channel and its output ($t(n)$) is expressed as follows [11][1]:

$$t(n) = \sum_{k=0}^{N_h-1} h(k)s(n-k) \tag{2.1}$$

where $h(k)$ denotes the wireless channel tap coefficients and N_h is the number of coefficients in the channel impulse response.

Furthermore, the NL block in Fig. 2.1 adds nonlinear distortion to wireless channel output and its output *b(n)* is given as follows:

$$b(n) = \psi(s(n), s(n-1), s(n-2), ...s(n-N_h+1); h(0), h(1), h(2).....h(N_h-1)) \tag{2.2}$$

where ψ represents the non-linearity generated using the 'NL' section

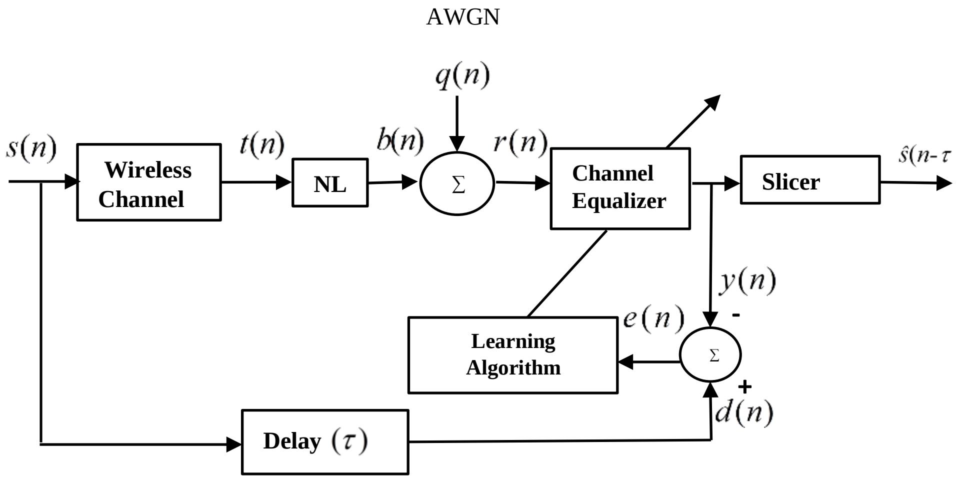


Fig. 2. 1**.** Block diagram of non-linear channel equalization

An additive white Gaussian noise q(n) degrades the channel output after adding non-linearity i.e. $b(n)$ Finally, the signal received at a receiver front end is $r(n)$ and can be written as

$$r(n) = \psi(s(n), s(n\text{-}1), s(n\text{-}2), ...s(n\text{-}N_h+1); h(0), h(1), h(2).....h(N_h-1)) + q(n) \qquad (2.3)$$

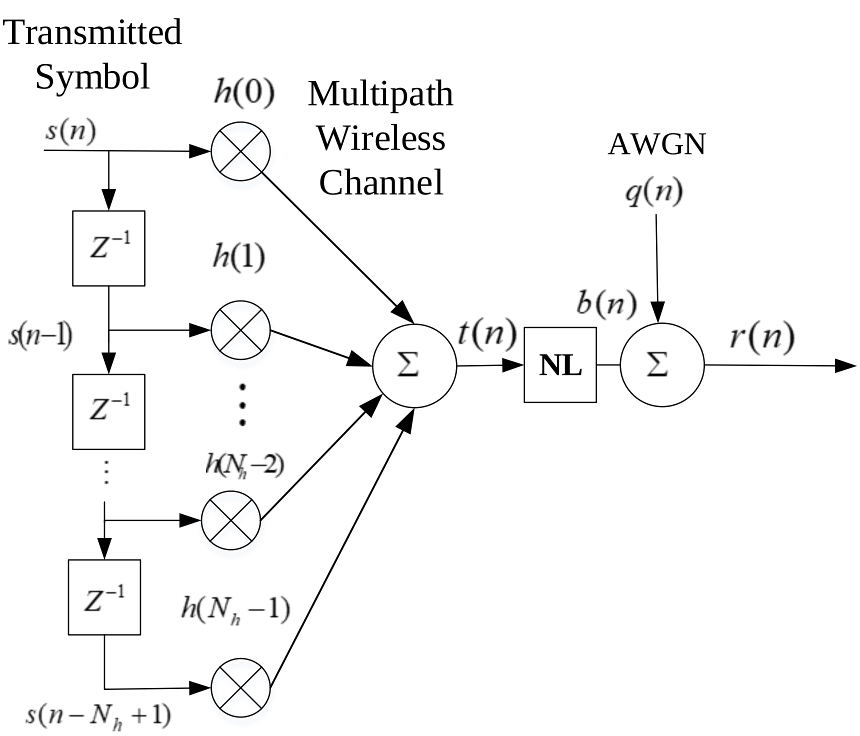


Fig. 2. 2. Multipath wireless channel with non-linear distortion and AWGN

The structural representation of multipath wireless channel along with non-linearity and AWGN is shown in Fig.2.2 .The equalizer recovers the sequence $s(n)$ or its delayed form i.e. $s(n-\tau)$, where τ is the delay related to wireless channel during transmission. The equalizer mitigates for noise, ISI, and nonlinear distortion to retrieve the $s(n)$. Furthermore, the desired signal $d(n)$ is generated from the input signal $s(n)$ by taking a delay of τ symbols i.e. $s(n-\tau)$ The error $e(n)$ is calculated by comparing equalizer output $y(n)$ with the desired signal $d(n)$ as follows:

$$e(n) = d(n) - y(n) \tag{2.4}$$

The slicer provides the estimate of the transmitted symbol and is written as follows:

$$\hat{s}(n-\tau) = \begin{cases} -1 & if \quad y(n) < 0 \\ 1 & if \quad y(n) \geq 0 \end{cases} \tag{2.5}$$

2.3 Machine Learning

The machine learning involves unsupervised learning, supervised learning, and reinforcement learning. However, in this thesis emphasis is given on supervised machine learning using the artificial neural network. The artificial neural network research has been started with the introduction of brain inspired computational model by McCulloch and Pitts [15,21]. In the last few decades, neural network have received significant attention from the researchers of various fields as they are capable of solving non-linear and complex engineering problems. The neural networks have been successfully applied to various engineering fields such as speech processing [22], signal processing [23], image processing [24] etc. The basic theory related to neural network in this section is taken from [15,25–28].

2.3.1 Single Neuron

The most basic form of neural network is a single layer perceptron which comprises of an input layer and output layer [15,27,29]. The basic neuron involves inputs, weights, activation function and single output.

The basic neuron has a number of inputs u_i, (i=1,2…n) and each input is associated with weights w_i (i=1,2…n). The weighted sum of inputs including bias is given as follows [25–27]:

$$Z = \sum_{i=1}^{n} w_i u_i + b \tag{2.6}$$

Where u is input, n represents the total number of inputs, w denotes the weight and b represents the bias

The final output of neuron Y can be written as follows:

$$Y = f(Z) \tag{2.7}$$

Where f denotes some non-linear activation function. The activation functions helps to limit the output of neuron in particular range. There are various activations fucntions are available such as sigmoid, hyperbolic tangent etc.

2.3.2 Neural Network Architectures

To solve the practical non-linear problems the neural network needs multiple layers of neurons rather than single neuron. Generally, the ANNs invovles input layer, output layer and hidden layer [25,26]. Furthermore, based on the connection of the neurons, neural networks architectures are catogorized as recurrent neural networks and feed-forward neural networks [25,27]. However, this study focuses on the feed-forward neural networks.

2.3.3 Training of Artificial Network Networks

In neural networks, the training process involves adjusting the weights of network to minimize the cost function. Traditionally, the training of the neural network is carried out using gradient- decent based algorithms [12].

2.3.4 Functional Link Artificial Neural Network (FLANN)

FLANN is a NN developed by Pao with no hidden layers, which has attracted the scientific community owing to its lower computational complexity and simplicity [30,31]. In FLANN, the input pattern is enhanced with the help of linearly independent non-linear functions which reduces the computational burden [1,32]. To make the signals linearly separable in the higher space, the input signals are converted into a higher-dimensional space [1,7,11,30,33,34]. Chebyshev, Legendre or trigonometric polynomials are used in this network to carry out the non-linear expansion of the input signal [1,7,11,30,34–36] .

Researchers utilized FLANN with different polynomials and proven that it outperforms other existing NNs for channel equalization [1,7,11,34]. Furthermore, adaptively combined FIR and FLANN and Chebyshev polynomials cascaded with FLANN have been employed for the equalization [32,35,37]. Recently, FLANN has been applied for function approximation [38], non-linear system identification[39–41], to solve differential equations [42] and for noise control [43].

In order to improve the representation of the input signal in higher-dimensional space, the expansion section in Fig. 2.3 expands the dimensions of the signal using trigonometric functions. The detailed description of FLANN can be found in [1,11,30,31,34,37].

Let $R(n) = [r_1 \;\; r_2]^T = [r(n) \;\; r(n-1)]^T$ be a input pattern. The Expansion section consists of trigonometric functions to expand this pattern as follows:

$$R^e(n) = [1 \;\; r_1 \;\; \sin(\pi r_1) \;\;\; \cos(\pi r_1) \;\; r_2 \sin(\pi r_2) \;\; \cos(\pi r_2)]^T$$

$$R^e(n) = [1 \;\;\; r(n) \;\;\; \sin(\pi r(n)) \;\;\; \cos(\pi r(n)) \;\;\; r(n\text{-}1) \;\; \sin(\pi r(n\text{-}1)) \;\;\; \cos(\pi r(n\text{-}1))]^T \tag{2.8}$$

Hence, the expanded pattern can be stated as,

$$R^e(n) = \Phi(R), \tag{2.9}$$

where $\Phi(R) = [\phi_1(R), \phi_2(R), \phi_3(R), \phi_4(R) ...... \phi_K(R)]^T$ is a vector of basis functions

Let $B = \{\phi_i \in L(A)\}$ $i \in I$, $I = \{1, 2....\}$ is a set of basis functions and a set B has the three main properties as follows: i) $\phi_1 = 1$ ii) $\sup_j \left[\sum_{i=1}^{j} \|\phi_i\|_A^2 \right]^{1/2} < \infty$ and iii) $B_K = \{\phi_i \in B\}_{i=1}^{K}$ is a linearly independent set [1].

The FLANN equalizer's coefficient vector is expressed as follows:

$$W(n) = \left[w_1(n) \; w_2(n) \; w_3(n) ..... w_K(n) \right]^T \tag{2.10}$$

The linear sum of expanded input is written as follows:

$$u(n) = W^T(n)R^e(n) \tag{2.11}$$

where $R^e(n) = \Phi(R) = [\phi_1(R), \phi_2(R), \phi_3(R), \phi_4(R) \ldots\ldots \phi_K(R)]^T$ is the vector of basis function

The FLANN output $y(n)$ is produced by passing the linear sum $u(n)$ through a nonlinear activation function and is given by,

$y(n) = \rho(u(n))$

$$y(n) = \tanh(u(n)) \tag{2.12}$$

The error signal $e(n)$ is computed by taking the difference between the output of FLANN $y(n)$ and the desired signal $d(n)$ and is written as follows:

$$e(n) = d(n) - y(n) \tag{2.13}$$

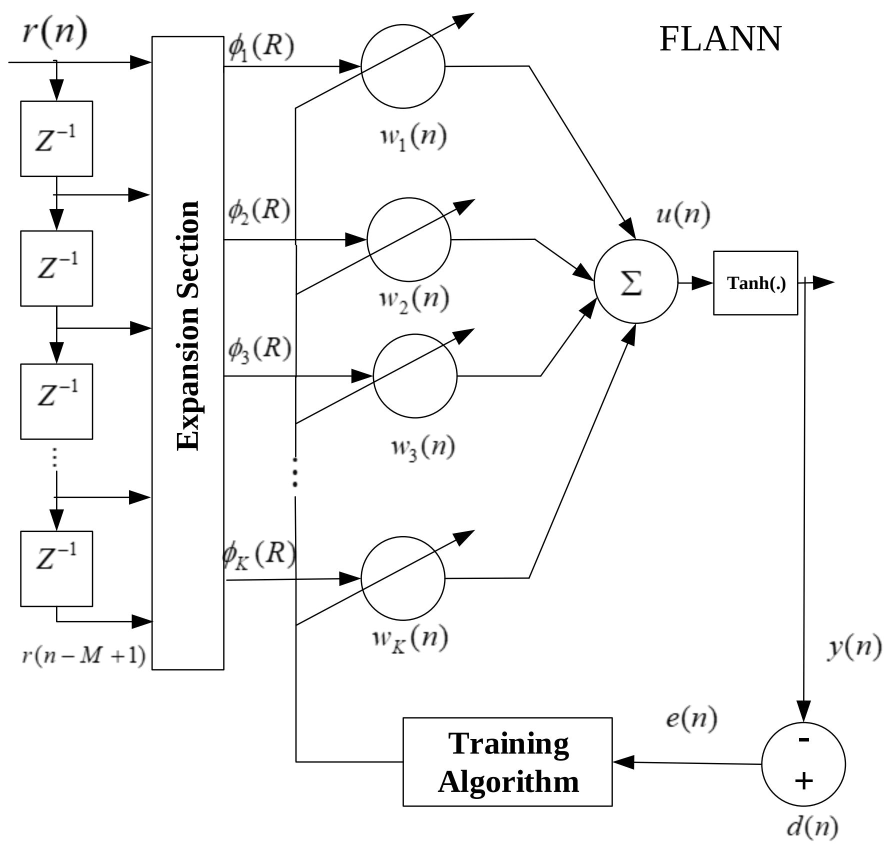


Fig. 2. 3. Block Diagram of FLANN [1,11,33]

2.4 Metaheuristic Algorithms

The technique which employs heuristics inspired by nature for integrating exploration and exploitation strategies is known as metaheuristic [15,44]. Exploration and exploitation are the key elements of any metaheuritic algorithms. Exploitation or intensification refers to using the knowledge of current good solution for directing the search in a local region [45]. On the other hand, exploration or diversification refers to exploring the solution space globally by generating the diverse solutions [45]. In the recent past, metaheuristic algorithms have gained enormous popularity among the research community of various engineering disciplines owing to their efficiency in finding the optimal solution. As a result of which metaheuristic algorithms have been successfully applied for various wireless communication and signal processing applications [46–48].

Primarily, metaheuristic algorithms can be classified into three main categories as follows [15]:

1)Single solution based metaheuristic algorithms

2) Population based or multi-solution based metaheuristic algorithms

3) Hybrid Algorithms

The single solution based algorithms consist of only one agent or solution whereas population based or multi-solution based metaheuristic algorithms consist of the number of solutions or agents [15]. The hybrid algorithms consist of combination of different metaheuristic algorithms or combination of conventional and metaheuristic algorithms to achieve the global optimality [15]. Simualted annealing is an example of single solution metaheuristic algorithm [49,50]. On the other hand, the algorithms such as genetic algorithm [51,52], particle swarm optimization (PSO) [53] and bacterial foraging optimization [54] are the examples of population based metaheuristic algorithms. In the recent past, several hybrid metaheuristic algorithms such as hybrid PSO and back-propagation algorithm [55] and hybrid genetic algorithm and PSO [56] have been proposed in the literature. This study focuses on the population based metaheuristic algorihms and brief description of the algorithms used in this study is given below.

2.4.1 Cuckoo Search Algorithm (CSA)

Cuckoo search algorithm (CSA) is one among the recent population based metaheuristic algorithms, which is proposed by Yang and Deb [57] . CSA has been inspired from the concept of brood parasitism of particular species of cuckoo [57][58][59]. Along with brood parasitism, this algorithm also involves Lévy flight movements observed in certain birds[57][58][60][59] . Lévy flight based jumps help CSA to avoid being trapped into local minima and to find the potential regions of solution space. The superior performance of CSA in terms of local optima avoidance capability has been shown by a number of studies when compared to other leading metaheuristics such as PSO, genetic algorithm, artificial bee colony algorithm and DE algorithm [61] [57]. CSA has been effectively used by the research community in various research fields due to its enhanced exploration and exploitation capabilities [58]. In the last few years, CSA has been widely used for multilevel thresholding in the image [62] , image enhancement [63], design of fractional order differentiator and FOPID controller [64][65] , spectrum allocation in a vehicular network [46], non-convex economic dispatch problem [66] and optimization of traffic signal controller [67]. Recently, CSA has been employed for system identification using the Hammerstein model and for feedback system identification [68][40] and superior performance of CSA has been reported over state-of-the-art methods.

2.4.2 JAYA algorithm

JAYA is one of the recent population-based metaheuristic algorithm, proposed by R. Venkata Rao in 2016 [69]. Besides being an effective and simple algorithm, JAYA does not need algorithm specific control parameters. The basic idea behind this algorithm is that the solution obtained for a problem should escape from the worst solution and should approach to the best one [70]. Since its introduction, owing to simplicity and ability to find global optimum solution JAYA has been successfully used by researchers in many engineering problems. In recent few years, JAYA and its improved versions [71–80] have been used to solve various engineering problems. Furthermore, JAYA has been used for power quality improvement [81], optimization of heat sink [82], tracking of maximum power point (MPP) of PV array [83], reliability–redundancy allocation problems [84], optimization of heat exchangers [85], mechanical design optimization [86], optimization of machining performance [87], parameter identification of photovoltaic model [88] and design optimization of heat exchangers [89].

2.4.3 Grasshopper Optimization Algorithm

Grasshopper Optimization Algorithm is one among the latest metaheuristic algorithms, developed by Saremi et al. in 2017 [90]. The GOA is an efficient swarm intelligence based algorithm motivated from the team hunting behavior of grasshoppers . Since its introduction, many complicated benchmark functions and engineering problems has been solved by GOA effectively [90,91]. GOA and its enhanced versions have been extensively used for electrical characterization of fuel cells [92], training of artificial neural network [91], feature selection [93–95], economic dispatch problem [96], data clustering [97], tuning of PID controller [98] and target tracking [99]. To enhance the performance of basic GOA, some improvement techniques have been introduced in [100–102], where the authors demonstrated the competitive performance of GOA compared to other metaheuristic algorithms. Recently, Mirjalili et al. [103] have developed a basic multi-objective GOA and Tharwat et al. proposed an improved version of multi-objective GOA [104].

2.5 Literature Survey

In order to alleviate the effect of ISI, R. W. Lucky introduced the first equalizer structure in the year 1965 which contains a tapped delay line and adaptive combiner [105]. Usually, the methods like recursive least square [106] and least mean square (LMS) [3][107] are used to tune the parameters of the adaptive combiner in a linear equalizer. However, most of the practical wireless channels are severely non-linear due to the presence of non-linearity in data converters [8][9][10]. Moreover, in satellite communication, amplifier saturation in satellite also contributes to non-linearity [108]. However, for severely non-linear and dispersive channels the performance of LMS based linear equalizers is poor [1,8–10]. Thus, to retrieve the information corrupted due to transmission through the non-linear wireless communication channels, the non-linear equalizers play an important role.

In recent years, the NN based channel equalizers emerged as promising alternatives to tackle the non-linearity in the wireless channels as they offer a lower bit error rate (BER) when compared to linear equalizers [8–10,109,110]. These attributes attracted the communication researchers to use several NNs for channel equalization in the wireless environment [8–10,109,111]. A multilayer perceptron (MLP) has shown superior performance than the linear equalizers [9][10]. Furthermore, networks like polynomial perceptron network [110] , radial basis function [109][112] and functional link artificial neural network (FLANN) with various polynomials [1,7,11,33,34,113] have been applied for equalization which are computationally efficient than MLP. In recent years, several nonlinear channel equalizers have been developed using FLANN which offer improved performance and complexity equivalent to the FLANN [32,35,37].

In general, the back-propagation (BP) algorithm is utilized to train NN for non-linear channel equalization [1,7,11,12,33,34]. However, the use of BP for training the NN based non-linear equalizer results in the performance degradation because of certain key factors like BP has a problem of stagnation in local minima [13–15], learning rate parameter and initial coefficients affect the convergence, and slower convergence rate of BP [16,114]. These limitations motivated the researchers to employ nature-inspired metaheuristic algorithms to train NN based channel equalizers [17–19,115]. The capability to deal with the problems which are nonlinear, non-differentiable and complex is present in metaheuristic algorithms whereas gradient-descent based approaches need differentiable and continuous fitness function [15] . Owing to the stochastic nature of population based metaheuristic algorithms, they are capable of escaping from local optima [16] and provides faster convergence [20]. Therefore, to evade the limitations of gradient-descent based approaches various signal processing and wireless communication problems were solved using population based metaheuristic algorithms [46–48,116,117].

By motivating from their advantages, several populaton based metaheuristic algorithms have been employed by communication researchers for equalization of wireless channels. In [118], researchers have developed a genetic algorithm (GA) based scheme for blind channel identification and it has been shown that the proposed scheme outperforms the existing methods. The joint channel and data estimation is implemented by chen and Wu in [119] with the micro genetic algorithm. The estimation of transmitted data is carried out by using the Viterbi algorithm after identifying the unknown channel with micro GA [119] and the proposed scheme has demonstrated better performance than the existing schemes. The hybrid genetic algorithm based approach has been used for non-linear channel blind equalization and it is illustrated that the proposed approach provide superior bit error rate perfroms than other existing variants of GA [120].

In last few years, population based metaheuristic algorithms also have been employed to design an adaptive filter based linear equalizers. In [121], authors have designed an adaptive equalizer based on particle swarm optimization (PSO) algorithm and demonstrated its superiority over least mean square (LMS) based equalizer in terms of MSE and BER performance. Furthermore, a modified version of PSO algorithm has been proposed in [108] for equalization of linear and non-linear wireless communication channels. It is shown that the proposed algorithm provide better performance when compared LMS and other versions PSO in terms of BER and MSE. In [122] , equalization of linear and non-linear communication channels has been carried out using hybrid PSO (HPSO) algorithm and superior performance of HPSO has been exhibited in terms of BER and convergence speed over LMS and other variants of PSO. An artificial immune system based equalizer has been employed in [123] for equalization of wireless channels and notable performance of proposed equalizer is demonstrated against GA and LMS based equalizer. Recently krill herd algorithm [124] and artificial bee colony algorithm [125] and bacterial foraging optimization [126][127] have been employed for channel equalization.

In order to overcome the shortcomings of gradient-decent based algorithms a number of population based metahheuristic algorithms have been employed for training the neural network based non-linear channel equalizers. In [18] , authors have used particle swarm optimization algorithm for optimizing architecture of neural networks, network parameters and transfer functions at various nodes and superior performance of the proposed equalizer is shown when compared to existing equalizer structures. Furthermore, a wavelet NN based equalizer trained using a symbiotic organism search algorithm has been introduced in [19] for robust non-linear channel equalization. The results confirmed that the proposed channel equalizer provides better performance than existing linear and non-linear equalizers. Furthermore, robustness of the equalizer is demonstrated by considering the burst error scenario in the occurring in the wireless environment. Lately, directed search optimization [17], Differential evolution (DE) algorithm [128], particle swarm optimization algorithm [36] and Shuffled frog leaping algorithm [115] have been employed for training a NN based channel equalizer. These studies reveal the superior performance of metaheuristic algorithms over gradient-descent based algorithms.

2.6 Conclusion

Literature survey shows that non-linear channel equalization is a well-researched area in wireless communication system. However, it still requires further investigation to improve equalization performance in terms of bit error rate (BER) by improving the efficiency of the training algorithms used for channel equalizers.

Chapter 3

A New Training Scheme for Neural Network based Non-linear Channel Equalizers using Cuckoo Search Algorithm

3.1 Introduction

Channel equalization has seen never ending drift of research in past few years. However, widespread use of the internet has resulted in a massive rise in the data rate of a wireless communication system. Eventually, to mitigate the effect of inter-symbol interference (ISI) in multipath wireless channels, designing a channel equalizer is becoming more demanding. For severe non-linear distortion, the neural network (NN) based non-linear channel equalizers provide superior performance than the adaptive filter based linear equalizers. To overcome the limitations of existing algorithms, this chapter proposes a training scheme using Cuckoo Search Algorithm (CSA) for functional link artificial NN (FLANN) based channel equalizers. The proposed training scheme has a better ability to escape from local minima, higher exploitation and exploration capabilities. To choose the optimum values of the parameters, the sensitivity analysis of the CSA based approach is performed with its key parameters. Furthermore, three non-linear channels have been simulated to demonstrate the equalization performance of the CSA based training scheme and the results have been compared with recent and well-established algorithms. The simulations confirm that the proposed training scheme performs substantially better than existing metaheuristic algorithms in terms of BER and MSE performance. To show the robustness of the CSA based method, the burst error scenario has been considered and results proved that the method is more successful in handling such scenarios when compared to other methods. The performance of the proposed scheme has been validated for a wide range of signal-to-noise ratio through simulation studies and it is observed that the scheme outperforms the other algorithms in poor SNR conditions as well. Also, to examine the statistical significance of the results provided by the proposed scheme, the Wilcoxon test is performed and the test reveals that the obtained results are statistically significant.

3.2 Cuckoo Search Algorithm (CSA)

Cuckoo search algorithm (CSA) is one among the recent population based metaheuristic algorithms, which is proposed by Yang and Deb [57] . CSA has been inspired from the concept of brood parasitism of particular species of cuckoo [57][58][59]. Along with brood parasitism, this algorithm also involves Lévy flight movements observed in certain birds[57][58][60][59] . Lévy flight based jumps help CSA to avoid being trapped into local minima and to find the potential regions of solution space. The superior performance of CSA in terms of local optima avoidance capability has been shown by a number of studies when compared to other leading metaheuristics such as PSO, genetic algorithm, artificial bee colony algorithm and DE algorithm [61] [57]. CSA has been effectively used by the research community in various research fields due to its enhanced exploration and exploitation capabilities [58]. In the last few years, CSA has been widely used for multilevel thresholding in the image [62] , image enhancement [63], design of fractional order differentiator and FOPID controller [64][65] , spectrum allocation in a vehicular network [46], non-convex economic dispatch problem [66] and optimization of traffic signal controller [67]. Recently, CSA has been employed for system identification using the Hammerstein model and for feedback system identification [68][40] and superior performance of CSA has been reported over state-of-the-art methods.

The enhanced performance of the CSA in these studies in terms of local optima avoidance, higher exploration and exploitation capabilities and local optima stagnation problem of the existing algorithms motivated us to use CSA as a new training scheme for non-linear channel equalization. Thus, this chapter proposes a training scheme for Functional link artificial neural network based non-linear channel equalizer using a cuckoo search algorithm. The performance of the CSA based approach has been compared with other well-established population based metaheuristic algorithms like PSO [129][130] and DE [131] and some latest methods such as sine cosine algorithm (SCA) [132], grey wolf optimizer (GWO) [133], dragonfly algorithm (DA) [134] and whale optimization algorithm (WOA) [135] .

In the cuckoo search algorithm, a random walk is offered by Lévy flight and the steps of the random walk follow a Lévy distribution. The steps drawn from Lévy flight has infinite mean and variance [57].

$$\text{Lévy} \sim u = t^{-1-\beta}, \; 0 < \beta < 2 \tag{3.1}$$

where β is an index of Lévy distribution

Following key rules have been used to explain the basic concept of the Cuckoo search algorithm in a simple manner [57][136]

i. The total nests available are fixed.

ii. At each point of time, every cuckoo bird lays only one egg and consequently, it drops the egg in a nest of another host bird in a random manner.

iii. The subsequent generations will have the quality eggs from the best nests carried forward from the earlier generation.

iv. In CSA the parameter p_a indicates the probability with which the host will discover the egg of cuckoo . If the cuckoo's egg is noticed by a host bird then it can leave the present nest and construct another nest or damage the cuckoo's egg.

The key steps of the Cuckoo Search Algorithm (CSA) can be explained as follows:

1. Initialize *N* host nests X_i $(i = 1, 2...N)$ randomly in the lower and upper limits of the solution space. For every i^{th} nest, its j^{th} *(j=1,2....D)* dimension i.e. $x_{i,j}$ is initialized in the range

   $[x_{\min,j} \quad x_{\max,j}]$ as given below:

$$x_{i,j} = x_{\min,j} + random(0,1) * (x_{\max,j} - x_{\min,j}) \tag{3.2}$$

   where $x_{\min,j}$ and $x_{\max,j}$ denotes a lower and upper limit of j^{th} dimension and $random(0,1)$ indicates a random number between zero and one which follows the uniform distribution,

2. Evaluate the fitness F_i of every nest and find the best nest i.e. X_{best} .

3. To generate the new nests, perform the Lévy flight for all nests by using the best nest obtained in step (2) as follows [57]:

$$X_i^{new} = X_i^{old} + \alpha \oplus Lévy(\beta), \qquad 0 < \beta < 2 \tag{3.3}$$

where β is an index of Lévy distribution, α represents the step size ($\alpha > 0$) and the operator $\oplus$ denotes the entry wise multiplication.

4. Perform the fitness function calculation F_i^{new} for each new nest obtained from Lévy flight in Step 3 and carry out the comparison between the resultant values of fitness function and the fitness of the old nests F_i^{old}.

5. If the problem is of minimization and $F_i^{new} < F_i^{old}$ then accept the new nest by replacing X_i^{old} with X_i^{new} else keep the previous nest i.e. X_i^{old}.

6. Use random flight to replace the fraction p_a of worst nests found in the step (5) and evaluate the fitness of newly generated nests.

7. Retain the better nests among the worst nests and new nests generated with random flight by performing the fitness comparison.

8. Update the best nest X_{best} by ranking all the nests as per their fitness.

   Report the optimal solution X_{best} if the termination condition is reached else repeat the steps 3-8.

3.3 Cuckoo Search Algorithm based training scheme for FLANN non-linear channel equalizer

This section describes the proposed approach based on the Cuckoo Search Algorithm (CSA) for training the Functional link artificial neural network non-linear channel equalizer**.** The flowchart of the cuckoo search algorithm based approach for channel equalization is depicted in Fig. 3.2 and the equalizer structure is illustrated in Fig. 3.1. The fitness function considered to evaluate the quality of individual nest is mean squared error (MSE). The main aim in non-linear channel equalization is minimizing the value of the fitness function (MSE) over the course of iterations. Moreover, the following interpretations are helpful to understand how the Cuckoo search algorithm based training scheme is capable of enhancing the performance of Functional link artificial neural network based non-linear channel equalizer,

- CSA makes use of Levy flight (LF) based search for exploration of search space instead of standard random walks [58][59]. CSA performs the more efficient exploration of solution space when compared to other algorithms with Gaussian process owing to infinite variance and mean of Levy flight and discovers the promising areas of the solution space [58][59].

- Higher exploration capability of the CSA assists to escape from local minima stagnation and avoids the convergence of the algorithm to the local optimal solution.

- CSA maintains the proper balance between global and local search capabilities during the entire search process with the help of discovery probability (p_a) [58][59]. Hence, the fine balance assists CSA to explore the solution space globally with more efficacy and subsequently, it increases the probability to obtain the global optimal solution [58] [59].

- The average fitness of all the nests (MSE) improves over the course of iterations and consequently, it enriches the initial random nests and guarantees the convergence of the CSA based training scheme to the optimal solution.

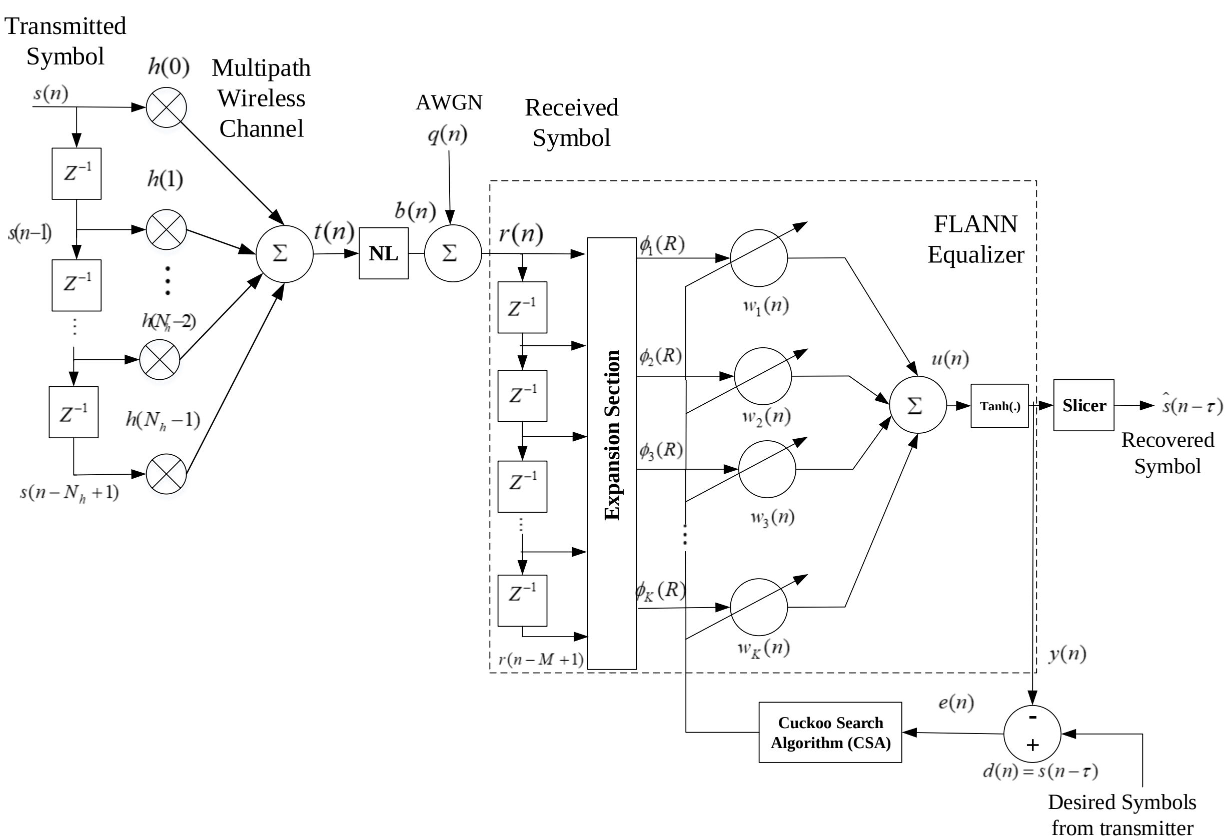


Fig. 3. 1 Block diagram of FLANN Equalizer using Cuckoo Search Algorithm based training scheme

The steps used in the cuckoo search algorithm based scheme to train the FLANN non-linear equalizer are explained as follows:

1. **Calculate the wireless channel output**: The transmitted sequence $s(n)$ contains random symbols which take values either -1 or +1. To calculate the corrupted channel output $r(n)$ as provided in Eq. (2.3), the sequence $s(n)$ is transmitted through a linear dispersive channel and degraded by noise $q(n)$ along with non-linear distortion from nonlinearity $\psi(.)$.

2**. Compute input and expanded vector of the equalizer**: The input vector $R(n)$ to FLANN equalizer is obtained by passing the received corrupted symbols $r(n)$ through the tap delay structure and *R(n)* is further expanded with trigonometric polynomials to get $R^e(n)$ as per Eq. (2.8).

3. **Initialize the population of *N* Nests**: Keeping each dimension of a nest in the lower bound ($x_{\min}$) and upper bound ($x_{\max}$) of the solution space, the population of *N* nests is initialized as follows:

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,D} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,D} \\ \vdots & \vdots & \ddots & \cdots \\ x_{N,1} & x_{N,2} & \cdots & x_{N,D} \end{bmatrix} \tag{3.4}$$

In Eq. (3.4), X_i denotes i^{th} nest and $x_{i,j}$ represents the j^{th} dimension of i^{th} nest

4. **Compute actual output of FLANN and MSE:** The FLANN output $y(n)$ is evaluated as per Eq. (2.12), which is used to compute the error by comparing it with the desired signal $d(n)$. The calculated error is used to evaluate the fitness of nests as follows:

$$MSE = \frac{1}{S}\sum_{n=1}^{S} e^2(n) \tag{3.5}$$

In Eq. (3.5), $e(n)$ denotes the error for the n^{th} symbol *and S* represents the block size or number of symbols transmitted.

5. **Generate new nests using Lévy flight**: After fitness evaluation, find the best nest i.e. X_{best} having minimum MSE and use the best nest (X_{best}) to generate new nests with Lévy flight as per Eq. (3.3).

6. **Calculate the fitness of new nests:** Calculate the fitness F_i^{new} of every new nest generated by Lévy flight and perform the comparison between the fitness of the old nests F_i^{old} and new nests F_i^{new}. If F_i^{new} is better than the old fitness F_i^{old} (for the channel equalization if $F_i^{new} <$ F_i^{old}) then accept a new nest by replacing X_i^{old} with X_i^{new}, else keep the previous nest i.e. X_i^{old}.

7. **Perform random flight on worst nests:** To replace a fraction p_a of worst nests found in step (6) perform the random flight and evaluate the fitness of a newly generated nest. Rank all the nests according to fitness and update the X_{best}.

8**. Termination condition:** Report the optimal solution X_{best} if the termination condition is reached else repeat the steps 4-7.

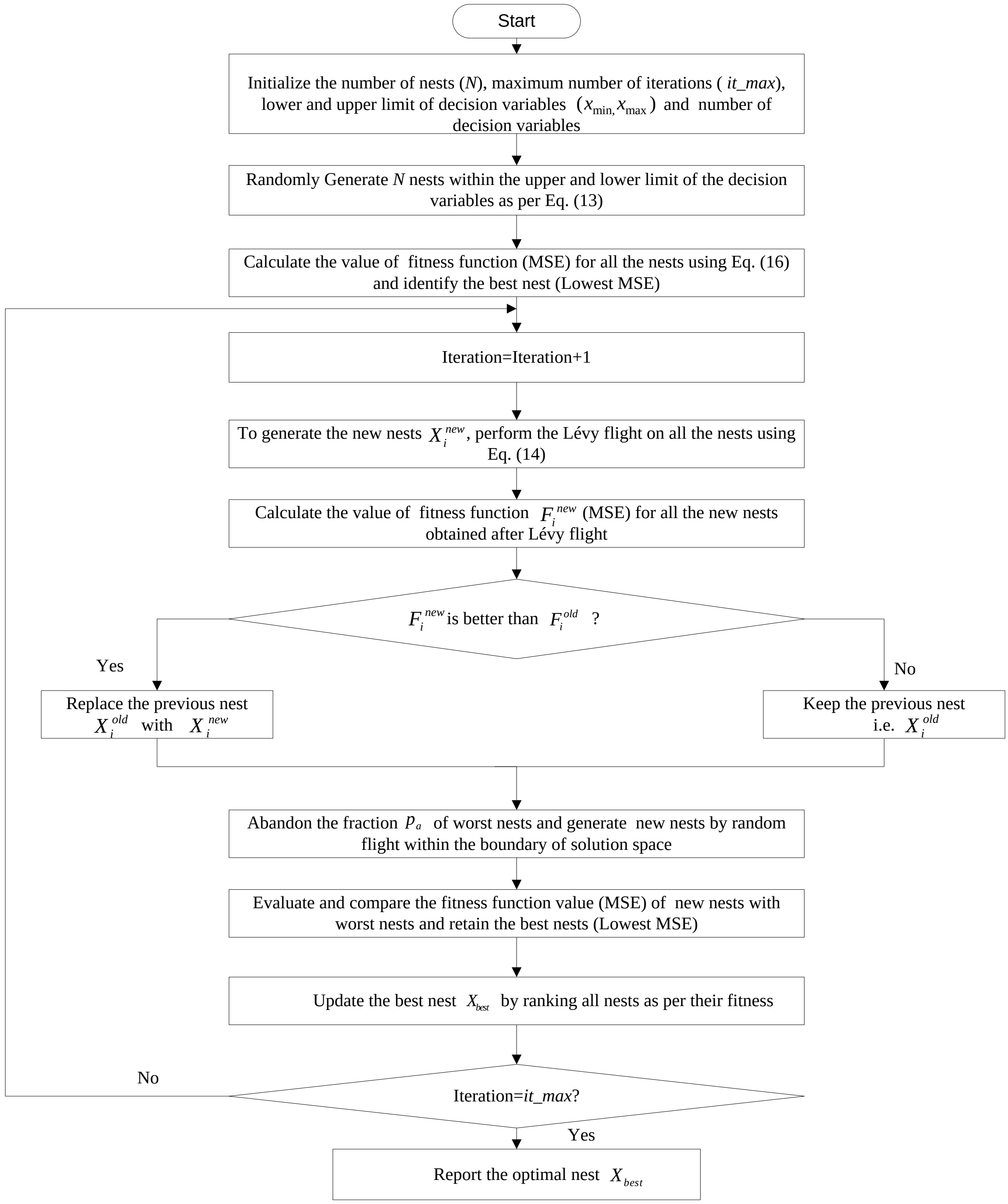


Fig. 3. 2. Flowchart of the Cuckoo Search Algorithm (CSA) for Non-linear Channel equalization

3.4 Simulation studies

The simulation experiments have been conducted on a personal computer with 8 GB of RAM and an Intel i5-4590S 3 GHz CPU. The cuckoo search algorithm based training scheme and other compared algorithms have been implemented in a MATLAB R2013a environment.

3.4.1 Channel Models considered for simulations

In this section, simulations have been performed to examine the channel equalization performance of the CSA based training scheme and other compared methods. The FLANN based channel equalizer is trained by CSA [57] , PSO [129][130], GWO [133] , DE [131] , SCA [132] , WOA [135] and DA [134] algorithm. Three wireless communication channels considered for simulations in this study are taken from [11] [1] [37][17] [108][19][137][138] and the corresponding channel transfer functions are given as follows:

$$
\begin{aligned}
H_1(z) &= 0.26 + 0.93z^{-1} + 0.26z^{-2} && : \text{Channel 1} \\
H_2(z) &= 0.304 + 0.903z^{-1} + 0.304z^{-2} && : \text{Channel 2} \\
H_3(z) &= 0.341 + 0.876z^{-1} + 0.341z^{-2} && : \text{Channel 3}
\end{aligned}
\tag{3.6}
$$

In Eq. (3.6), the wireless channels i.e. Channel 1, 2, 3 have the eigen value ratio (EVR) of 11.12, 21.71 and 46.82 respectively [1]. Therefore, Channel 3 has a highest EVR and is a highly dispersive channel. The nonlinearities taken for simulations in this study are as follows [1][37][19][137]:

$$
\begin{aligned}
b(n) &= t(n) && : \text{NL}=0 \\
b(n) &= \tanh(t(n)) && : \text{NL}=1 \\
b(n) &= t(n) + 0.2t^2(n) - 0.1t^3(n) && : \text{NL}=2 \\
b(n) &= t(n) + 0.2t^2(n) - 0.1t^3(n) + 0.5\cos(\pi t(n)) && : \text{NL}=3
\end{aligned}
\tag{3.7}
$$

where $t(n)$ refers to the wireless channel output and $b(n)$ denotes the channel output after introducing the non-linear distortion.

Among the above mentioned nonlinearities, a nonlinearity NL=1 characterizes the non-linear distortion arising as a result of the amplifier saturation. A linear channel model with no non-linear distortion is denoted by NL=0. Furthermore, NL =2 and NL =3 are two arbitrary nonlinearities out of which NL=3 represents a case of the severe non-linear distortion [1][37][32][19][137]. The description of parameters used in simulations is given in Table 3.1.

Table 3. 1 Description of parameters

Symbol	Purpose of the symbol
N	Population Size (Number of nests)
M	No. of taps of the equalizer
S	Block Size
p_a	Discovery probability
β	Index of Lévy Distribution
N_h	Length of the channel impulse response
it_max	Maximum number of iterations

3.4.2 Sensitivity analysis of the Cuckoo Search Algorithm (CSA) based equalizer training scheme

In population-based metaheuristic algorithms, the tuning of control parameters significantly affects the performance of the algorithm [108]. Therefore, sensitivity analysis of CSA is conducted with respect to five parameters such as a number of host nest (N), data block size (S), discovery/switching probability (p_a), index of Lévy distribution (β) and the number of taps (M). Thus, Figs. 3.3 to 3.7 illustrate the results of the sensitivity analysis for Channel 1 (Eq. 3.6) with NL=1 (Eq. (3.7)).

A) Number of nests (or Population size *N*)

Generally, fast convergence and global search capability can be achieved with an increase in the number of nests. It can be seen from Fig. 3.3 that with an increase in the number of nests (N), MSE decreases significantly. However, it is also observed from this figure that N=20 number of nests is enough to escape from the local optima, which can be confirmed from MSE results of CSA in Table 3.2. It can be noticed that increase in the value of N further does not result in any significant MSE reduction.

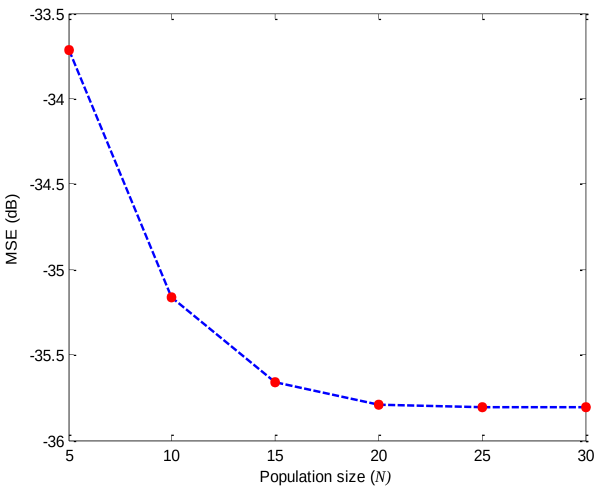


Fig. 3. 3**.** Effect of Population size (*N*)

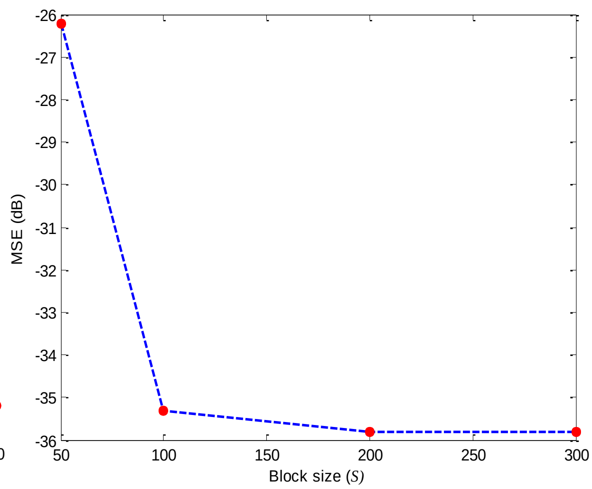


Fig. 3. 4**.** Effect of Block Size (*S*)

Table 3. 2 Effect of variation of *N* on MSE

Number of nests (*N*)	MSE				
	Min	Max	Median	Mean	SD
5	3.2598e-04	6.4256e-04	4.1365e-04	4.2493e-04	9.4951e-05
10	2.7415e-04	3.5316e-04	3.0151e-04	3.0458e-04	2.2527e-05
15	2.6121e-04	2.8761e-04	2.6859e-04	2.7164e-04	9.0161e-06
20	2.6038e-04	2.7147e-04	2.6272e-04	**2.6368e-04**	**3.1024e-06**
25	2.6110e-04	2.6521e-04	2.6271e-04	2.6268e-04	1.1318e-06
30	2.6037e-04	2.6583e-04	2.6261e-04	2.6259e-04	1.6224e-06

B) Block Size (S)

Usually, the increase in block size (*S*) results in the reduction of MSE owing to increased error estimates for every nest. Increasing the value of *S* from 50 to 200 leads to better MSE performance which can be seen from Table 3.3 and Fig. 3.4. However, by selecting a block size more than S=200 for the channel $H_1(z)$ with non-linearity NL=1 (Eqs. (3.6-3.7)) hardly improves the MSE performance. Thus, the data block size is considered as 200 for simulations in this study.

C) Discovery/switching probability (p_a)

The Discovery or switching probability p_a controls the global and local search capabilities of CSA[58][59]. As seen from Fig. 3.5, the probability p_a of 0.1 gives the minimum MSE for non-linear channel equalization. The best values obtained for minimum, maximum, average and standard deviation of MSE are 2.5979e-04, 2.6261e-04, 2.6078e-04 and 1.0147e-06 respectively, which are obtained at the probability of 0.1 as shown in Table 3.4. Furthermore, Fig. 3.5 and the statistical comparison of MSE in Table 3.4 shows that for this problem, p_a of 0.1 only leads to an optimal solution.

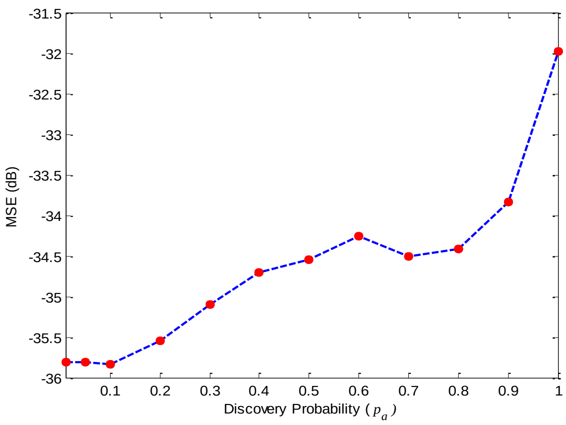


Fig. 3. 5 Effect of Discovery Probability p_a

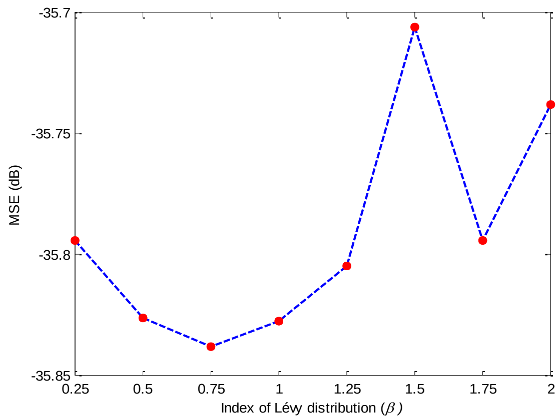


Fig. 3. 6. Effect of the index of Lévy distribution (β)

Table 3. 3 Effect of the block size (S) on MSE

Block Size (S)	MSE				
	Min	Max	Median	Mean	SD
50	2.3761e-03	2.4055e-03	2.3864e-03	2.3882e-03	9.1651e-06
100	2.9044e-04	2.9929e-04	2.9441e-04	2.9469e-04	3.1306e-06
200	2.6084e-04	2.6467e-04	2.6245e-04	**2.6249e-04**	**1.1007e-06**
300	2.6001e-04	2.6879e-04	2.6202e-04	2.6245e-04	2.5224 e-06

Table 3. 4 Effect of variation of discovery probability p_a on MSE

Discovery probability (p_a)	MSE				
	Min	Max	Median	Mean	SD
0.01	2.6014e-04	2.6444e-04	2.6245e-04	2.6250e-04	1.5306e-06
0.05	2.5977e-04	2.7693e-04	2.6027e-04	2.6240e-04	5.2541e-06
0.1	2.5979e-04	2.6261e-04	2.6030e-04	**2.6078e-04**	**1.0147e-06**
0.2	2.6648e-04	2.9633e-04	2.7827e-04	2.7845e-04	8.4185e-06
0.3	2.8115e-04	3.3685e-04	3.1027e-04	3.0881e-04	1.6003e-05
0.4	2.9591e-04	3.6952e-04	3.3550e-04	3.3875e-04	2.6693e-05
0.5	3.0656e-04	3.8943e-04	3.5254e-04	3.5098e-04	2.6189e-05
0.6	2.9033e-04	4.2596e-04	3.8872e-04	3.7509e-04	4.1974e-05
0.7	2.9005e-04	4.3419e-04	3.4957e-04	3.5378e-04	4.4003e-05
0.8	3.0805e-04	4.3134e-04	3.5289e-04	3.6161e-04	3.7363e-05
0.9	3.5520e-04	4.9081e-04	4.1559e-04	4.1376e-04	4.0016e-05
1	3.3815e-04	9.2405e-04	6.3173e-04	6.3438e-04	1.9652e-04

D) Index of Lévy distribution (β)

The value of the Lévy index (β) is varied from 0.25 to 2 with a step of 0.25. The effect of variation of values of β is illustrated in Fig. 3.6 and Table 3.5. The best values obtained for minimum, maximum, average and standard deviation of MSE are 2.5988e-04, 2.6217e-04, 2.6073e-04 and 6.4060e-07 respectively, which are obtained at a value of β equal to 0.75 as reported in Table 3.5. Therefore, the value of β is fixed at 0.75 for simulations in this paper.

Table 3. 5 Effect of variation of index of Lévy distribution (β) on MSE

Index of Lévy distribution (β)	MSE				
	Min	Max	Median	Mean	SD
0.25	2.6068e-04	2.6668e-04	2.6310e-04	2.6339e-04	2.0829e-06
0.5	2.6040e-04	2.6502e-04	2.6088e-04	2.6146e-04	1.5204e-06
0.75	2.5988e-04	2.6217e-04	**2.6067e-04**	**2.6073e-04**	**6.4060e-07**
1	2.5995e-04	2.6322e-04	2.6085e-04	2.6136e-04	1.1719e-06
1.25	2.5997e-04	2.6604e-04	2.6241e-04	2.6274e-04	2.3268e-06
1.5	2.5993e-04	3.0873e-04	2.6250e-04	2.6878e-04	1.4929e-05
1.75	2.6050e-04	2.7183e-04	2.6248e-04	2.6339e-04	3.2805e-06
2	2.6191e-04	2.7503e-04	2.6626e-04	2.6681e-04	4.3568e-06

E) Number of taps (*M*)

The results in Fig. 3.7 indicate the effect of variation of a number of taps (*M*) on MSE performance. Usually, an increased number of equalizer taps leads to a reduction in the MSE. It is observed from this figure that four tap equalizer provides the optimum performance. But, Fig. 3.7 and Table 3.6 shows that the value of *M* more than four does not result in any significant reduction of MSE. Therefore, the number of taps considered for simulations in this paper is four.

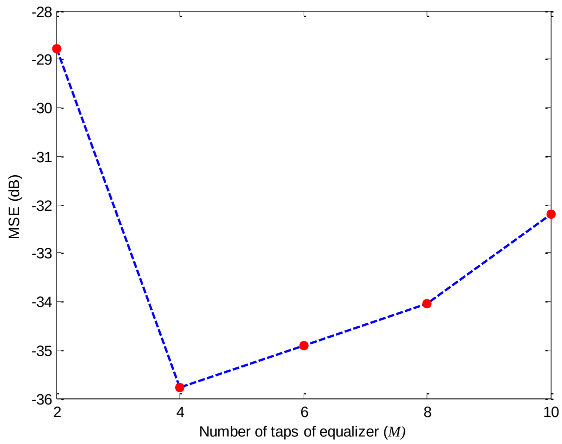


Fig. 3. 7. Effect of Number of taps on MSE (M)

Table 3. 6 Effect of variation of M on MSE

Number of taps (*M*)	MSE				
	Min	Max	Median	Mean	SD
2	0.0013	0.0013	0.0013	0.0013	4.7550e-16
4	2.6302e-04	2.6711e-04	2.6397e-04	**2.6440e-04**	**1.5502e-06**
6	2.7553e-04	3.6014e-04	3.3021e-04	3.2323e-04	2.8749e-05
8	3.2291e-04	4.5946e-04	3.9436e-04	3.9375e-04	4.8087e-05
10	3.8320e-04	8.3285e-04	5.8837e-04	6.0405e-04	1.3135e-04

3.4.3 Performance Analysis of the CSA based Training Scheme for Non-linear Channel Equalization

Each transmitted symbol is a real-valued and takes values in the form of {-1,+1} following the uniform distribution. The noise added to the wireless communication channel output after the introduction of non-linear distortion is AWGN of signal-to-noise ratio (SNR) 10 dB, 20 dB and 30 dB. The block size of 200 symbols is taken as the input in the training process of FLANN. The input symbols are delayed by two units to generate the desired signal which is utilized in error computation during the training phase and for BER calculation during the testing phase. The expansion block of FLANN takes four inputs from the tapped delay segment and generates thirteen terms along with bias by expanding every input into 3.

The CSA based training method has been used to train FLANN equalizer for 500 iterations over thirty independent runs. The min, max, standard deviation (SD) and mean values of MSE are reported in results. Mean refers to MSE averaged over thirty runs and thus the capability of effectively escaping from the local minima and converging to an optimal solution is represented by the lesser mean. Furthermore, the standard deviation is also considered to find the results dispersion. Min and max denote the minimum and maximum value of MSE in thirty runs. The results distribution achieved by the algorithms over thirty runs is demonstrated with box plots.

The channel equalization capability of the cuckoo search algorithm based training scheme for FLANN has been compared with PSO [129][130], GWO [133], DE [131] , SCA [132], WOA [135] and DA [134] algorithm. Exhaustive simulation experiments are performed to choose the key parameters of all the compared algorithms for a fair comparison among the CSA based equalizer training scheme and other algorithms. The parameters which provided the optimum results for all experiments are chosen for the compared algorithms. Table 3.7 presents the parameters taken for all the metaheuristic algorithms for comparative study. In Table 3.7, the *rand* represents a random number between 0 and 1.

Table 3. 7 Parameters used in the simulation

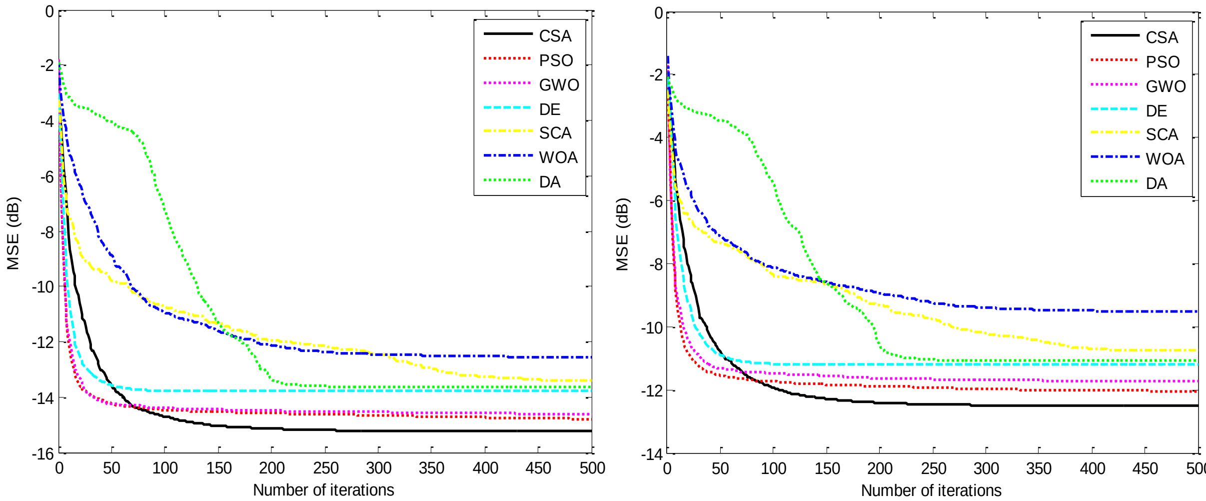
Algorithm	Parameter	Value
CSA	Population size *(N)*	20
	Discovery probability (p_a)	0.1
	Index of Lévy distribution (β)	0.75
PSO	Population size *(N)*	20
	C1	2
	C2	2
	Inertia weight (*w*)	0.9
GWO	Population size *(N)*	20
	r_1	*rand*
	r_2	*rand*
	Convergence constant *a*	[2 0]
DE	Population size *(N)*	20
	Mutation Factor (*F*)	0.4
	Crossover rate (*CR*)	0.9
SCA	Population size *(N)*	20
	r_1, r_4	*rand*
	r_2	2**pi*rand*
	r_3	2**rand*
	a	2
WOA	Population size *(N)*	20
DA	Population size *(N)*	20

3.4.3.1 MSE performance

The training of FLANN is performed for 500 iterations to examine the convergence performance of the cuckoo search algorithm based training scheme and other algorithms and the MSE is averaged over thirty runs. The MSE convergence for three non-linear channels is presented in this section for a signal-to-noise ratio of 10 dB, 20 dB and 30 dB.

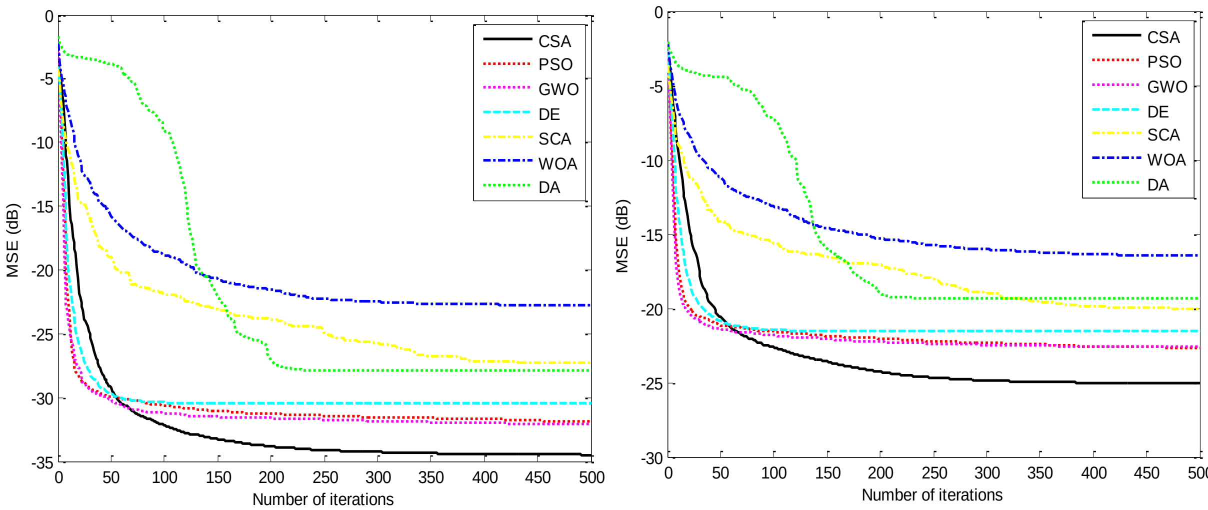
Case A: Channel 1

In this case, Channel 1 (as illustrated in Eq. 3.6) which corresponds to EVR of 11.12 has been taken into consideration for evaluation of MSE performance of CSA based training strategy with two different nonlinearities for signal-to-noise ratio conditions of 10-30 dB.



(a) (b)

Fig. 3. 8. Convergence curves for Channel 1 with nonlinearities (a) NL=2 and (b) NL=3 at SNR=10 dB



(a) (b)

Fig. 3. 9. Convergence curves for Channel 1 with nonlinearities (a) NL=2 and (b) NL=3 at SNR=20 dB

The convergence performance comparison of the cuckoo search algorithm based training scheme with the other algorithms for channel 1 with nonlinearities NL=2 and NL=3 at SNR of 10 dB is illustrated in Fig. 3.8 (a) and (b) respectively. It can be seen from this figure that even the signal-to-noise conditions are poor CSA attains the least MSE amongst all the algorithms. Thus, the MSE performance of CSA is superior in comparison with PSO, GWO, DE, SCA, WOA and DA algorithm. When the nonlinearity increased from NL=2 to NL=3, the MSE performance showed degradation for all the methods whereas the CSA was found to behave persistently despite severe non-linearity and poor SNR scenario.

Furthermore, from Figs. 3.8, 3.9 and 3.10 it is observed that the CSA based training technique provides the better performance among all the algorithms for a wide range of signal-to-noise ratio (10-30dB) conditions.

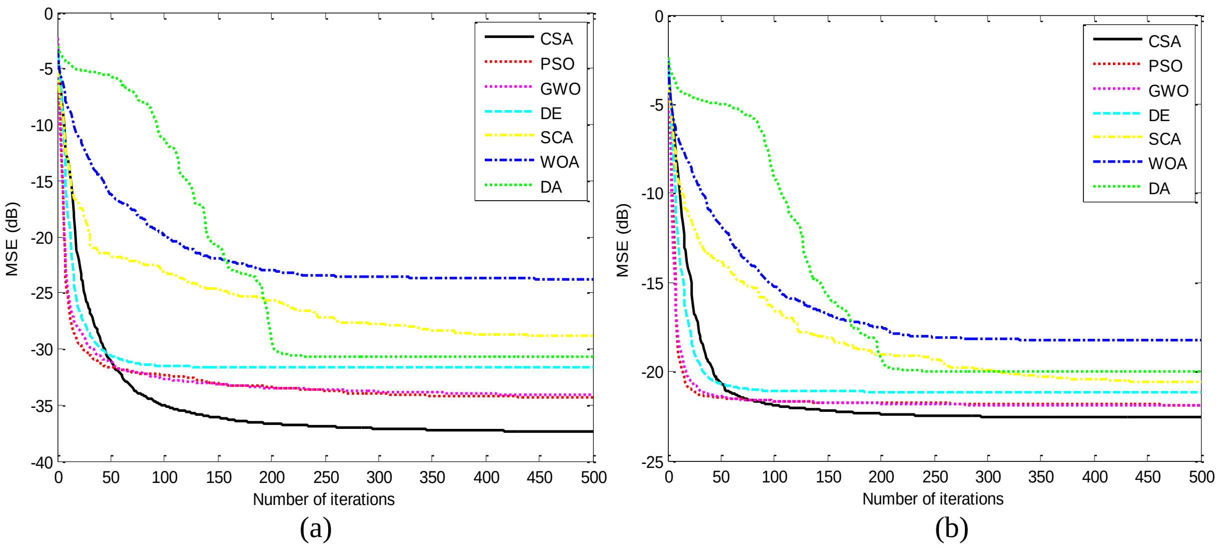
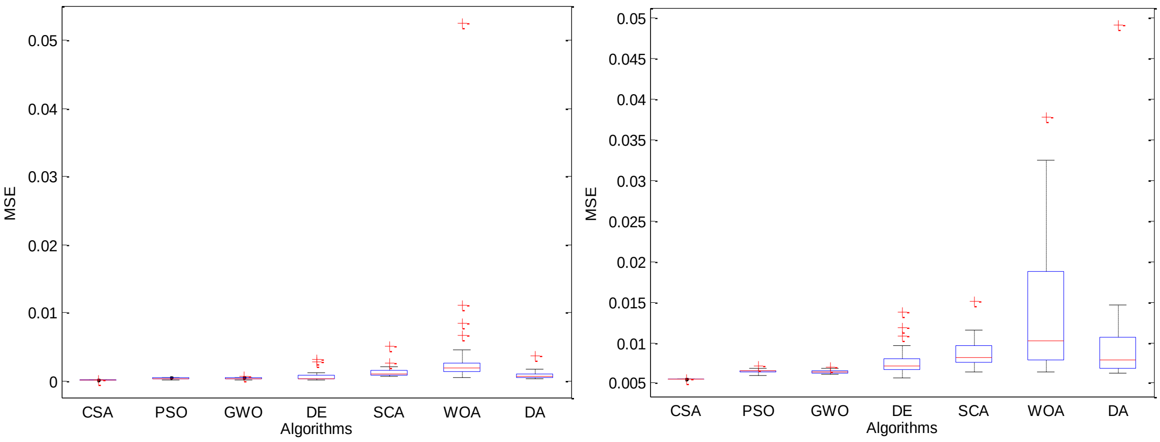


Fig. 3. 10**.** Convergence curves for Channel 1 with nonlinearities (a) NL=2 and (b) NL=3 at SNR=30 dB

The MSE results for signal-to-noise ratio (SNR) of 30 dB in Table 3.8 also confirm the superiority of the cuckoo search algorithm based training scheme. Furthermore, the lowest values of the standard deviation and mean of MSE from this table proves the enhanced equalization competence of CSA in terms of local minima avoidance capability.

MSE Box plots at SNR of 30 dB for Channel 1 by considering nonlinearities NL=2 and NL=3 are shown in Fig. 3.11 (a) and (b) respectively. Generally, Box plots are used to illustrate and analyze the distribution of results [139]. Fig**.** 3.11 shows that the interquartile range and median of the MSE results provided by CSA are least among all the algorithms. Thus, these box plots validate the superiority of the proposed training scheme.



(a) (b)

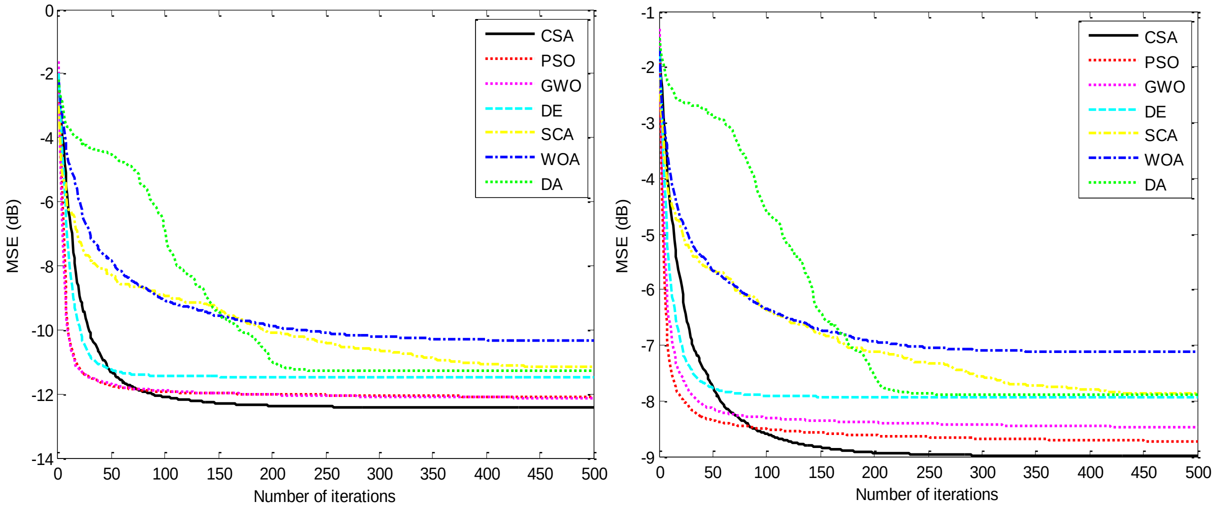
Fig. 3. 11. MSE Box plots of all algorithms for Channel 1 with nonlinearities (a) NL=2 and (b) NL=3 at SNR=30 dB

Table 3. 8 MSE results for Channel 1 with nonlinearities NL=2 and NL=3 at SNR=30 dB

Non-Linearity	Algorithm	MSE				
		Min	Max	Median	Mean	SD
NL=2	CSA	1.7990e-04	1.9051e-04	1.8185e-04	**1.8223e-04**	**2.4112e-06**
	PSO	2.0952e-04	5.2596e-04	3.7860e-04	3.6555e-04	8.0511e-05
	GWO	2.1916e-04	7.1988e-04	3.8761e-04	3.8507e-04	1.2840e-04
	DE	2.0045e-04	0.0032	3.6078e-04	6.8391e-04	6.8681e-04
	SCA	6.0591e-04	0.0051	0.0011	0.0013	8.5292e-04
	WOA	4.2355e-04	0.0525	0.0020	0.0042	0.0094
	DA	3.0776e-04	0.0037	6.8683e-04	8.4127e-04	6.6028e-04
NL=3	CSA	0.0055	0.0055	0.0055	**0.0055**	**1.3121e-06**
	PSO	0.0059	0.0072	0.0065	0.0065	2.6051e-04
	GWO	0.0060	0.0070	0.0064	0.0064	2.2487e-04
	DE	0.0056	0.0137	0.0072	0.0077	0.0018
	SCA	0.0063	0.0151	0.0082	0.0087	0.0018
	WOA	0.0064	0.0379	0.0102	0.0149	0.0092
	DA	0.0063	0.0491	0.0078	0.0100	0.0078

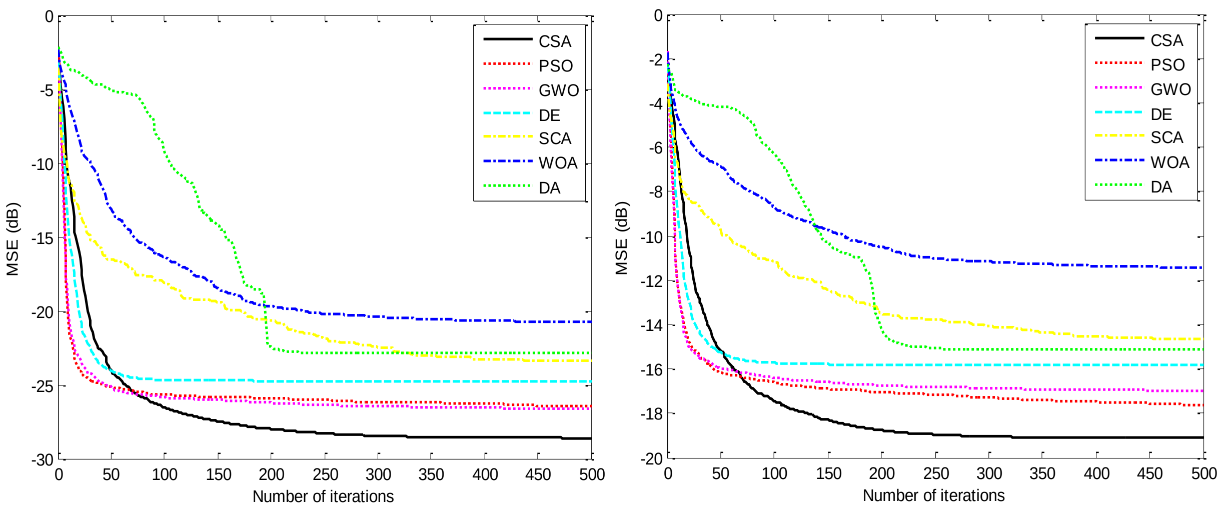
Case B: Channel 2

Channel 2 corresponding to EVR of 21.71 is considered to demonstrate the equalization capability of CSA based training scheme. Upon comparison with channel 1, this channel is more dispersive. Similar to the case of channel 1, for this channel also MSE performance is evaluated with two different nonlinearities i.e. NL=2 and NL=3 for a wide range of SNR.



(a) (b)

Fig. 3. 12. Convergence curves for Channel 2 with nonlinearities (a) NL=2 and (b) NL=3 at SNR=10 dB

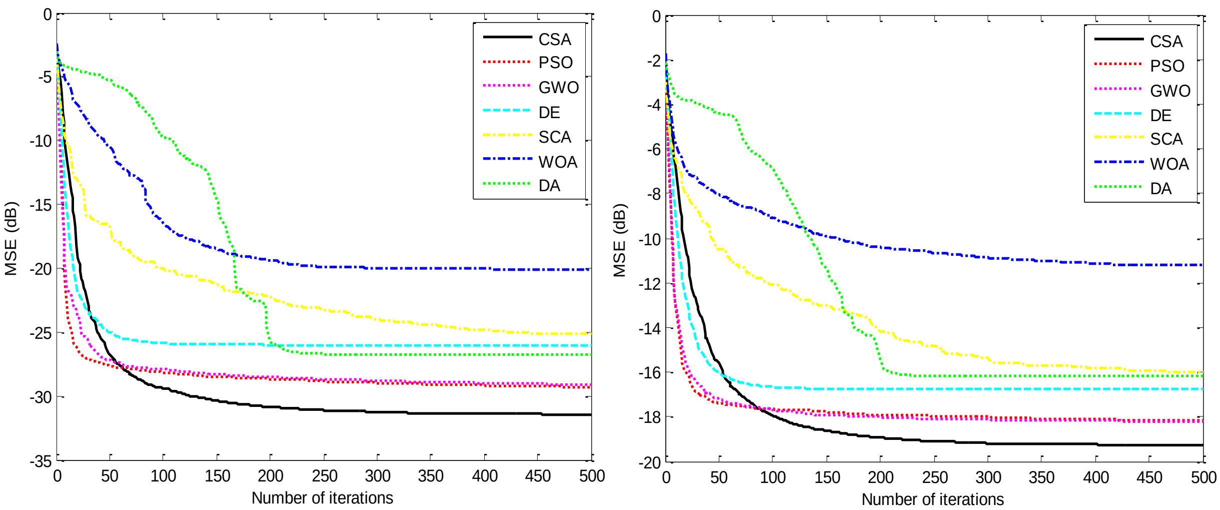


(a) (b)

Fig. 3. 13. Convergence curves for Channel 2 with nonlinearities (a) NL=2 and (b) NL=3 at SNR=20 dB

Figs. 3.12-3.14 shows convergence curves for channel 2 with nonlinearities NL=2 and NL=3 at SNR of 10, 20 and 30 dB respectively. It is observed from convergence curves that CSA has the best performance in terms of escaping from local minima and offers better exploration capability among the compared algorithms. Despite the worsening of the MSE performance with increased EVR from Channel 1 to Channel 2 (Figs 3.8 and 3.12), CSA has exhibited superior performance in comparison to other algorithms.

The improved efficacy of CSA for training the FLANN is clear from the MSE results at SNR of 30 dB in Table 3.9. The values of standard deviation and mean of MSE obtained from CSA in this table confirms the superiority of CSA over all the compared algorithms. The inter-quartile range and median of CSA from the MSE boxplots in Fig. 3.15 indicates the better equalization competence of CSA in comparison with all the other compared algorithms for non-linear channels.



(a) (b)

Fig. 3. 14. Convergence curves for Channel 2 with nonlinearities (a) NL=2 and (b) NL=3 at SNR=30 dB

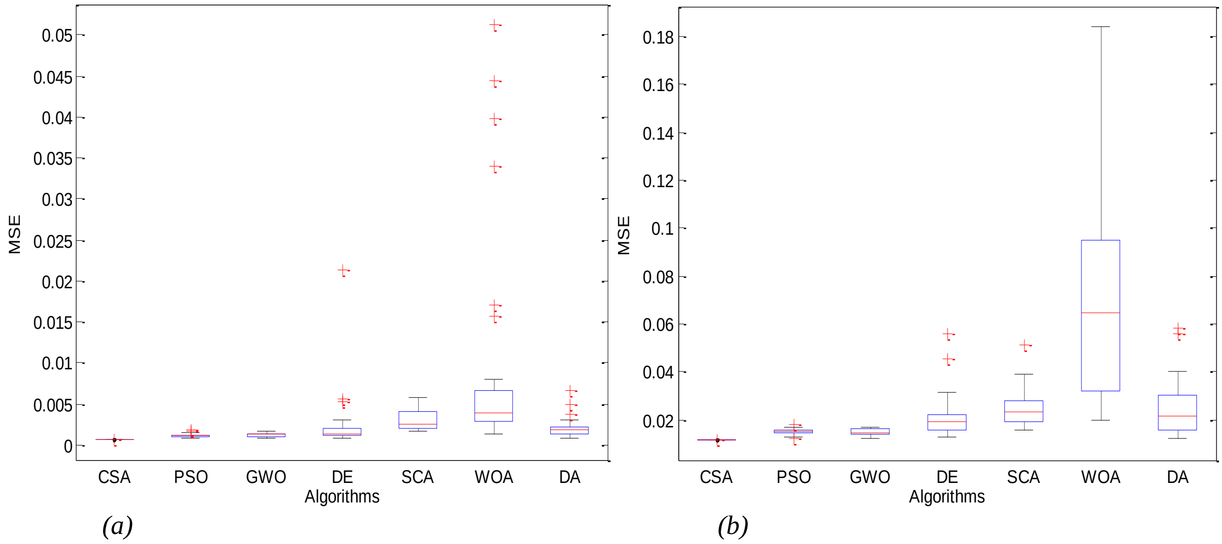


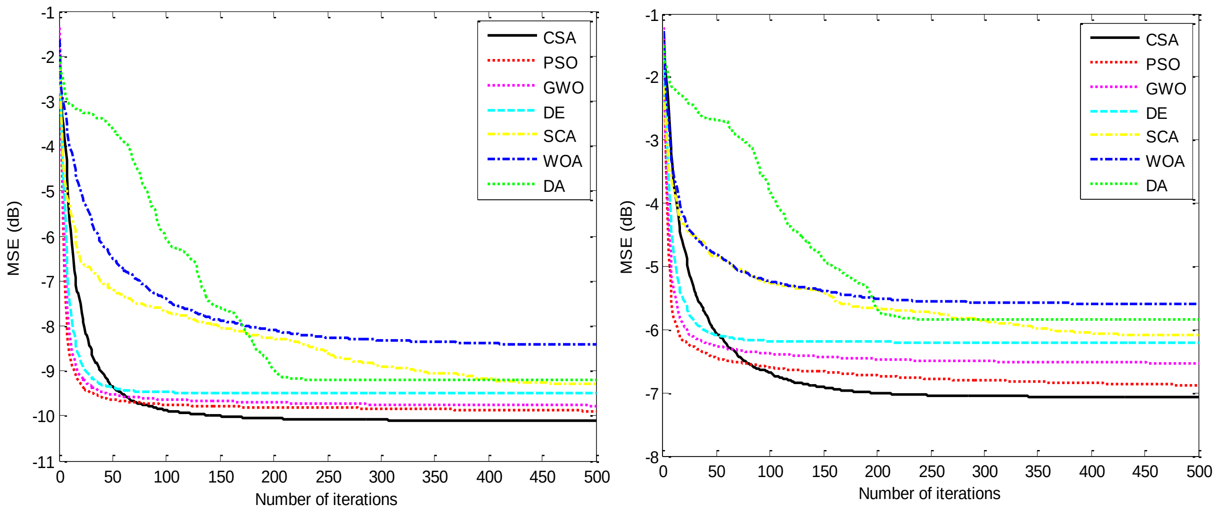
Fig. 3. 15. MSE Box plots of all algorithms for Channel 2 with nonlinearities (a) NL=2 and (b) NL=3 at SNR=30 dB

Table 3. 9 MSE results for Channel 2 with nonlinearities NL=2 and NL=3 at SNR=30 dB

Non-linearity	Algorithms	MSE				
		Min	Max	Median	Mean	SD
NL=2	CSA	7.1579e-04	7.3380e-04	7.1759e-04	**7.1884e-04**	**3.6926e-06**
	PSO	8.8650e-04	0.0018	0.0011	0.0012	2.1379e-04
	GWO	8.2525e-04	0.0018	0.0013	0.0012	2.4067e-04
	DE	8.8582e-04	0.0214	0.0014	0.0025	0.0037
	SCA	0.0017	0.0058	0.0026	0.0031	0.0012
	WOA	0.0014	0.0512	0.0040	0.0098	0.0137
	DA	7.7078e-04	0.0067	0.0018	0.0021	0.0012
NL=3	CSA	0.0119	0.0119	0.0119	**0.0119**	**6.2444e-06**
	PSO	0.0126	0.0180	0.0151	0.0152	0.0013
	GWO	0.0123	0.0172	0.0149	0.0151	0.0013
	DE	0.0131	0.0562	0.0190	0.0210	0.0093
	SCA	0.0159	0.0515	0.0235	0.0251	0.0080
	WOA	0.0201	0.1840	0.0650	0.0754	0.0501
	DA	0.0123	0.0581	0.0214	0.0241	0.0115

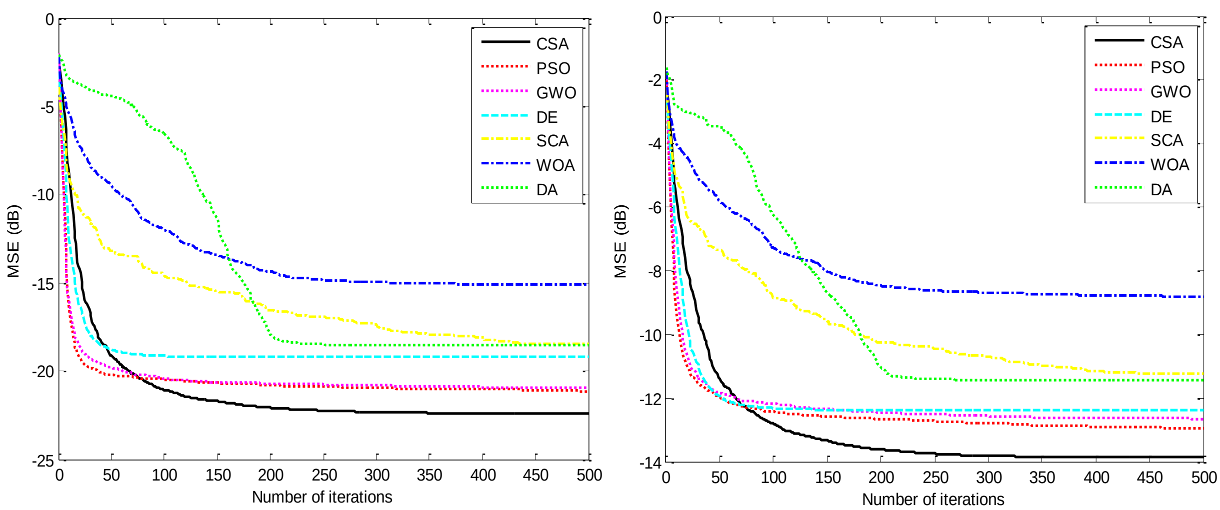
Case C: Channel 3

In an attempt to further validate the equalization performance of CSA, a highly dispersive channel i.e. Channel 3 (as depicted in Eq. (3.6)) corresponding to EVR of 46.82 is considered in this case.



(a) (b)

Fig. 3. 16**.** Convergence curves for Channel 3 with nonlinearities (a) NL=2 and (b) NL=3 at SNR=10 dB



(a) (b)

Fig. 3. 17. Convergence curves for Channel 3 with nonlinearities (a) NL=2 and (b) NL=3 at SNR=20 dB

The convergence behavior for Channel 3 for different SNR conditions ranging from 10-30dB by considering both the nonlinearities i.e. NL=2 and NL=3 is shown in Figs. 3.16-3.18. These figures indicate that CSA has a better exploration capability and avoids the stagnation problem more efficiently. Thus, despite poor signal-to-ratio conditions (SNR=10 dB), CSA provided the lowest MSE. Moreover, even though MSE increases with an increase in EVR from 11.12 to 46.82 (Figs. 3.8, 3.12 and 3.16) for all the algorithms, CSA is performing better than other algorithms.

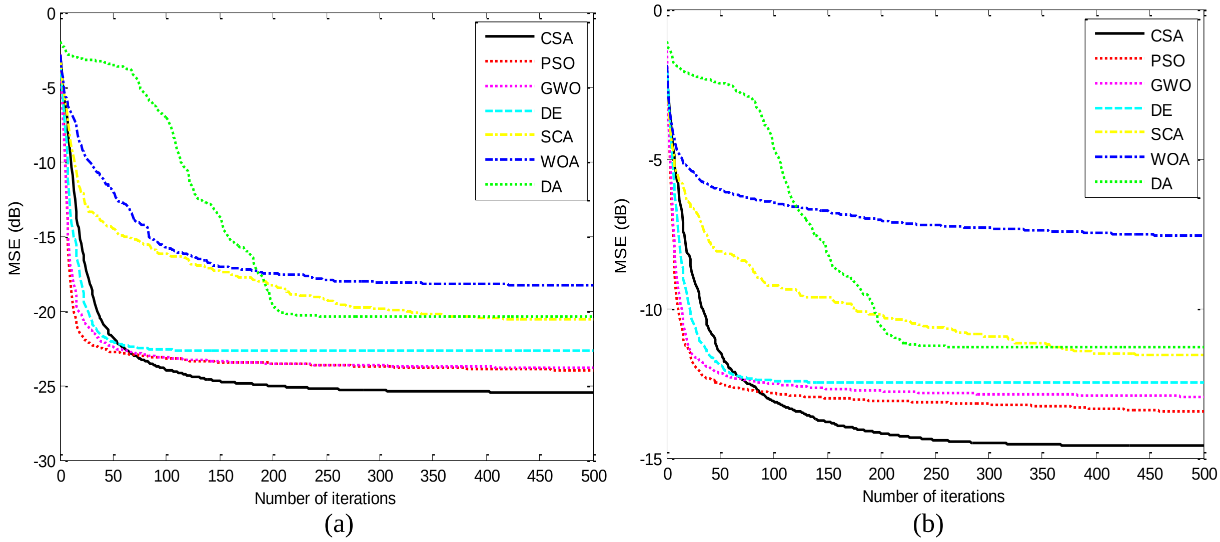
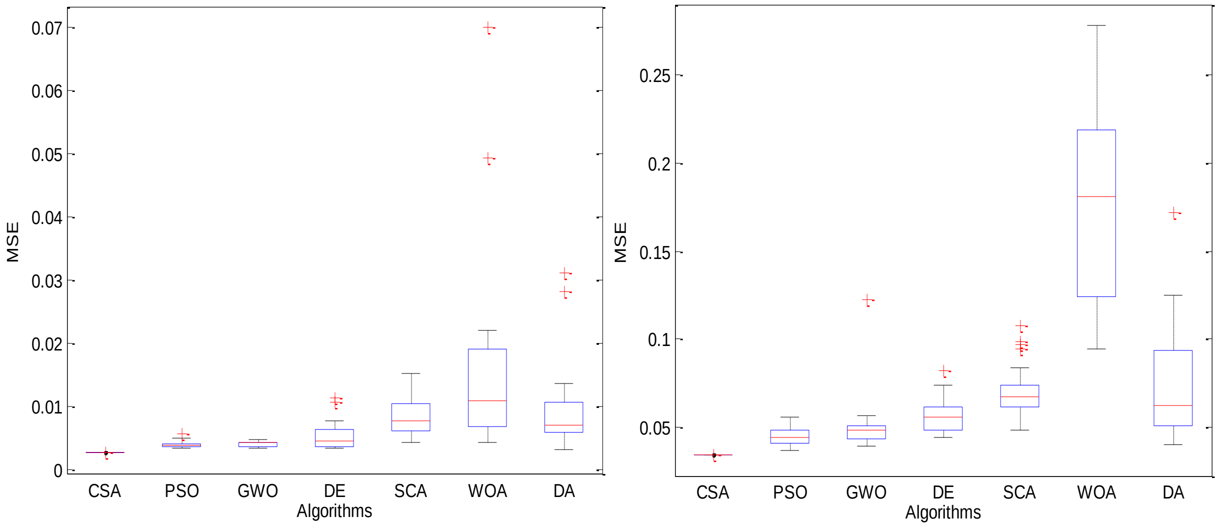


Fig. 3. 18. Convergence curves for Channel 3 with nonlinearities (a) NL=2 and (b) NL=3 at SNR=30 dB

Table 3. 10 MSE results for Channel 3 with nonlinearities NL=2 and NL=3 at SNR=30 dB

Non-Linearity	Algorithms	MSE				
		Min	Max	Median	Mean	SD
NL=2	CSA	0.0028	0.0029	0.0028	**0.0028**	**6.0392e-06**
	PSO	0.0034	0.0058	0.0038	0.0040	5.2184e-04
	GWO	0.0034	0.0049	0.0043	0.0041	3.9298e-04
	DE	0.0035	0.0113	0.0046	0.0054	0.0020
	SCA	0.0043	0.0152	0.0078	0.0088	0.0030
	WOA	0.0044	0.0699	0.0109	0.0149	0.0136
	DA	0.0032	0.0311	0.0072	0.0091	0.0062
NL=3	CSA	0.0348	0.0349	0.0348	**0.0348**	**1.3186e-05**
	PSO	0.0371	0.0559	0.0447	0.0451	0.0049
	GWO	0.0398	0.1228	0.0489	0.0508	0.0144
	DE	0.0446	0.0821	0.0562	0.0564	0.0092
	SCA	0.0484	0.1075	0.0673	0.0701	0.0144
	WOA	0.0942	0.2776	0.1810	0.1752	0.0595
	DA	0.0407	0.1720	0.0628	0.0745	0.0324

The statistical comparison of MSE for Channel 3 with both nonlinearities at SNR=30 dB is reported in Table 3.10. This table indicates the superior MSE performance of the CSA in comparison with other methods. The Box plot diagram of all the algorithms for Channel 3 with both nonlinearities at SNR of 30 dB is indicated in Fig. 3.19 and median and inter-quartile range in these boxplots confirms the superiority of the proposed method over other methods.



(a) (b)

Fig. 3. 19. MSE Box plots for Channel 3 with nonlinearities (a) NL=2 and (b) NL=3 at SNR=30 dB

3.4.3.2 Statistical Analysis

In population based metaheuristic algorithms, comparing the performance with the standard deviation and mean values is not adequate [140] and to analyze the statistical significance of the results obtained from the metaheuristic algorithm, performing the statistical test is essential [141]. Thus, to demonstrate the considerable gain in the performance of the algorithm in comparison with state-of-the-art algorithms it is obligatory to conduct the statistical test [132][16]. Hence, a nonparametric statistical test, Wilcoxon rank-sum test [142] , is performed to validate whether CSA based training scheme's results differ significantly from other algorithms. The Wilcoxon test is carried out by choosing the best algorithm and then comparing it with the other algorithms. Furthermore not applicable (NA) is noted down for the best performing algorithm.

Table 3. 11 *p*-values obtained from Wilcoxon test for all 3 channels corresponding to MSE results of SNR=30 dB

Algorithms	Channel 1		Channel 2		Channel 3	
	NL=2	NL=3	NL=2	NL=3	NL=2	NL=3
CSA	NA	NA	NA	NA	NA	NA
PSO	1.49180e-06	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11
GWO	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11
DE	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11
SCA	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11
WOA	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11
DA	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11	3.019859e-11

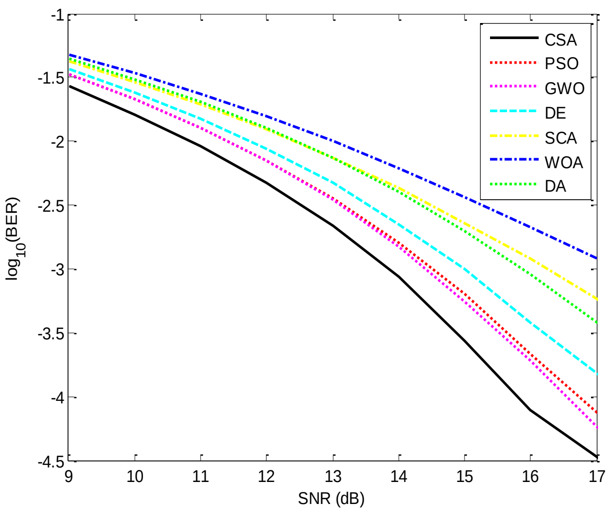
Table 3.11 presents the *p*-values computed by performing the statistical test. This table proves the superiority of CSA based training scheme over other algorithms for all the channels with both nonlinearities. The reported values of *p* in Table 3.11 are lesser than 0.05 which proves that the solutions provided by the CSA have statistical significance in comparison with other methods

3.4.3.3 Bit Error Rate (BER) performance

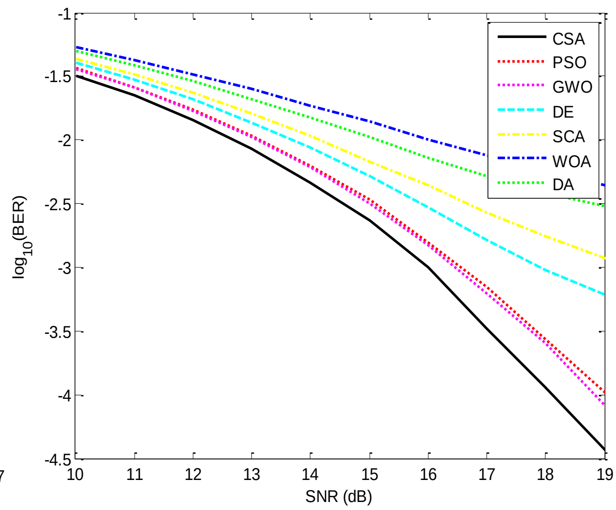
In this section, to analyze the bit error rate performance of the CSA based training scheme 3 different non-linear wireless communication channel models have been taken into consideration. To achieve this, AWGN with a wide range of signal-to-noise ratio is introduced to the output of a channel. During the channel equalization process, if the equalizer output and transmitted symbol are unequal, the error is increased by 1.

Case A: Channel 1

To analyze the bit error rate performance of the proposed scheme, this case considers a Channel 1 which corresponds to EVR of 11.12 with two different nonlinearities



(a)



(b)

Fig. 3. 20. BER performance for the Channel 1 with nonlinearities (a) NL=2 and (b) NL=3 at SNR=30 dB

Fig. 3.20 shows the BER curves of the CSA and other algorithms based training schemes for Channel 1 considering both the nonlinearities (NL=2 and NL=3) at SNR of 30 dB. An SNR enhancement of nearly 1 dB is attained by CSA over PSO and GWO at BER of 7.7667e-05 for NL=2 and above 2 dB for the remaining algorithms. Moreover, CSA has provided a similar BER improvement for the non-linearity NL=3. Fig. 3.20 (a) and (b) indicate that with an increase in nonlinearity from NL=2 to NL=3, BER of all the algorithms increased but still CSA provides lesser BER than others even in severely non-linear scenario (i.e. NL=3).

Case B: Channel 2

This case takes into consideration the Channel 2 corresponding to EVR of 21.71, to illustrate the BER performance of the CSA based training scheme.

The BER curves for Channel 2 having two different nonlinearities at SNR of 30 dB is shown in Fig. 3.21. This figure shows that for non-linearity NL=2, CSA has obtained around 0.75 dB gain in SNR over PSO and above 2 dB over the other compared algorithms at a bit error rate of 2.0633e-04. Moreover, it is observed from Fig. 3.21 (a) and (b) that CSA is outperforming the other six algorithms even more significantly for NL=3 than NL=2. This is the impact of **s**uperior exploration and exploitation capability of the CSA based training scheme, resulting in increased accuracy for channel equalization.

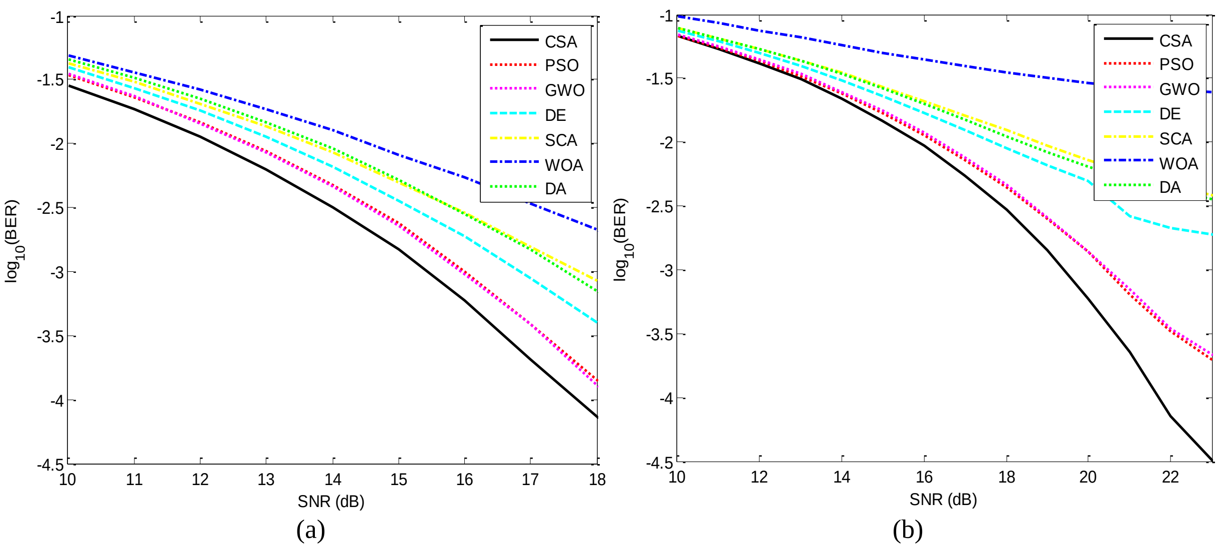
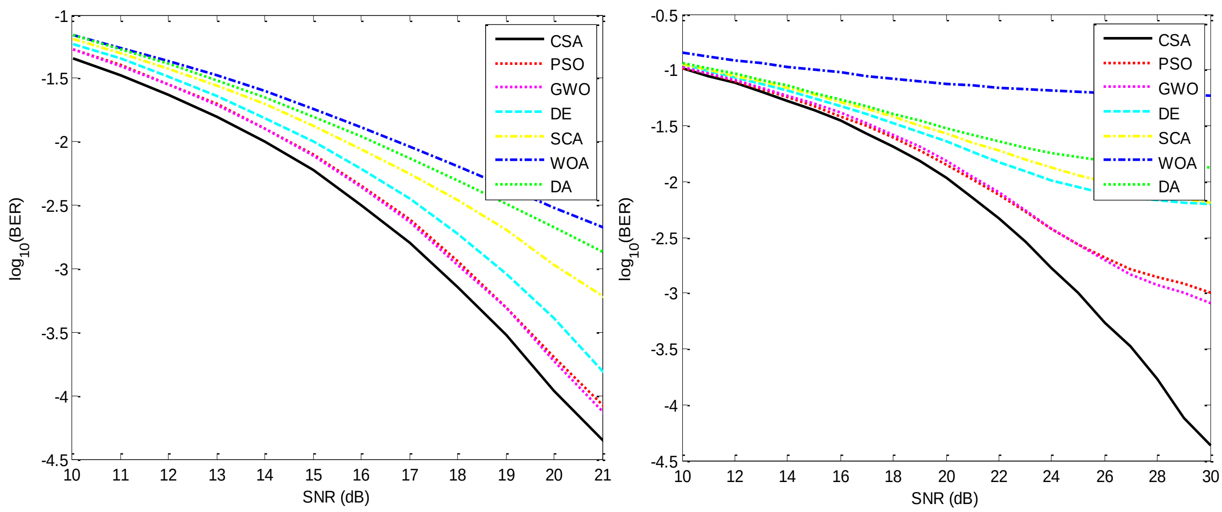


Fig. 3. 21. BER performance for the Channel 2 with nonlinearities (a) NL=2 and (b) NL=3 at SNR=30 dB

Case C: Channel 3

In this section, Channel 3 which is having the highest EVR is considered for validation of the BER performance of a CSA based training approach at SNR of 30 dB.

The BER performance of a highly dispersive channel (channel 3) is demonstrated in Fig. 3.22. Figs. 3.20-3.22 illustrates that despite the worsening of BER performance with increased EVR (from channel 1 (11.12) to channel 3 (46.82)), CSA has provided superior performance than other algorithms. Furthermore, Fig. 3.22 shows that although BER increased with non-linearity (NL=2 to NL=3), CSA has significantly outperformed the other algorithms and provided more pronounced results particularly for the severely non-linear scenario (NL=3). As can be seen from Fig. 3.22(b), CSA has attained almost 6 dB gain in SNR over PSO and above 6 dB over other algorithms at a bit error rate of 0.0017.



(a) (b)

Fig. 3. 22**.** BER curves for the Channel 3 with nonlinearities (a) NL=2 and (b) NL=3 at SNR=30 dB

3.4.3.4 Performance analysis in burst-error scenario

To verify the robustness of the proposed training method, its performance is evaluated in a burst error scenario occurring in the wireless environment. The use of a channel equalizer to deal with burst error scenarios is emphasized in the latest Patent [143]. Generally, the burst error occurs when there is an occurrence of consistent zeros or ones for a particular time interval [19]. The imperfect nature of the physical channels is responsible for the occurrence of burst error and it deteriorates the equalizer performance [19].

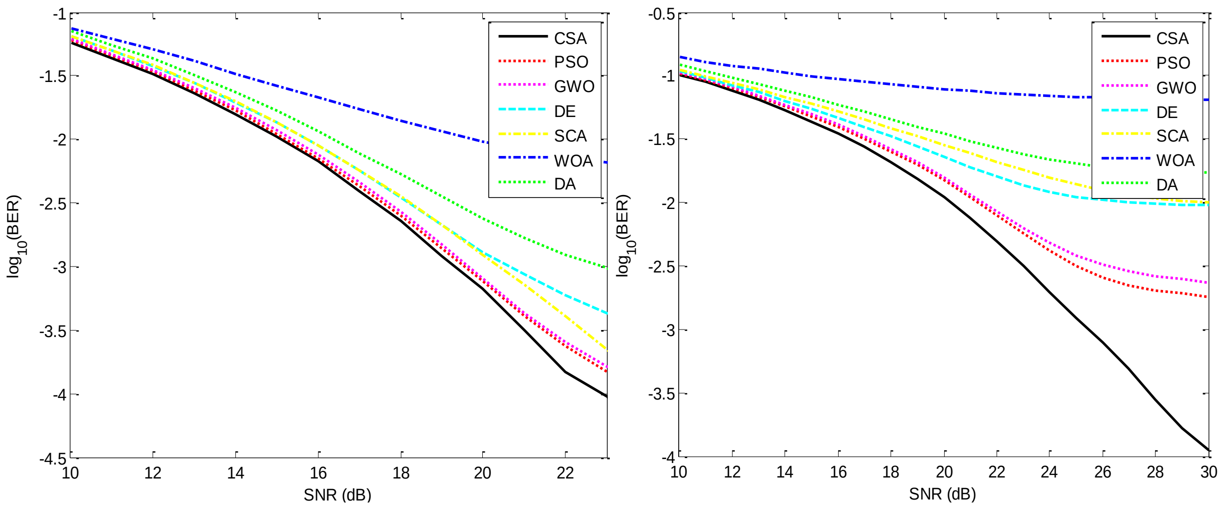


Fig. 3. 23**.** BER performance in burst error scenario for Channel 3 with nonlinearities (a) NL=2 and (b) NL=3

In order to evaluate the performance of a proposed training technique to handle burst error conditions, the training of FLANN based non-linear equalizer is performed for 500 iterations. The training is performed by considering the channel with the highest distortion (channel 3) with two different nonlinearities (i.e. NL=2 and NL=3) and testing is done after the completion of training. During the equalizer testing process, the BER obtained for channel 3 considering both nonlinearities (i.e. NL=2 and NL=3) is demonstrated in Fig. 3.23 (a) and (b) respectively. It can be noticed from this figure that, burst error conditions result in substantial performance degradation for all the algorithms and interestingly, CSA is able to tackle the difficulties encountered by other algorithms. However, for the burst error scenario the BER performance of CSA, PSO, GWO, DE, SCA, WOA and DA algorithm has been deteriorated (Fig. 3.23(a) and (b)) in comparison with BER without burst errors (Fig**.** 3.23 (a) and (b)) and BER performance of the CSA is still superior.

3.5 Conclusion

In this chapter, a scheme has been proposed for training the Functional link artificial neural network based non-linear channel equalizer by using Cuckoo Search Algorithm (CSA). The local minima avoidance capabilities and exploration competence of the proposed approach helped to discover the promising areas of the solution space. Furthermore, the exploitation ability assisted to completely exploit the promising region for finding the optimal solution. The balance between the exploration and fine-tuning ability of the scheme facilitated to achieve better accuracy in the non-linear channel equalization. Three non-linear channels were taken for simulations to validate the superiority of the CSA based training scheme and the results have been compared with recent algorithms like GWO, SCA, WOA, DA and well-established algorithms like PSO, DE. Furthermore, sensitivity analysis of the proposed approach was performed to optimize vital parameters for the CSA and the optimized values of these parameters were used to perform the simulation study. The simulations proved that CSA based training method offers improved performance in terms of MSE and BER when compared to existing algorithms**.** The improvement is more significant particularly in severely non-linear and highly dispersive channels. The robustness of the cuckoo search algorithm (CSA) based training scheme has been shown by considering the BER performance in a burst error scenario and it is observed that the scheme significantly outperforms the compared algorithms by effectively handling the burst errors. The performance of the proposed scheme has been validated for a wide range of signal-to-noise ratio (SNR 10 to 30 dB) values through simulation studies and it is observed that the scheme outperforms the other algorithms in poor SNR conditions as well. Moreover, the Wilcoxon rank-sum test proved that the proposed approach provided statistically significant results in comparison with competing approaches.

Chapter 4

An efficient JAYA Algorithm with Levy Flight

4.1 Introduction

This chapter involves the development of an efficient JAYA algorithm with Levy flight for non-linear channel equalization. JAYA is an effecient and simple population based metaheuristic algorithm. Hence, its application to channel equalization problem is worth investigating. Despite being an efficient and simple algorithm, JAYA gets trapped into local optima owing to its inadequate solution diversity and its weak exploration competence. To alleviate these issues, in this paper the concept of Lévy flight (LF) and greedy selection scheme has been incorporated into the basic JAYA algorithm. The LF concept enhances the population diversity and thus avoids the state of stagnation. The greedy selection scheme is employed to improve the exploitation ability without loss of population diversity. Furthermore, the exploitation and exploration capabilities of the algorithm have been balanced by proposing an adaptive Lévy index using a linear control parameter strategy. In order to validate the local optima avoidance ability, exploitation and convergence rate of the proposed JAYALF algorithm, it is tested on unimodal and multimodal benchmark functions and to verify the effectiveness of the JAYALF for non-linear channel equalization problem, three wireless communication channels with two different nonlinearities have been considered for simulation. In addition, the non-parametric pairwise Wilcoxon rank-sum test has been employed to test the statistical validity of the results obtained from JAYALF. The results of experiments and statistical test demonstrate that the proposed algorithm significantly outperforms JAYA, variants of JAYA, and other metaheuristic algorithms in terms of convergence rate, solution quality, and robustness. Furthermore, simulations show that proposed JAYALF algorithm provides faster convergence without being trapped into local optima and has a better exploration ability.

4.2 JAYA Algorithm

JAYA is one of the recent population-based metaheuristic algorithm, proposed by R. Venkata Rao in 2016 [69]. Most of the existing population based metaheuristic algorithms need algorithm-specific control parameters. Whereas, being an effective and simple algorithm, JAYA algorithm needs only number of iterations and population size as parameters. The basic idea behind this algorithm is that the solution obtained for a problem should escape from the worst solution and should approach to the best one [70]. Since its introduction, owing to simplicity and ability to find global optimum solution JAYA has been successfully used by researchers in many engineering problems. In recent few years, JAYA and its improved versions [71] [73] [72][74] [75] [76] [77][78][79] [80] have been used to solve various engineering problems. Furthermore, JAYA has been used for power quality improvement [81], optimization of heat sink [82], tracking of maximum power point (MPP) of PV array [83], reliability–redundancy allocation problems [84], optimization of heat exchangers [85], mechanical design optimization [86], optimization of machining performance [87], parameter identification of photovoltaic model [88] and design optimization of heat exchangers [89].

Various steps involved in JAYA algorithm are narrated as follows:

1) Initialize the population size (*N*), number of decision variables (*D*), upper and lower bounds of decision variables ($X_{\max,j}$, $X_{\min,j}$) and a maximum number of iterations $it_{\max}$ (i.e. $it = 1, 2, \ldots, it_{\max}$) as a termination criterion.

2) Randomly initialize the population of N solutions $X_i (i = 1, 2 \ldots N)$ within the boundary of a search space. Each j^{th} dimension of i^{th} solution i.e. $X_{j,i}$ is initialized between $X_{\max,j}$ and $X_{\min,j}$ as follow:

$$X_{j,i} = X_{\min,j} + rand(0,1) * (X_{\max,j} - X_{\min,j}) \tag{4.1}$$

where $rand(0,1)$ denotes a random number in the range 0 and 1 with uniform distribution and $X_{\max,j}$ and $X_{\min,j}$ represents the upper and lower bound of j^{th} dimension.

3) Calculate the fitness function F_i for every individual solution as per given problem and identify the best and worst solution i.e. X_{best} and X_{worst} respectively.

4) Considering the random numbers r_1 and r_2 between 0 and 1 with uniform distribution, the values for decision variables are updated as follows [69]:

$$X_{j,i}^{new} = X_{j,i} + \underbrace{r_1(X_{j,best} - |X_{j,i}|)}_{I} - \underbrace{r_2\,(X_{j,worst} - |X_{j,i}|)}_{II} \tag{4.2}$$

where $X_{j,i}$ denotes the j^{th} variable for i^{th} solution, $X_{j,best}$ denotes a j^{th} variable of best solution, $X_{j,worst}$ denotes the j^{th} variable of worst solution and $X_{j,i}^{new}$ denotes the modified version of $X_{j,i}$.

5) Evaluate the fitness F_i^{new} of each newly generated solution, compare it with the old fitness F_i of the previous solution.

6) For a maximization problem, if $F_i^{new} > F_i$ and for a minimization problem, if $F_i^{new} < F_i$ then replace X_i with a newly generated solution i.e. X_i^{new} otherwise keep the previous solution i.e. X_i and update the corresponding value of fitness function.

7) The updated values of fitness function are used to identify the best and worst solutions. The corresponding solutions are taken as the best and worst solution for the next iteration.

8) Repeat steps 4-7 until a termination criterion is reached and report the optimum solution X_{best}.

4.3 Lévy flight

Lévy flight (LF) is a random walk in which the lengh of steps is determined by the Lévy distribution [144][145]. The name Lévy flight is derived from the name of French mathematician, Paul Lévy. Lévy flight represents several phenomenon like earthquake analysis, fluid dynamics, cooling behavior, noise etc. [146]. Food searching path of several animals also depicted by LF [147] [148]. In the past few years, researchers utilized LF in Delay and Disruption Tolerant Network [146], human mobility fields [149] and Internet Traffic Models [150]. Lévy flight was utilized to improve the efficiency of the Bees algorithm [151], cuckoo search (CS) algorithm [57], particle swarm optimization algorithm [152], Firefly Algorithm[153], and grey wolf optimization algorithm[154]. These findings demonstrate that LF can significantly improve the performance of metaheuristic algorithms.

Lévy probability distribution is drawn in terms of a power-law formula $L(s) \sim |S|^{-1-\beta}$, where $0 < \beta \leq 2$ is an Lévy distribution index [144]. A basic form of Lévy distribution can be described as [145] [155].

$$L(s,\gamma,\mu) = \begin{cases} \sqrt{\dfrac{\gamma}{2\pi}} \exp\left[-\dfrac{\gamma}{2(s-\mu)}\right] \dfrac{1}{(s-\mu)^{3/2}} & if \quad 0 < \mu < s < \infty \\ 0 & if \quad s \leq 0 \end{cases} \tag{4.3}$$

where μ denotes a shift parameter and $\gamma > 0$ is a scale parameter

Lévy distribution should be described in terms of Fourier transform and can be expressed as [145] ,

$$F(k) = \exp[-\alpha |k|^{\beta}], \quad 0 < \beta \leq 2, \tag{4.4}$$

where α is skewness parameter or scale factor and β is Lévy index.

4.4 JAYA Algorithm with Levy Flight (JAYALF)

Despite being an efficient and simple algorithm, JAYA algorithm has some shortcomings. The basic idea behind this algorithm is that the solution obtained for a problem should escape from the worst solution and should approach to the best one [70]. Since with this approach all the solutions are attracted towards the best solution, the convergence speed of the algorithm is accelerated but it may results in loss of solution diversity. This issue may cause premature convergence of an algorithm to a local optimum solution.

To overcome these problems an efficient JAYA algorithm with Lévy flight (JAYALF) is presented in this chapter. Three key modifications have been proposed into basic JAYA to yield JAYALF algorithm. First, the concept of Lévy flight is incorporated to maintain the solution diversity and thus improve the global search capabilty. Second, the greedy selection scheme from Differential evolution (DE) algorithm [131] is employed to improve the exploitation capability without loss of diversity of the population. Third, an adaptive Lévy index is introduced to balance the exploration and exploitation capabilities of the algorithm throughout the search process.

When trapped into local optima, Lévy flight assists the JAYA algorithm to jump out of it towards a better solution. Thus, the LF can support JAYA in maintaining population diversity and enhancing diversification capability. The greedy selection strategy from the DE algorithm [131] is employed to improve the exploitation capability without loss of diversity of the population. The index of Lévy distribution β is made adaptive by increasing it over the course of iterations, which helps JAYA in maintaining the balance between exploration and exploitation tendencies throughout the search process.

Due to the incorporation of the Lévy flight (LF) concept into JAYA algorithm, solutions take long jumps which is very effective at the time of stagnation at local optima and helps in finding the promising region of the search space to obtain global optima. Thus the LF enhances the diversity of the solutions and facilitates JAYA to perform exploration of the entire fitness function landscape. Thus, in the proposed algorithm, the global search is enhanced through Lévy flight to eliminate the problem of stagnation at local optima.

New candidate solutions are generated using Lévy flight as follows [145]:

$$X^{new1} = X^{new} + \alpha \oplus Lévy(\beta) \tag{4.7}$$

where α is the step size and is taken as a random number for each dimension of the solution [152]

$$X^{new1} = X^{new} + random(size(D)) \oplus Lévy(\beta) \tag{4.8}$$

The scheme of creating step size s samples is described in [145,155]. The scheme is stated as follows [152]:

$$s = random(size(D)) \oplus Lévy(\beta) \sim 0.01 \frac{u}{|v|^{1/\beta}} (X^{new} - X_{best}), \tag{4.9}$$

u and *v* follows a normal distribution, i.e. $u \sim N(0, \sigma_u^2)$ $\quad$ v~N$(0, \sigma_v^2)$,

σ_u and σ_v are taken as follows:

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left[\left(\frac{1+\beta}{2}\right)\right]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta}, \quad \sigma_v = 1 \tag{4.10}$$

where Γ is the standard Gamma function

The greedy selection scheme from Differential Evolution (DE) algorithm [131] is used retain elite solution at each index. This scheme is described as follows:

$$X(it+1) = \begin{cases} X^{new1} & if \ f(X^{new1}) \leq f(X^{new}) \\ X^{new} & otherwise \end{cases} \tag{4.11}$$

Thus, if the new solution obtained through LF i.e. X^{new1} has a lower or equal value of the objective function (considering minimization problem), it replaces the corresponding previous solution X^{new} (Eq. (4.2)) else the previous solution is retained in the population. Therefore, the greedy selection scheme results in solutions with better or equal fitness but never results in inferior solutions. New solution X^{new} will be considered although objective function value is same for the previous solution and new solution X^{new}. This feature allows solutions to move over flat fitness landscapes with iterations [156]. The greedy selection scheme from the DE algorithm preserves the best solution obtained so far at each index and provides better exploitation ability without loss of diversity of the population.

In order to balance and use the exploration and exploitation capabilities of the JAYALF algorithm more effectively, a linear control parameter strategy is introduced, i.e. the index of Lévy distribution β is linearly increased over the course of iterations. Small values of β results in long jumps, and large values of β causes short jumps. Generally, small values of β are expected to cause jumps to unexplored regions of search space facilitating exploration and avoid getting stuck in local optima [152]. On the other hand, large values of β cause the search for new solutions in the obtained promising region facilitating exploitation. Hence, the value of β is linearly increased over the course of iteration as follows:

$$\beta(it) = \left[(\beta_{final} - \beta_{initial}) * \frac{it}{it_{\max}} \right] + \beta_{initial} \tag{4.12}$$

where $it_{\max}$ and it denotes the maximum number of iterations and the current iteration number respectively. β_{final} and $\beta_{initial}$ represents the final and initial values of the parameter β.

In this manner, small values of β at initial stage facilitate the global search and large values of β at a later stage accelerate local search in promising regions of search space enhancing the convergence towards the global optimum solution.

Thus, in this chapter, the authors have proposed a modified JAYA algorithm with Lévy flight (JAYALF) in order to eliminate the problem of getting trapped into local optima and to promote the exploration competence of basic JAYA algorithm. In JAYALF algorithm, after updating the solutions using JAYA as per Eq. (4.2) the better solution among old and new solutions are accepted. Furthermore, the solutions are upadted using Lévy flight as per Eq. (4.7). Greedy selection scheme from DE is used to select the best solutions at each index which helps in improving exploitation capability without loss of diversity. Moreover, to balance the exploration and exploitation capabilities of the JAYALF algorithm the index of Lévy distribution is made adaptive as per Eq. (4.12). The flowchart of JAYALF algorithm is shown in Fig. 4.1.

Various steps involved in JAYALF algorithm are narrated as follows:

1) Initialize the population size (N), number of decision variables $D(j=1,2...D)$, upper and lower bounds of decision variables ($X_{\max,j}$, $X_{\min,j}$) and maximum number of iterations $it_{\max}$ (i.e. it = 1, 2,…, $it_{\max}$) as a termination criterion.

2) Randomly initialize the population of N solutions X_i $(i=1,2...N)$ within the boundary of a search space. Each j^{th} dimension of i^{th} solution i.e. $X_{j,i}$ is initialized between $X_{\max,j}$ and $X_{\min,j}$ as per Eq. (4.1).

3) Calculate the fitness function F_i for every individual solution as per given problem and identify the best and worst solution i.e. X_{best} and X_{worst} respectively.

4) Considering the random numbers r_1 and r_2 between 0 and 1 with uniform distribution, the values for decision variables are updated using Eq. (4.2) for each solution.

5) Evaluate the fitness F_i^{new} for each newly generated solution, compare it with the fitness F_i of the previous solution.

6) For a maximization problem, if $F_i^{new} > F_i$ and for a minimization problem, if $F_i^{new} < F_i$ then replace X_i with a newly generated solution i.e. X_i^{new} otherwise keep the previous solution i.e. X_i and update the corresponding value of fitness function.

7) Rank all the solutions based on the fitness and perform Lévy flight on the best 50% of solutions as per Eq. (4.7).

8) Evaluate the fitness F_i^{new1} of new solutions generated via Lévy flight (X_i^{new1}).

9) Use a greedy selection scheme to select the best solutions among old solutions (X_i^{new}) and solutions generated via Lévy flight (X_i^{new1}) as per Eq. (4.11) and update the the new fitness values.

10) Update the value of the index of Lévy distribution β using Eq. (4.12).

11) The updated values of fitness function are used to identify the best and worst solutions. The corresponding solutions are considered as the best and worst solution for the next iteration.

12) Repeat steps 4-11 until the termination criterion is reached else report the optimum solution X_{best}.

4.5 Channel equalization as a JAYALF based optimization problem

The non-linear channel equalization problem has been solved by using a JAYALF algorithm. The FLANN equalizer structure trained using proposed JAYALF algorithm is shown in Fig. 2.

The steps used for training of FLANN are described as follows:

1. **Calculation of channel output**: The channel input $s(k)$ is a random sequence of binary signal taking values in the form {-1, +1}. The input signal after passing through a communication channel of transfer function $H(z)$ gets distorted by non-linearity $\psi(.)$ and further corrupted by additive white Gaussian noise $q(k)$ to give output $r(k)$ as per Eq. (2.3).

2. **Computation of FLANN structure input**: The received signal $r(k)$ is passed through tap delay section to give input vector $R(k)$ to FLANN structure.

3. **Expansion of input Vector**: The functional expansion of the input vector $R(k)$ is carried out using trigonometric functions which results in $R^e(k)$ as given in Eq. (2.8).

4. **Solution initialization**: The N number of candidate solutions are initialized with uniform distribution within the upper and lower limit of search space i.e. $X_{\max}$ and $X_{\min}$ as follows:

$$\begin{bmatrix} X_1 \\ X_2 \\ \vdots \\ X_N \end{bmatrix} = \begin{bmatrix} X_{1,1} & X_{2,1} & \cdots & X_{D,1} \\ X_{1,2} & X_{2,2} & \cdots & X_{D,2} \\ \vdots & \vdots & \ddots & \cdots \\ X_{1,N} & X_{2,N} & \cdots & X_{D,N} \end{bmatrix}, \tag{4.13}$$

where $X_{j,i}$ is the j^{th} dimension of i^{th} solution.

5. **Equalizer output and fitness calculation:** The actual output of the equalizer $y(k)$ is computed as per Eq. (2.12) for every k^{th} input sample. The error $e(k)$ is calculated as per Eq. (2.13) by taking the difference between actual output $y(k)$ and the desired output *d(k)*. The fitness function used is mean squared error (MSE) and is given by

$$MSE = \frac{1}{S} \sum_{k=1}^{S} e^2(k) \tag{4.14}$$

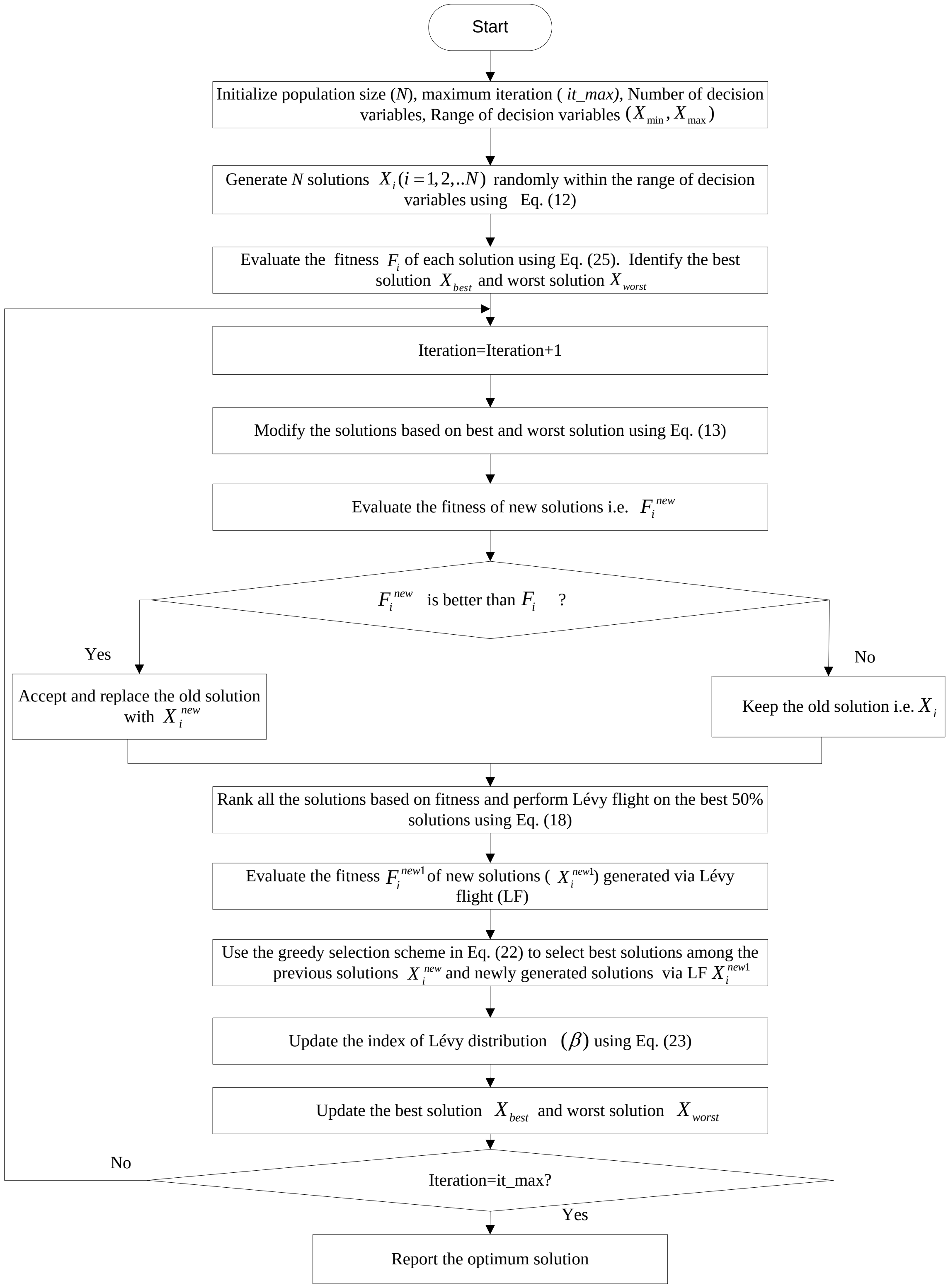


Fig. 4. 1**.** Flowchart of the proposed JAYALF algorithm

where S denotes the total number of transmitted samples or data block size, $e(k)$ is the error for k^{th} transmitted sample.

In the channel equalization problem, the goal is to minimize the fitness function value i.e. MSE. Identify the best solution i.e. X_{best} and the worst solution X_{worst} .

6. **Solution updating:** Update all the solutions according to Eq. (4.2) based on the best solution (X_{best}) and worst solution (X_{worst}) obtained in step 5 and accept the updated solutions if it is better than the old solution otherwise keep the old solution.

7. **Ranking the solutions:** Rank all the solutions based on fitness, identify the best solution and select the best 50% of solutions for Lévy flight.

8. **Perform Lévy flight**: Use the obtained X_{best} to perform Lévy flight using Eq. (4.7).

9. **Evaluate the fitness and use the greedy selection scheme:** Evaluate the fitness F_i^{new1} for each newly generated solution and compare it with the previous solution fitness F_i^{new}. Use the greedy selection scheme from DE to choose among newly generated solutions via Lévy flight X_i^{new1} and solutions X_i^{new} obtained in step 6. Update the value of β using Eq. (4.12).

10**. Termination criteria:** Repeat the steps 5-9 until a termination criterion is reached and report the optimum solution.

4.6 Simulation studies

All the experiments are carried out in a MATLAB R2013a platform on a 64 bit Windows 8.1 PC with an Intel i5-4590S 3 GHz processor and 8 GB RAM.

4.6.1 Channel characteristics for simulation

To examine the performance of the equalizer based on JAYALF and other competing algorithms extensive simulation experiments have been conducted for the channel equalization. The non-linear equalizer using functional link artificial neural network is optimized by JAYALF**,** JAYA [69]**,** moth flame optimization **(**MFO) [157], ant lion optimizer **(**ALO) [158], and SCA algorithm. Three benchmark channels with two different nonlinearities have been considered in this study. The channel transfer function considered in this study is taken from [108][2][19][1][11][159][37] [137] and is given by

$$H_1(z) = 0.209 + 0.995z^{-1} + 0.209z^{-2} \qquad \text{: Channel 1}$$

$$H_2(z) = 0.260 + 0.930z^{-1} + 0.260z^{-2} \qquad \text{: Channel 2}$$

$$H_3(z) = 0.341 + 0.876z^{-1} + 0.341z^{-2} \qquad \text{: Channel 3} \qquad (4.15)$$

The channels Channel 1, channel 2 and channel 3 corresponds to EVR values of 6.08, 11.12 and 46.82 respectively [1]. Following two different types of nonlinearity have been considered for simulation.

$$b(k) = t(k) \qquad \text{: NL=0}$$

$$b(k) = t(k) + 0.2t^2(k) - 0.1t^3(k) \qquad \text{: NL=1}$$

$$b(k) = \tanh(t(k)) \qquad \text{: NL=2} \qquad (4.16)$$

In Eq. (4.16) above $t(k)$ is the output of a wireless channel as shown in Fig. 2.1. NL=0 represents a linear channel without any non-linearity. NL=2 represents a nonlinearity which may occur due to saturation of amplifiers used at the transmitter and NL =1 is the arbitrary nonlinearity. Among the two nonlinearities, NL=2 is a case of severe non-linearity [19] [1] [137].

4.6.2 Sensitivity analysis of proposed JAYALF algorithm

As it is evident from the literature that the efficiency of an algorithm to a great extent depends on the precise tuning of its controlling parameters [108]. Therefore, sensitivity analysis of JAYALF is performed with respect to three key parameters namely population size (N), data block size (S) and a number of taps of the equalizer (M).

The outcomes of the sensitivity analysis are demonstrated in Figs. 5 to 7 for the channel 2 (Eq. (4.15)) with NL=1(refer to Eq. (4.16)).

A) Population size (N)

It is evident from Fig. 4.2 that the population size of N=30 is adequate in obtaining a global optimum solution which is also reasserted through quantitative assessment of the efficiency of JAYALF in terms of MSE in Table 4.1. It is also observed that any further increase in population size will hardly improve the performance.

Table 4. 1 Statistical comparison of MSE (over 30 independent runs) for variation of N

Population Size	MSE			
(N)	Best	Worst	Mean	Std. Dev.
5	6.4453e-04	0.1408	0.0072	0.0253
10	4.4062e-04	0.0015	8.2144e-04	2.8815e-04
15	4.0037e-04	0.0013	7.1544e-04	2.5808e-04
20	4.3322e-04	0.0013	6.6490e-04	1.7762e-04
25	4.3883e-04	0.0012	6.0668e-04	1.5034e-04
30	3.9093e-04	0.0012	**5.6593e-04**	**2.0378e-04**
35	3.3119e-04	0.0011	5.6341e-04	1.6183e-04

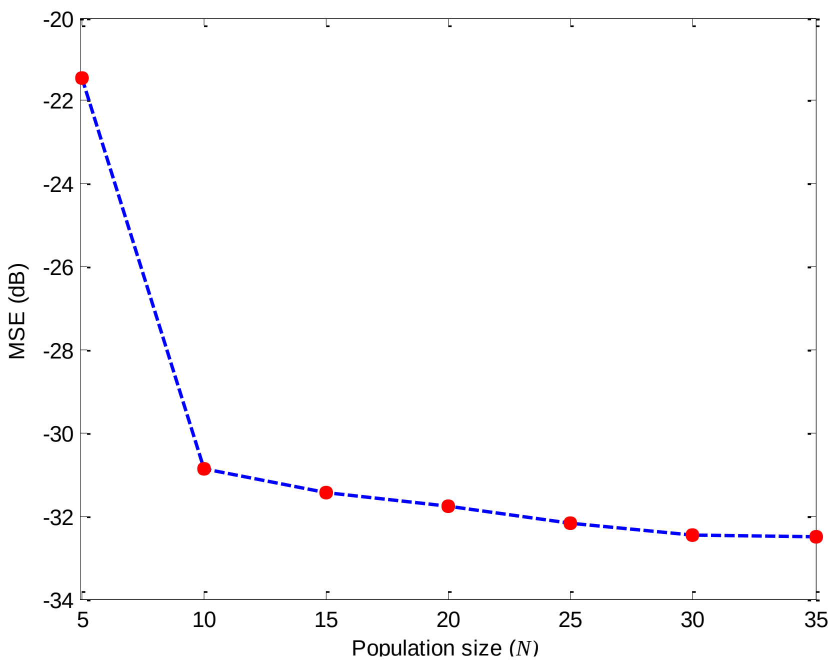


Fig. 4. 2. Effect of Population size (N)

B) Block Size (S)

It is evident from Fig. 4.3 and Table 2.2 that increase in the value of S from 50 to 200 results in a reduction of MSE. But, there is no significant improvement in MSE for values of S greater than 200. The requirement of data block size may vary from one problem to another.

Table 4. 2 Statistical comparison of MSE (over 30 independent runs) for variation of data block size (S)

Block Size(S)	MSE			
	Best	Worst	Mean	Std. Dev.
50	0.0048	0.0064	0.0052	3.0619e-04
100	3.0272e-04	9.5224e-04	5.6552e-04	2.1876e-04
200	3.8879e-04	9.1577e-04	**5.2992e-04**	**1.3169e-04**
500	3.8103e-04	9.0356e-04	5.2236 e-04	1.2907e-04

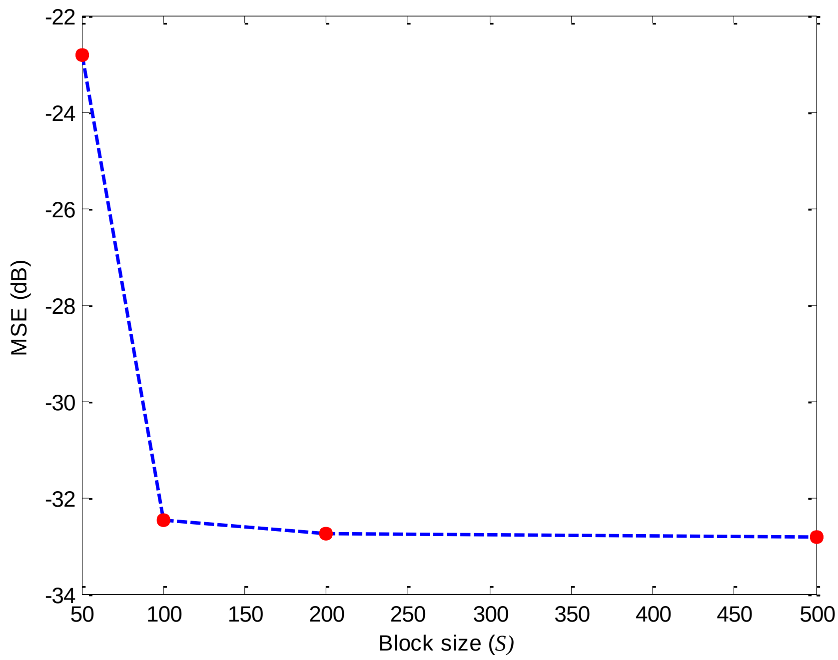


Fig. 4. 3. Effect of Block Size (S)

C) Number of taps of equalizer or order of equalizer (M)

Finally, Fig. 4.4 demonstrates the effect of a number of taps of an equalizer (M) on MSE in which the number of taps (M) is varied from 2 to 10 with a step increment of 2, from this it became apparent that equalizer with four taps gives the best performance. Fig. 4.4 and Table 3.3 shows that increasing value of M more than 4 doesn't results in any MSE improvement.

Table 4. 3 Statistical comparison of MSE (over 30 independent runs) for variation of number of taps of the equalizer (M)

No. of taps (M)	MSE			
	Best	Worst	Mean	Std. Dev.
2	0.0010	0.0010	0.0010	**2.6991e-06**
4	3.7933e-04	7.6902e-04	**5.0515e-04**	9.6004e-05
6	4.8715e-04	0.0021	0.0011	4.1962e-04
8	0.0010	0.0030	0.0018	5.2294e-04
10	0.0019	0.0057	0.0035	0.0011

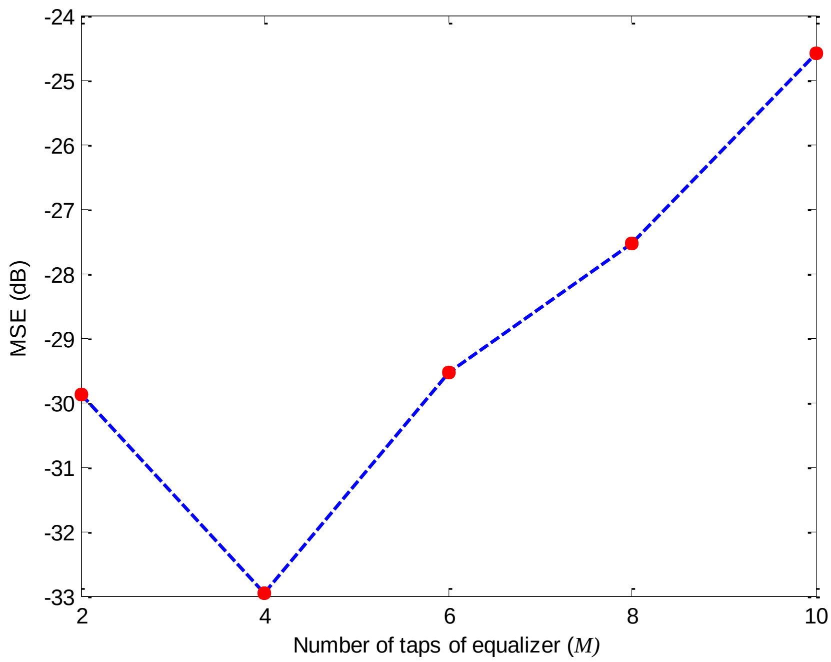


Fig. 4. 4. Effect of Number of taps of the equalizer (M)

4.6.3 Performance of JAYALF on benchmark functions

Before applying the proposed JAYALF algorithm for the non-linear channel equalization problem, we have tested its performance with unimodal and multimodal benchmark functions listed in Table 4.4. Detailed information about these benchmark functions is provided in [132][160]. The results in Table 4.4 are taken considering the population size as 30 over 500 iterations for all algorithms.

Recent literature [141] [140] suggest the use of statistical tests for confirming the statistical validity of results obtained from metaheuristic algorithms. For a particular problem to prove that a proposed new algorithm offers a substantial enhancement over other existing algorithms a statistical test is necessary [16]. To verify whether the results of JAYALF differ from JAYA, Jaya algorithm with time-varying acceleration coefficients (Jaya-TVAC), LJaya-TVAC [84], MFO, ALO and SCA [132] in a statistically substantial way, a Wilcoxon's rank-sum test [142], was conducted. The obtained values of *p* from Wilcoxon's test are reported in Table 4.5.

Table 4 demonstrates that the proposed JAYALF provides superior perfromance for 12 out of the 17 different benchmark functions. These results show that JAYALF has high exploration and local optima avoidance capability when compared to other algorithms. The p-values in Table 5 are very much less than 0.05 which confirms the statistical significance of results obtained by JAYALF.

Table 4. 4 Minimization results of 17 benchmark functions **(**over 30 independent runs**)**

Function	D	Algorithm	Best	Worst	Mean	Std. Dev.
Sphere	20	JAYALF	1.9195e-04	0.0016	4.8338e-04	2.8421e-04
		JAYA	0.0095	0.0415	0.0210	0.0091
		Jaya-TVAC	0.0027	0.0396	0.0162	0.0078
		LJaya-TVAC	7.6137e-30	5.4403e-22	**4.3746e-23**	1.3833e-22
		MFO	3.0528e-04	0.0668	0.0039	0.0120
		ALO	2.5623e-06	6.0643e-05	2.1889e-05	1.6521e-05
		SCA	2.9342e-06	1.9206	0.0753	0.3497
Schwefel 2.21	20	JAYALF	1.1658	2.4560	**1.7295**	**0.3943**
		JAYA	1.9788	9.2245	4.4490	1.5886
		Jaya-TVAC	2.0973	8.1774	4.0385	1.2923
		LJaya-TVAC	0.1648	6.4234	1.7426	1.4746
		MFO	15.4269	68.1830	44.9348	12.8716
		ALO	1.8430	16.8084	8.0079	3.9917
		SCA	0.3442	16.8323	5.0126	4.9542
Step	20	JAYALF	0.1512	0.3926	0.2472	0.0611
		JAYA	1.2912	3.4253	2.4270	0.5027
		Jaya-TVAC	0.9453	2.9982	1.8338	0.4111
		LJaya-TVAC	1.5758e-06	5.3491e-04	9.6220e-05	1.2559e-04
		MFO	7.0991e-05	2.0001e+04	2.3301e+03	5.0351e03
		ALO	1.4203e-06	8.5339e-05	**2.8450e-05**	**2.4857e-05**
		SCA	1.9370	3.5614	2.4670	0.3788
Schwefel 2.26	20	JAYALF	1.5989e+03	2.8202e+03	**2.2759e+03**	**313.3233**
		JAYA	2.3037e+03	4.9587e+03	3.9545e+03	689.9680
		Jaya-TVAC	1.9844e+03	5.0378e+03	4.1229e+03	698.3834
		LJaya-TVAC	1.2649e+03	3.7078e+03	2.2868e+03	611.2557
		MFO	950.5412	3.6125e+03	2.5772e+03	654.1796
		ALO	2.5464e+03	4.7679e+03	4.6111e+03	397.3435
		SCA	4.6166e+03	5.8537e+03	5.3296e+03	297.2677
Rastrigin	20	JAYALF	26.6811	45.8806	37.9985	5.4067
		JAYA	92.5149	160.6344	134.1872	16.9918
		Jaya-TVAC	109.1304	173.8001	138.2198	15.3030
		LJaya-TVAC	5.8495	21.0024	13.5422	3.8439
		MFO	41.7883	161.1823	79.3619	25.8234
		ALO	23.8790	107.4550	55.2201	19.7609
		SCA	3.3390e-04	76.8442	**10.6840**	16.6174
Ackley	20	JAYALF	0.0110	9.2785	0.3277	1.6906
		JAYA	0.0512	2.7791	**0.2350**	**0.4876**
		Jaya-TVAC	0.0562	19.9615	0.8696	3.6175
		LJaya-TVAC	2.2204e-14	3.6549	1.6167	1.0755
		MFO	0.0045	19.9652	8.0294	8.9895
		ALO	0.0016	4.1672	1.8553	0.9445
		SCA	8.8175e-04	20.2355	9.5645	9.8779

Table 4.4 (Continued)

Function	D	Algorithm	Best	Worst	Mean	Std. Dev.
Griewank	20	JAYALF	0.0900	0.2040	0.1418	0.0337
		JAYA	0.0866	0.7753	0.5880	0.1442
		Jaya-TVAC	0.0950	0.8875	0.5782	0.1887
		LJaya-TVAC	0	0.1743	0.0374	0.0400
		MFO	9.7656e-04	0.0863	0.0361	0.0238
		ALO	5.7407e-04	0.0642	**0.0236**	**0.0173**
		SCA	6.6440e-04	0.9482	0.3221	0.2557
Penalized	20	JAYALF	0.0311	0.6360	**0.1635**	**0.1260**
		JAYA	0.9980	7.3751	3.8141	1.7262
		Jaya-TVAC	0.8936	9.6537	3.0623	2.3015
		LJaya-TVAC	1.9119e-07	1.8825	0.1691	0.3940
		MFO	9.0416e-05	117.5161	4.8557	21.32
		ALO	3.2157	25.1145	9.1656	4.6060
		SCA	0.2457	3.5319	0.8265	0.8210
Penalized 2	20	JAYALF	1.4334e-05	0.0024	**2.9155e-04**	**5.1886e-04**
		JAYA	0.0012	0.1079	0.0185	0.0210
		Jaya-TVAC	8.7231e-04	0.0450	0.0151	0.0129
		LJaya-TVAC	1.3498e-32	1.5975	0.0576	0.2910
		MFO	6.1736e-04	4.0732	0.4018	0.8578
		ALO	4.9551e-06	1.4474	0.0660	0.2622
		SCA	1.2220	12.1864	2.4759	2.4050
Foxholes	2	JAYALF	0.9980	0.9980	**0.9980**	2.7714e-10
		JAYA	0.9980	1.0654	1.0014	0.0123
		Jaya-TVAC	0.9980	1.0298	1.0010	0.0085
		LJaya-TVAC	0.9980	0.9980	**0.9980**	0
		MFO	0.9980	5.9288	1.7566	1.4128
		ALO	0.9980	5.9288	2.2190	1.5039
		SCA	0.9980	10.7632	1.9866	1.9065
Kowalik	4	JAYALF	3.1412e-04	4.3948e-04	**3.5087e-04**	**3.2623e-05**
		JAYA	3.1089e-04	0.0017	5.2702e-04	3.2828e-04
		Jaya-TVAC	3.2824e-04	0.0013	4.9511e-04	2.3584e-04
		LJaya-TVAC	3.0749e-04	0.0017	3.6339e-04	2.4823e-04
		MFO	6.1996e-04	0.0022	0.0011	4.7605e-04
		ALO	6.0523e-04	0.0209	0.0036	0.0068
		SCA	4.3499e-04	0.0016	9.4275e-04	3.7518e-04
Goldstein-Price	2	JAYALF	3.00000000	3.00000000	**3.00000000**	**1.3730e-15**
		JAYA	3.00001464	3.00912874	3.00125439	0.0018
		Jaya-TVAC	3.00000000	3.0045	3.0009	0.0011
		LJaya-TVAC	3.00000000	3.0000	3.00000000	1.4496e-15
		MFO	3.00000000	3.00000000	3.00000000	2.5283e-15
		ALO	3.00000000	3.00000000	3.00000000	9.5311e-13
		SCA	3.00000041	3.00061407	3.00011534	1.7069e-04

Table 4.4 (Continued)

Function	D	Algorithm	Best	Worst	Mean	Std. Dev.
Hartman 3	3	JAYALF	-3.86278214	-3.86278214	**-3.86278214**	**2.7101e-15**
		JAYA	-3.86278214	-3.86278214	**-3.86278214**	**2.7101e-15**
		Jaya-TVAC	-3.86278214	-3.86278214	**-3.86278214**	**2.6962e-15**
		LJaya-TVAC	-3.86278214	-3.86278214	**-3.86278214**	**2.7101e-15**
		MFO	-3.86278214	-3.86278214	**-3.86278214**	**2.7101e-15**
		ALO	-3.86278214	-3.86278214	-3.86278214	4.0630e-12
		SCA	-3.8614247	-3.84870459	-3.85381491	0.0025788
Hartman 6	6	JAYALF	-3.3220	-3.2031	**-3.2903**	**0.0535**
		JAYA	-3.3220	-3.1724	-3.2521	0.0597
		Jaya-TVAC	-3.3220	-3.2031	-3.2576	0.0595
		LJaya-TVAC	-3.3220	-3.2031	-3.2862	0.0553
		MFO	-3.3220	-3.0839	-3.2129	0.0742
		ALO	-3.3220	-3.1982	-3.2858	0.0562
		SCA	-3.1254	-2.6629	-3.0027	0.1131
Shekel 5	4	JAYALF	-10.1532	-2.6829	**-8.3036**	**2.7157**
		JAYA	-10.1532	-2.5205	-6.2536	2.7478
		Jaya-TVAC	-10.1532	-2.5458	-5.9859	2.7912
		LJaya-TVAC	**-10.1532**	-2.6305	-6.2316	3.3876
		MFO	-10.1532	-2.6305	-5.5554	3.2208
		ALO	-10.1532	-2.6305	-6.2066	2.9944
		SCA	-6.7683	-0.4982	-2.8016	2.0058
Shekel 7	4	JAYALF	-10.4029	-2.7659	**-9.7631**	**1.6818**
		JAYA	-10.4029	-2.6816	-7.9960	3.1225
		Jaya-TVAC	-10.4029	-1.8376	-7.9798	2.7812
		LJaya-TVAC	-10.4029	-2.7659	-8.7791	3.0012
		MFO	-10.4029	-2.7519	-6.1335	3.3775
		ALO	-10.4029	-2.7519	-6.3767	3.0046
		SCA	-5.9049	-0.5239	-2.8195	1.7343
Shekel 10	4	JAYALF	-10.5364	-10.1999	**-10.5252**	**0.0614**
		JAYA	-10.5364	-2.3826	-7.7707	2.9968
		Jaya-TVAC	-10.5364	-2.4110	-8.1621	3.1516
		LJaya-TVAC	-10.5364	-2.4217	-8.8293	3.1590
		MFO	-10.5364	-1.8595	-6.8863	3.7778
		ALO	-10.5364	-1.6766	-7.0519	3.6367
		SCA	-9.8907	-0.9403	-3.5906	1.9186

Table 4. 5 p values obtained for Wilcoxon rank-sum test corresponding to the results of Table 4.4

Function	Algorithms						
	JAYALF	JAYA	Jaya-TVAC	LJaya-TVAC	MFO	ALO	SCA
Sphere	3.0199e-11	3.0199e-11	3.0199e-11	N/A	3.0199e-11	3.0199e-11	3.0199e-11
Schwefel 2.21	N/A	1.0776e-10	6.5949e-11	0.0500	2.9729e-11	1.0776e-10	0.0033
Step	3.0199e-11	3.0199e-11	3.0199e-11	0.0339	4.0772e-11	N/A	3.0199e-11
Schwefel 2.26	N/A	1.4643e-10	8.1014e-10	0.8073	0.0023	5.8561e-11	3.0199e-11
Rastrigin	2.4386e-09	3.0199e-11	3.0199e-11	0.0040	9.8951e-11	9.7555e-10	N/A
Ackley	5.5727e-10	N/A	0.7618	2.1213e-04	0.0012	6.5277e-08	0.1761
Griewank	3.0199e-11	3.0199e-11	3.0199e-11	0.3112	0.0364	N/A	4.4205e-06
Penalized	N/A	3.0161e-11	3.0161e-11	0.0011	0.0850	3.0161e-11	1.2857e-11
Penalized 2	N/A	5.4941e-11	6.0658e-11	0.0079	4.0772e-11	2.5974e-11	3.0199e-11
Foxholes	N/A	3.0199e-11	1.2118e-11	N/A	0.0243	0.0469	3.0180e-11
Kowalik	N/A	6.7320e-05	1.7171e-06	8.9487e-08	2.9766e-11	2.9803e-11	3.2949e-11
Goldstein-price	N/A	3.0199e-11	3.0199e-11	7.8180e-12	2.7167e-11	3.0180e-11	3.0199e-11
Hartman 3	N/A	N/A	N/A	N/A	N/A	1.2118e-12	1.2118e-12
Hartman 6	N/A	0.0495	0.1095	0.6755	4.7936e-04	0.0085	2.0008e-11
Shekel 5	N/A	0.0015	0.0015	0.0492	0.0438	0.0053	4.1743e-09
Shekel 7	N/A	0.9583	0.0531	0.0017	0.0948	0.0080	1.7667e-10
Shekel 10	N/A	0.0137	0.9350	0.0012	0.8291	0.0423	3.0161e-11

4.6.4 Application of JAYALF algorithm to Non-linear Channel Equalization

The transmitted digital message is with a 2-PAM signal. Each symbol is obtained from a uniform distribution taking values either +1 or -1. During the training of equalizer, the block size (S) of 200 samples is used as input. A zero mean white Gaussian noise of signal-to-noise ratio (SNR) 20 dB is added to the output of the channel. The tap delay section of FLANN having four taps providing 4 inputs to trigonometric expansion block. The trigonometric expansion block expands each term into three using $r(k)$, $\sin(\pi\, r(k))$, $\cos(\pi\, r(k))$ and one bias term thereby producing 13 terms.

The FLANN structure has been trained with JAYALF and other compared algorithms for 500 iterations. In every simulation experiment, the fitness function MSE, defined in Eq.

(4.14) is averaged over 30 independent runs. Wilcoxon's rank-sum test [142], is carried out at a 5% significance level. The population size taken as 30 for all the metaheuristic algorithms.

4.6.4.1 MSE performance

To study the convergence performance of JAYALF and competing algorithms, the equalizer is trained with 500 iterations for each algorithm. The learning performance of the algorithms for 3 different channels with two different nonlinearities is demonstrated in this section.

Case I: Non-linearity 1(NL=1)

In this case, nonlinearity considered is NL=1 as given in Eq. (4.16), which is one of the arbitrary non-linearity encountered in a communication system.

Table 4. 6 Statistical comparison of MSE (over 30 independent runs) for channel 1 with NL=1

Algorithm	MSE(Training)				*p* values
	Best	Worst	Mean	Std. Dev.	
JAYALF	8.6017e-05	2.3488e-04	**1.1830e-04**	**3.2726e-05**	N/A
JAYA	1.0337e-04	4.1708e-04	2.0000e-04	6.7348e-05	1.1567e-07
MFO	9.0751e-05	8.7533e-04	2.2774e-04	1.9961e-04	8.6634e-05
ALO	1.1465e-04	4.5701e-04	2.3603e-04	1.0637e-04	2.0062e-08
SCA	1.5564e-04	0.0014	6.1906e-04	3.1958e-04	6.6955e-11
BP	7.8487e-04	0.0011	9.2236e-04	1.0043e-04	3.0199e-11

Fig. 4.5 depicts the learning performance of all the algorithms for channel 1 with non-linearity NL=1. As evident from Fig. 4.5 JAYALF provides better convergence and lesser MSE when compared to other algorithms. Moreover, the MSE results in Table 4.6 shows the efficiency of the FLANN equalizer optimized by JAYALF. The *p*-values in Table 4.6 proves the statistical validity of results obtained from JAYALF. Furthermore, the box plots in Fig. 4.6 shows the superiority of the JAYALF over other competing algorithms.

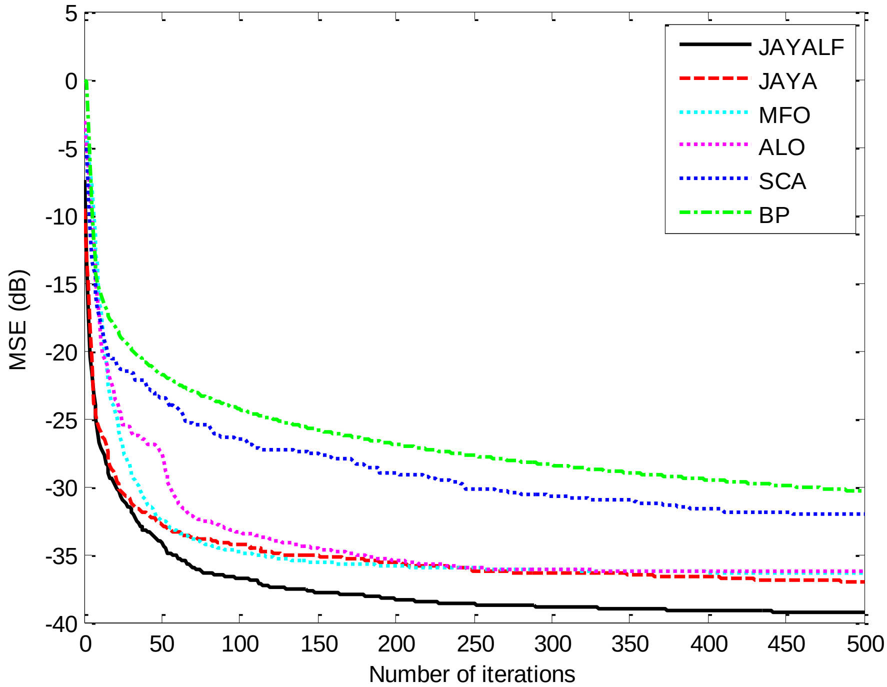


Fig. 4. 5**.** Learning curves of six algorithms for channel 1 with NL=1

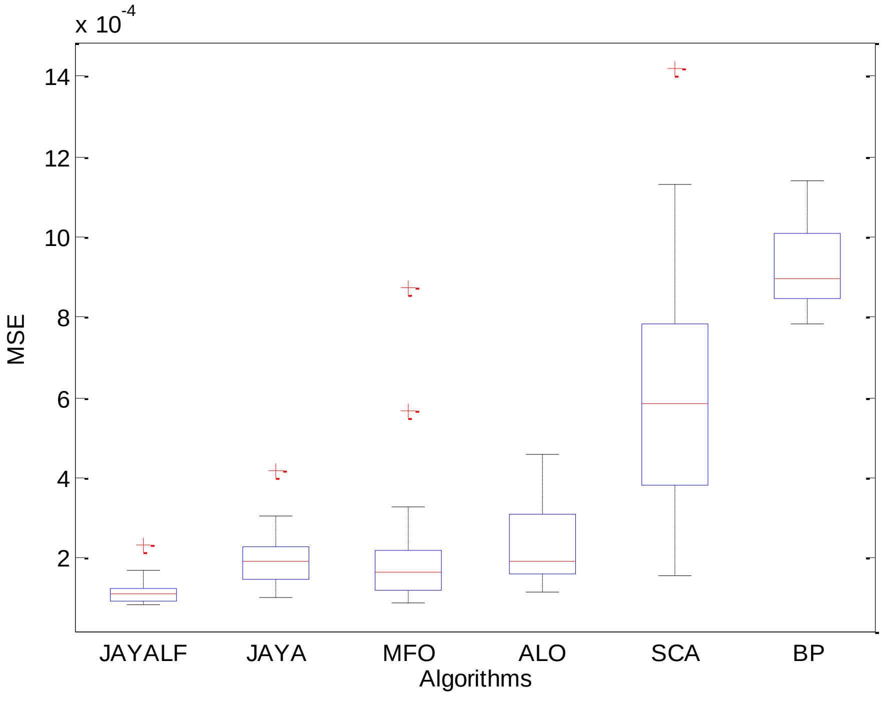


Fig. 4. 6**.** MSE box plot over 30 independent runs of six algorithms for channel 1 with NL=1

A similar performance is attained by JAYALF for channel 2, which has a higher eigenvalue ratio (EVR) of 11.12 as shown in Fig. 4.7. As evident from Figs. 4.5 and 4.7, with an increase in EVR, the performance of all the algorithms degrades but still the performance of JAYALF is consistent and superior when compared to other competing algorithms. Standard deviation and *p*-values in Table 4.7 demonstrates the reliability of results obtained by JAYALF algorithm over other algorithms. The MSE boxplots in Fig. 4.8 also illustrates the superior performance of JAYALF algorithm over other competing algorithms.

Table 4. 7 Statistical comparison of MSE (over 30 independent runs) for channel 2 with NL=1

Algorithm	MSE(Training)				*p* values
	Best	Worst	Mean	Std. Dev.	
JAYALF	3.8637e-04	9.0434e-04	**5.5901e-04**	1.2783e-04	N/A
JAYA	5.2643e-04	1.4000e-03	8.5418e-04	2.4880e-04	5.5999e-07
MFO	3.9947e-04	0.0027	9.7029e-04	5.8516e-04	4.9818e-04
ALO	5.6857e-04	0.0014	9.3462e-04	2.6556e-04	3.0797e-08
SCA	6.0819e-04	3.6000e-03	1.4000e-03	6.8234e-04	1.9568e-10
BP	0.0019	0.0023	0.0021	9.7390e-05	3.0199e-11

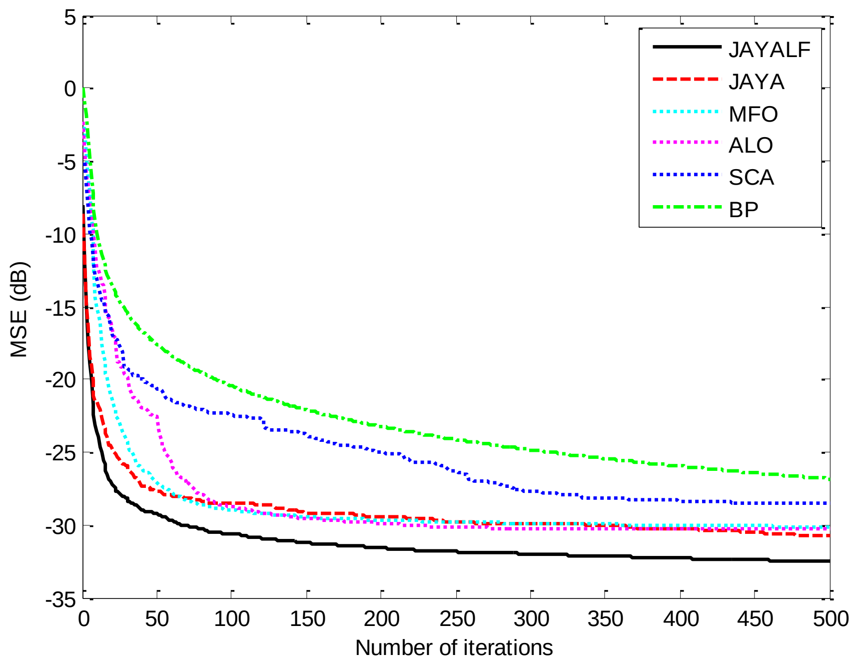


Fig. 4. 7. Learning curves of six algorithms for channel 2 with NL=1

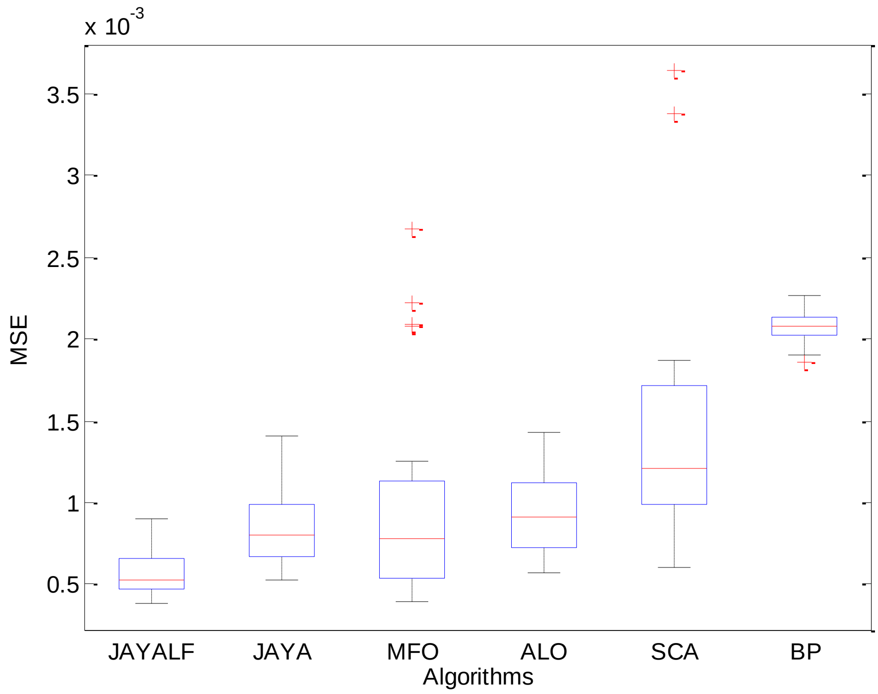


Fig. 4. 8. MSE box plot over 30 independent runs of six algorithms for channel 2 with NL=1

Finally, Fig. 4.9 illustrates the learning curves for Channel 3 which has EVR of 46.82. Channel 3 is a highly dispersive channel when compared to channel 1 and channel 2. As noticed from learning curves in Figs. 4.5, 4.8 and 4.9, an increase in EVR degrades the MSE performance of the algorithms but still JAYALF is performing consistently. The average MSE, standard deviation and *p* values in Table 4.8 and Box plot in Fig. 4.10 depicts the the superiority of JAYALF.

Table 4. 8 Statistical comparison of MSE (over 30 independent runs) for channel 3 with NL=1

Algorithm	MSE(Training)				*p* values
	Best	Worst	Mean	Std. Dev.	
JAYALF	0.0061	0.0081	**0.0069**	5.7143e-04	N/A
JAYA	0.0069	0.0151	0.0100	0.0021	3.8202e-10
MFO	0.0058	0.0262	0.0105	0.0051	0.0022
ALO	0.0070	0.0151	0.0109	0.0023	1.6123e-10
SCA	0.0087	0.0209	0.0122	0.0028	3.0199e-11
BP	0.0133	0.0149	0.0143	3.7566e-04	3.0199e-11

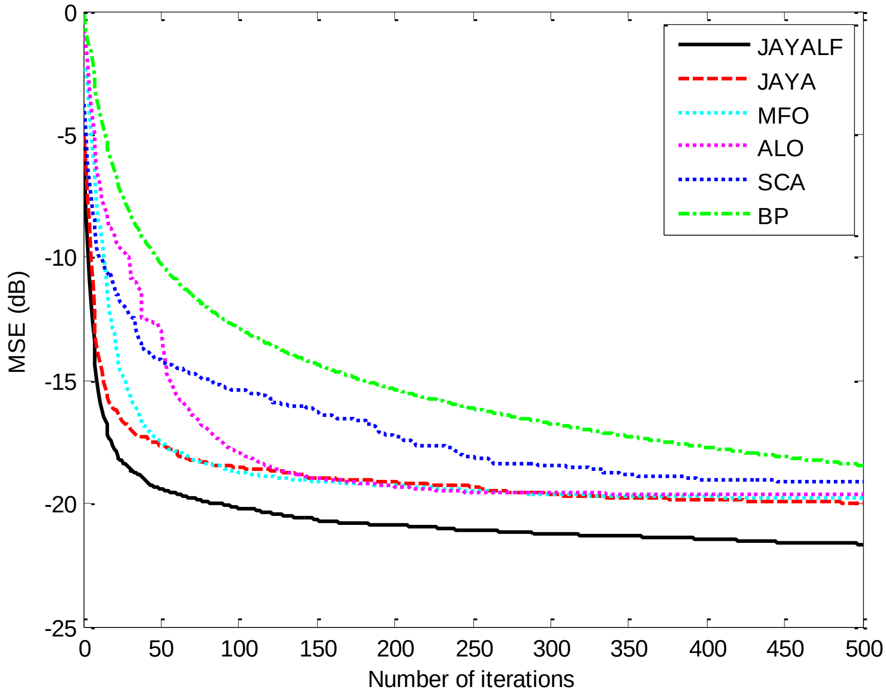


Fig. 4. 9**.** Learning curves of six algorithms for channel 3 with NL=1

Case II: Non-linearity 2

In this case, nonlinearity considered is NL=2 as given in Eq. (4.16), it represents a nonlinearity which occurs due to saturation of amplifiers used at the transmitter in a communication system. This non-linearity is more severe than non-linearity 1.

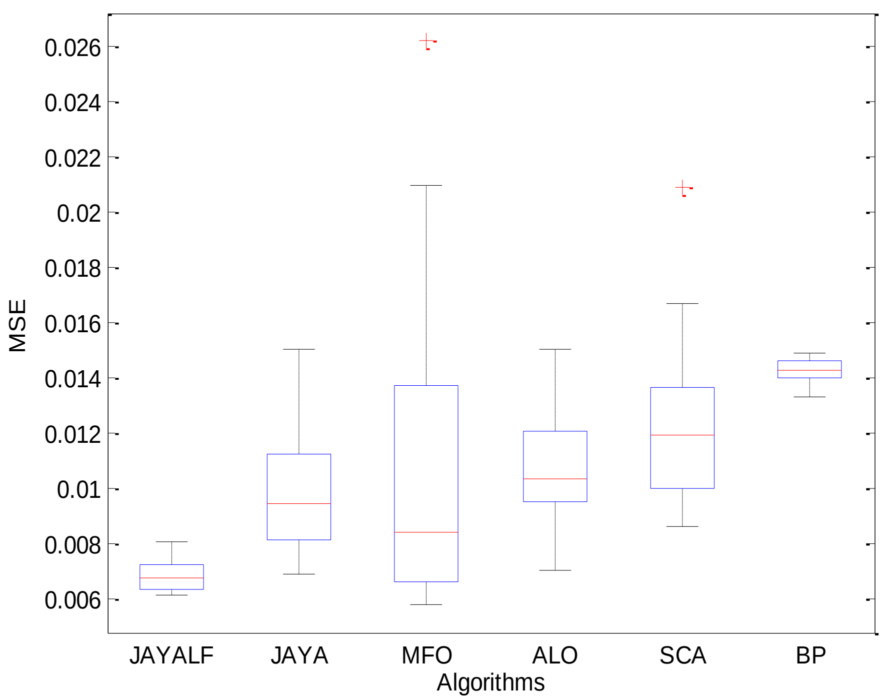


Fig. 4. 10 MSE box plot over 30 independent runs of six algorithms for channel 3 with NL=1

The MSE performance for channel 1 with severe non-linearity i.e. NL=2 (Eq. (4.16)) is shown in Fig. 4.11. An improved equalization performance of JAYALF is evident from MSE results Table 4.9 and boxplot in Fig. 4.12. The *p* values are much less than 0.05 for all algorithm which represents the statistical significance of results obtained by JAYALF. The convergence curves in Fig. 4.11 shows the better ability of JAYALF to avoid the stagnation at local minima problem with a higher convergence rate. It is observed from the MSE performance of channel 1 in Figs. 4.5 and 4.11, with an increase in non-linearity the MSE performance of all algorithms get deteriorated, but equalization performance of JAYALF is still consistent and superior when compared to other competing algorithms.

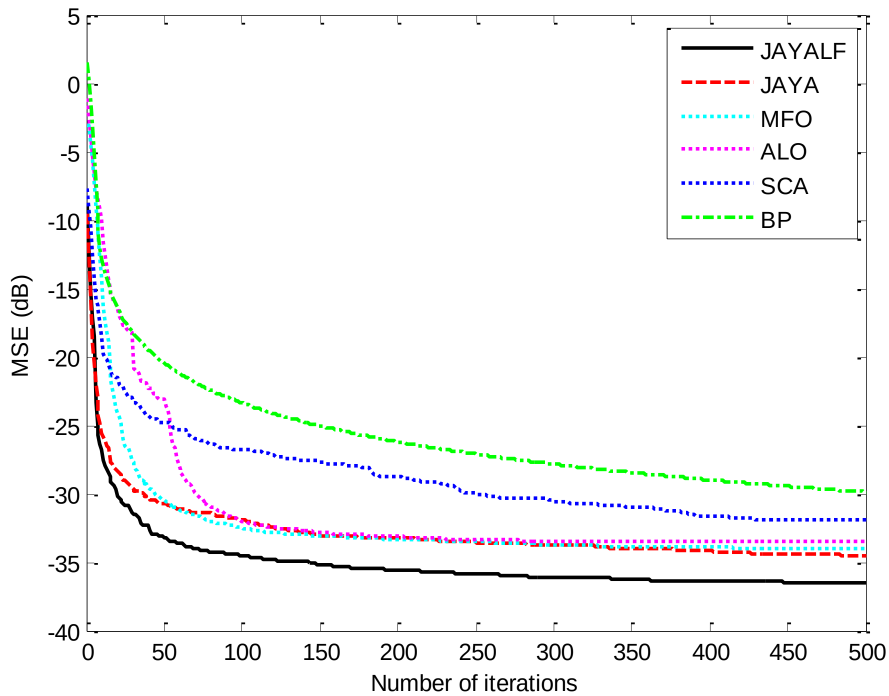


Fig. 4. 11. Learning curves of six algorithms for channel 1 with NL=2

Table 4. 9 Statistical comparison of MSE (over 30 independent runs) for channel 1 with NL=2

Algorithm	MSE(Training)				*p* values
	Best	Worst	Mean	Std. Dev.	
JAYALF	1.6660e-04	3.1953e-04	**2.2288e-04**	**3.3745e-05**	N/A
JAYA	2.6791e-04	5.4432e-04	3.5272e-04	7.3159e-05	2.3715e-10
MFO	1.6551e-04	9.3975e-04	3.9282e-04	2.3473e-04	7.2901e-04
ALO	2.2870e-04	8.2266e-04	4.4503e-04	1.9008e-04	1.4036e-09
SCA	3.4614e-04	0.0017	6.5410e-04	2.9980e-04	3.0199e-11
BP	9.1135e-04	0.0012	0.0010	7.1126e-05	3.0199e-11

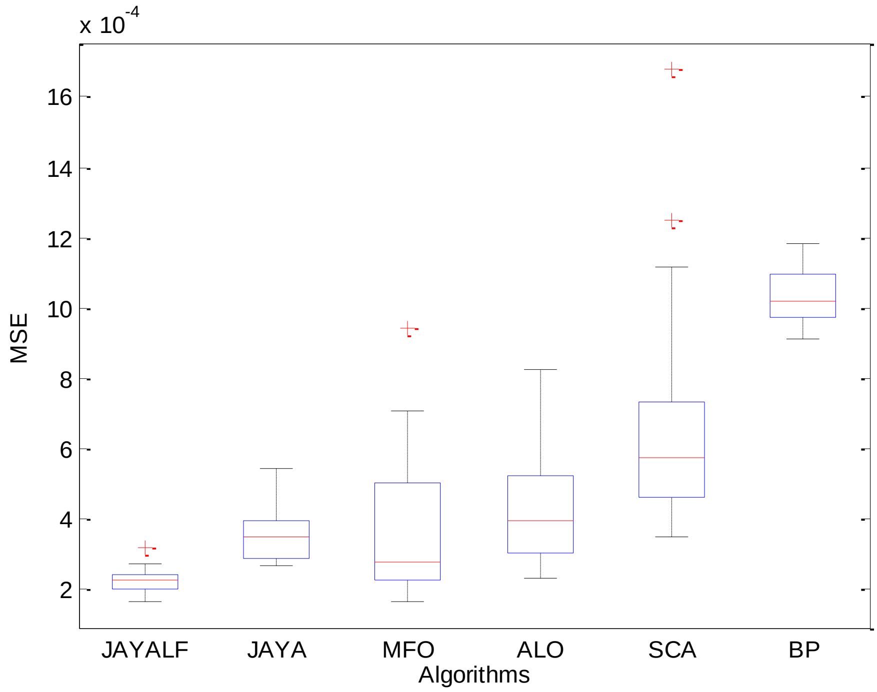


Fig. 4. 12. MSE boxplot over 30 independent runs of six algorithms for channel 1 with NL=2

A similar enhancement in performance is achieved for the channel 2 which is having more EVR than channel 1 and is demonstrated in Fig. 4.13. The convergence curves in Fig. 4.13 illustrates the higher convergence rate of the proposed algorithm when compared to other algorithms. The results for mean MSE, a standard deviation of MSE and *p*-values in Table 4.10 indicate that JAYALF is significantly better at avoiding local minima than other competing algorithms. The MSE box plot in Fig. 4.14 represents the distribution of results obtained by all six algorithms over 30 runs with the superiority of the proposed algorithm.

Table 4. 10 Statistical comparison of MSE (over 30 independent runs) for channel 2 with NL=2

Algorithm	MSE(Training)				*p* values
	Best	Worst	Mean	Std. Dev.	
JAYALF	5.2251e-04	7.0165e-04	**6.2625e-04**	**4.2394e-05**	N/A
JAYA	7.2172e-04	0.0016	0.0010	2.1419e-04	2.9155e-11
MFO	5.1465e-04	0.0031	0.0012	6.1850e-04	3.2708e-08
ALO	5.4195e-04	0.0019	0.0011	3.6498e-04	2.1403e-08
SCA	7.0691e-04	0.0039	0.0016	7.5407e-04	2.9155e-11
BP	0.0023	0.0026	0.0024	9.9695e-05	2.9000e-10

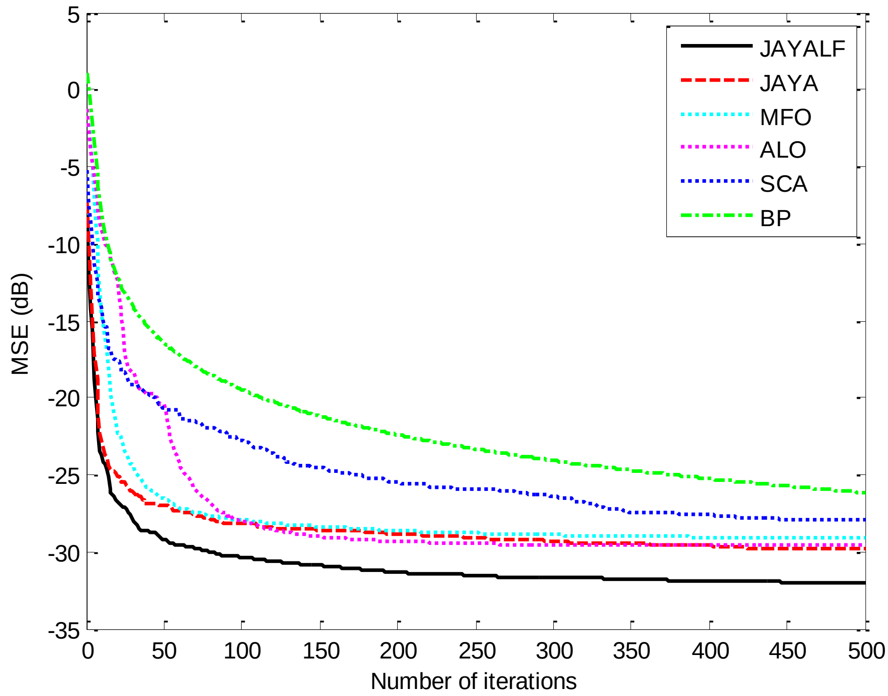


Fig. 4. 13. Learning curves of six algorithms for channel 2 with NL=2

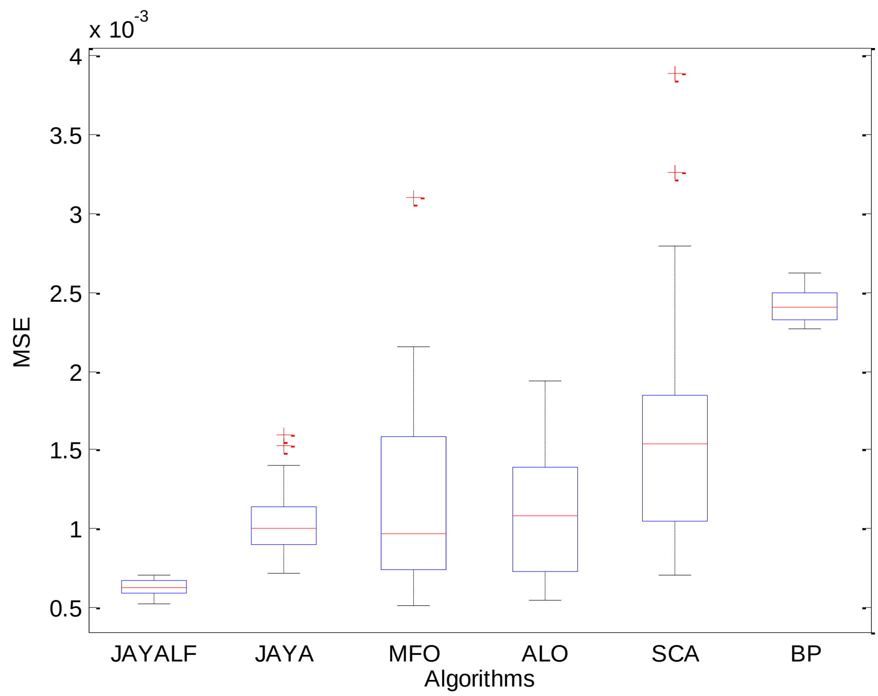


Fig. 4. 14. MSE box plot over 30 independent runs of six algorithms for channel 2 with NL=2

Finally, MSE performance for channel 3 with NL=2 which is a highly dispersive channel is depicted in Fig. 4.15. As observed from Figs. 4.11, 4.13 and 4.15, an increase in EVR from channel 1 to channel 3 results in a degradation in performance of all algorithms but still JAYALF is outperforming over other five algorithms. The enhanced efficiency of JAYALF in equalizing the channel is evident from the statistical comparison of MSE and *p* values as presented in Table 4.11 and MSE boxplot as shown in Fig. 4.16.

Table 4. 11 Statistical comparison of MSE (over 30 independent runs) for channel 3 with NL=2

Algorithm	MSE(Training)				*p* values
	Best	Worst	Mean	Std. Dev.	
JAYALF	0.0061	0.0083	**0.0071**	5.8771e-04	N/A
JAYA	0.0074	0.0118	0.0089	0.0011	1.8567e-09
MFO	0.0060	0.0181	0.0095	0.0024	1.4918e-06
ALO	0.0075	0.0117	0.0094	0.0014	4.6006e-10
SCA	0.0078	0.0222	0.0124	0.0037	4.5043e-11
BP	0.0145	0.0155	0.0151	2.4846e-04	3.0199e-11

4.6.4.2 BER performance

In order to explore the consistency in the performance of the proposed JAYALF algorithm, BER performance is evaluated for the three channels with two different nonlinearities. To calculate bit error rate (BER), 100,000 input samples are transmitted and noise of different SNR is added to channel output.

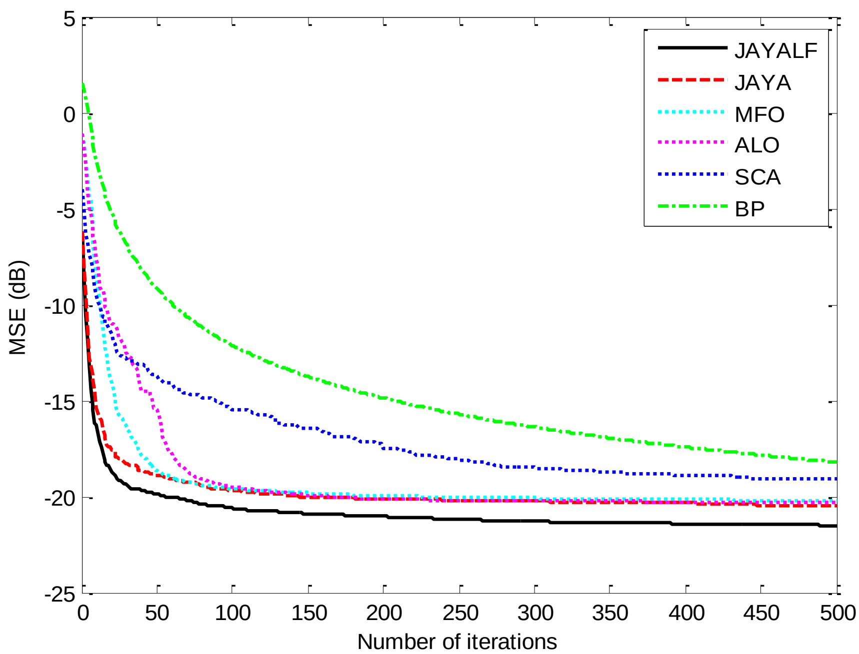


Fig. 4. 15. Learning curves of six algorithms for channel 3 with NL=2

Case I: Non-linearity 1 (NL=1)

In this case for analyzing BER performance, the non-linearity considered is NL=1 (Eq. (4.16)) which is one of the arbitrary non-linearity encountered in a communication system.

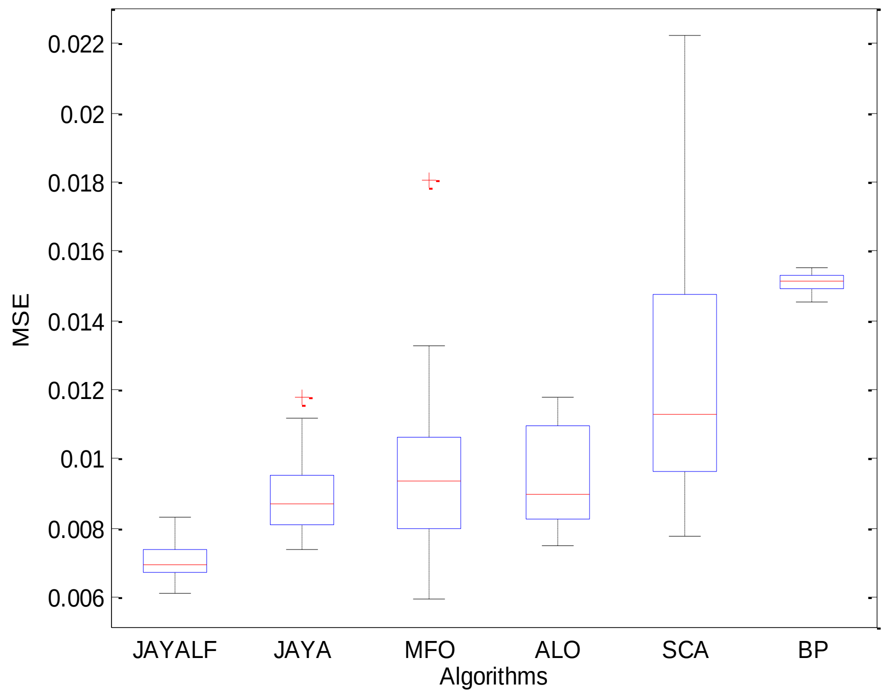


Fig. 4. 16. MSE box plot over 30 independent runs of six algorithms for channel 3 with NL=2

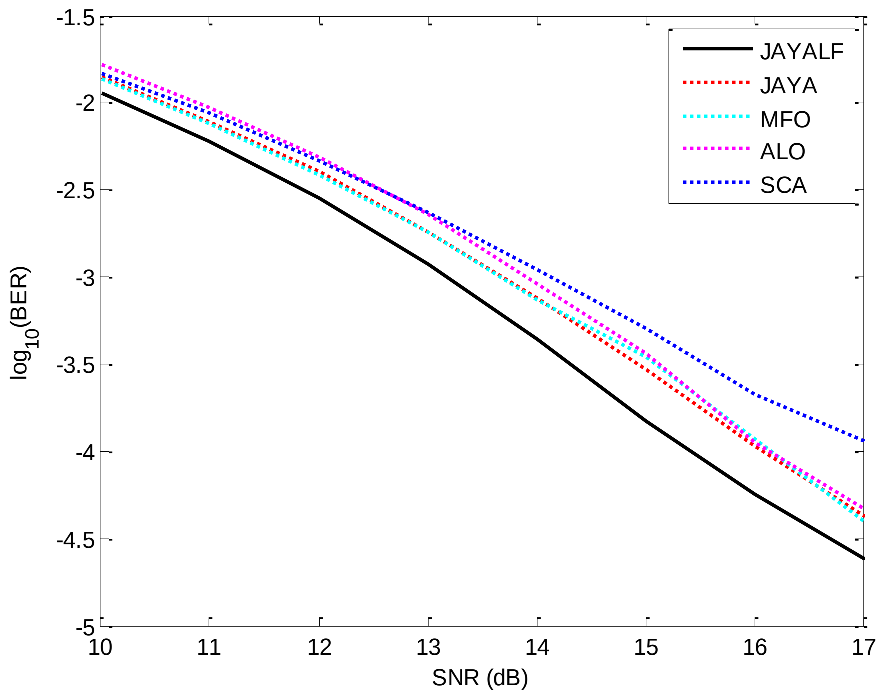


Fig. 4. 17. BER performance for Channel 1 with NL=1

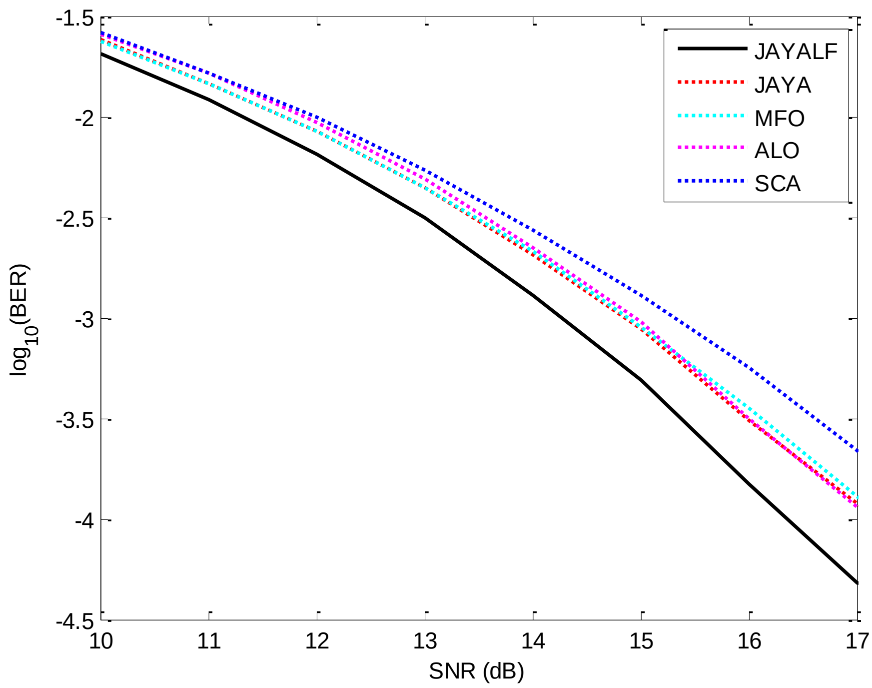


Fig. 4. 18. BER performance for Channel 2 with NL=1

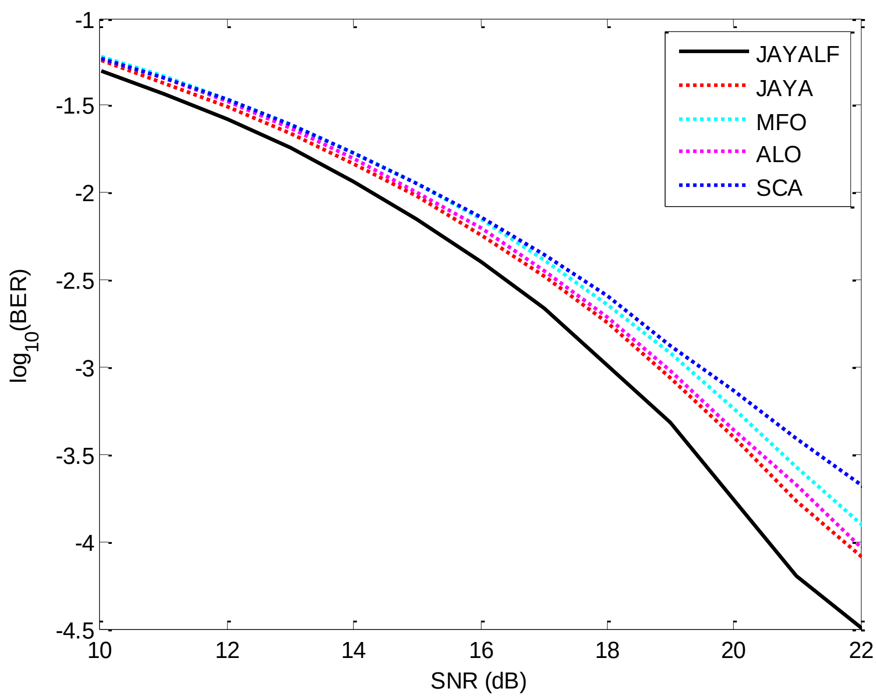


Fig. 4. 19**.** BER performance for Channel 3 with NL=1

The BER performance for channel 1 with non-linearity 1 is shown in Fig. 4.17 which depicts the superiority of JAYALF over the other five algorithms. A similar improvement in BER performance is attained by the proposed algorithm for Channel 2 with NL=1 which is demonstrated in Fig. 4.18. As depicted in this figure, a gain of approximately 1 dB is achieved by JAYALF at a BER of 1.4767e-04 over JAYA algorithm and more than 1 dB over other competing algorithms.

Finally, BER performance for channel 3 which is a highly dispersive channel is shown in Fig. 4.19. It observed from Fig. 4.19 that the proposed algorithm outperforms the other competing algorithms in terms of BER performance. The gain of approximately 1.5 dB is made by JAYALF over JAYA and more than 2 dB over the other algorithms. As can be seen from Figs. 4.17, 4.18, and 4.19 the superiority of JAYALF in term of BER performance is more significant for highly dispersive channel i.e. Channel 3.

Case II: Non-linearity 2(NL=2)

In this case, nonlinearity considered is NL=2 as given in Eq. (4.16), it represents a nonlinearity which occurs due to saturation of amplifiers used at the transmitter in a communication system. This non-linearity is more severe than non-linearity 1.

The BER performance for channel 1 with non-linearity 2 is shown in Fig. 4.20. The superior performance of JAYALF is evident from this figure. As seen from Figs. 4.17 and 4.20, with an increase in non-linearity the BER performance of all the algorithms degrades, but the superiority of JAYALF increases.

Similar enhancement in performance is attained for channel 2 and channel 3 which are more dispersive than channel 1. As it can be noticed from Figs. 4.20-4.22, with an increase in EVR from 6.08 to 46.82, the BER performance for all the algorithms gets deteriorated but the superiority of JAYALF is still consistent and more significant in case of highly dispersive channel i.e. channel 3.

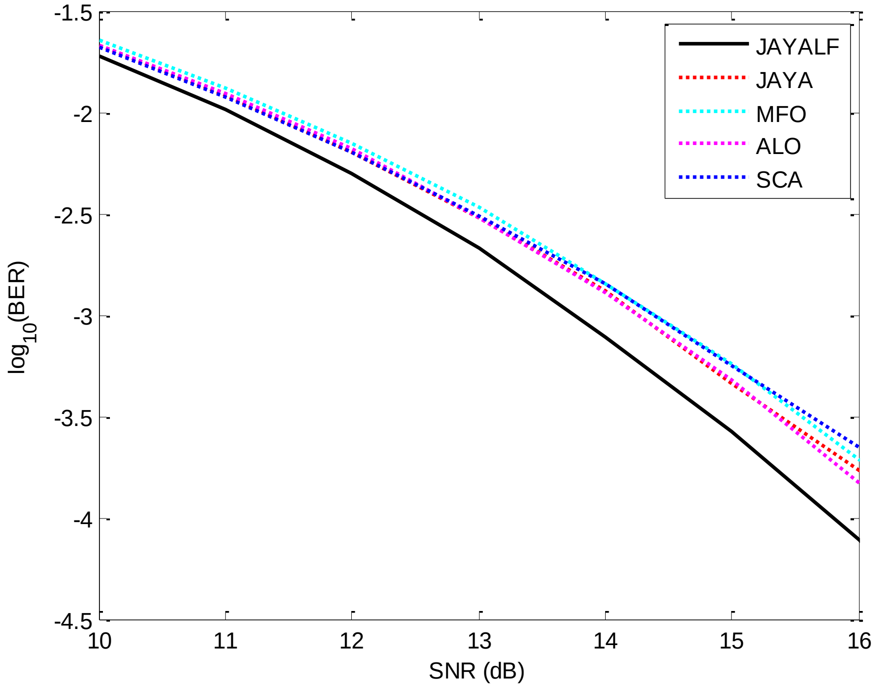


Fig. 4. 20. BER performance for Channel 1 with NL=2

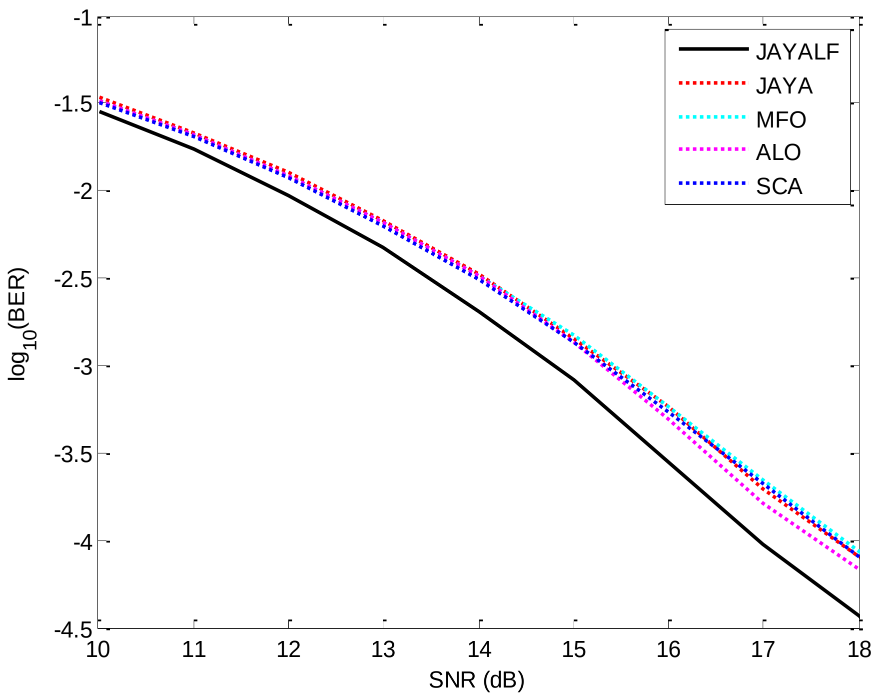


Fig. 4. 21. BER performance for Channel 2 with NL=2

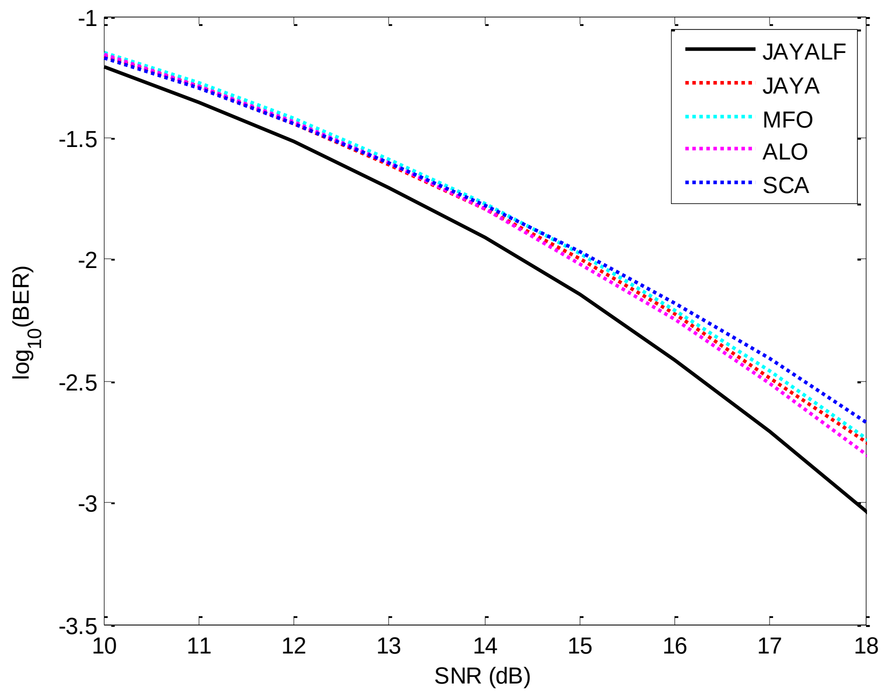


Fig. 4. 22. BER performance for Channel 3 with NL=2

4.7 Conclusion

In this chapter, an efficient JAYA algorithm with Lévy Flight (JAYALF) is proposed for the non-linear channel equalization problem. To alleviate the problem of loss of population diversity and stagnation at local optima, the concept of Lévy Flight has been incorporated into the basic JAYA algorithm. A greedy selection scheme has been employed to improve exploitation capability without loss of diversity. To balance the exploration and exploitation capabilities of JAYALF the adaptive Lévy index is proposed, which facilitates global search at the initial stage and local search at the latter stage. The exploration and local optima avoidance capabilities of the algorithm improved due to the incorporation of Lévy flight technique into basic JAYA algorithm. The Lévy flight based search enhanced the diversity of solutions and decreased the likelihood of converging on local optima, which in turn aided JAYALF to find promising regions of the search space. Moreover, the greedy selection scheme helped in fully exploiting the promising areas to find the global optimal solution. Thus, the proposed modifications support JAYALF in enhancing the exploration and exploitation capabilities throughout the search process.

The performance of the proposed algorithm was evaluated on unimodal and multimodal functions and results confirmed that JAYALF has better capability to escape from local optima with a higher convergence rate when compared to JAYA, variants of JAYA and other metahheuristic algorithms. To evaluate the performance of the proposed JAYALF algorithm for non-linear channel equalization problem, three wireless communication channels with two different nonlinearities were considered for simulation. The simulation studies showed that JAYALF based FLANN equalizer provides superior performance than other equalizers compared algorithms in terms of convergence speed, steady-state MSE, and BER.

Chapter 5

Modified Grasshopper Optimization Algorithm

5.1 Introduction

This chapter proposes a modified grasshopper optimization algorithm for equalization of non-linear wireless communication channels. Even though GOA is an efficient algorithm, it has drawbacks such as being trapped in local optima due to loss of swarm diversity and weakness of exploration capability. Another concern with GOA is that there is no provision to retain the elite grasshopper found so far at each index which weakens the exploitation capability and convergence rate of GOA. These limitations of GOA are alleviated in this work by incorporating the Lévy flight concept and greedy selection operator from Differential Evolution algorithm. Moreover, a threshold parameter is introduced to detect the inefficient search region. Lévy Flight is integrated with the basic GOA to improve the diversity of grasshopper swarm and the greedy selection operator is used to preserve the best grasshopper found so far at each index of swarm. The superiority of the proposed modified grasshopper optimization algorithm (MGOA) is illustrated over the existing metaheuristic algorithms. The key parameters of MGOA are selected by performing the sensitivity analysis. The simulation results on four non-linear channels demonstrate the efficiency of the proposed MGOA algorithm in terms of BER and MSE. The statistical validity of the results provided by MGOA is confirmed through Wilcoxon rank-sum test.

5.2 Grasshopper Optimization Algorithm (GOA)

Grasshopper Optimization Algorithm (GOA) is one among the latest population based metaheuristic algorithm, developed by Saremi et al. in 2017 [90]. The GOA is an efficient swarm intelligence based algorithm motivated from the team hunting behavior of grasshoppers. GOA mimics the behavior of grasshoppers by emitting the repulsion and attraction forces between them [92]. Generally, larval and adulthood are the two key phases of grasshopper life cycle [101]. It has been shown that GOA is capable of outperforming several leading metaheuristic algorithms [90].

Since its introduction, many complicated benchmark functions and engineering problems has been solved by GOA effectively [90,91]. GOA and its enhanced versions have been extensively used for electrical characterization of fuel cells [92], training of artificial neural network [91], feature selection [93–95], economic dispatch problem [96], data clustering [97], tuning of PID controller [98] and target tracking [99]. To enhance the performance of basic GOA, some improvement techniques have been introduced in [100–102], where the authors demonstrated the competitive performance of GOA over other metaheuristic algorithms. Recently, basic multi-objective GOA has been developed by Mirjalili et al. [103] and Tharwat et al. proposed an improved version of multi-objective GOA [104].

The three components which affect the grasshopper flying path are social interaction (S_i), wind advection (A_i) and gravity (G_i) and are mathematically modeled as follows [161][90]:

$$X_i = S_i + G_i + A_i \tag{5.1}$$

Where S_i denotes the social interaction component
A_i is the wind advection.
G_i represents the gravity force on the grasshopper and
X_i is the i^{th} grasshopper position,

Mathematically, the social interaction component S_i is given by [90]

$$S_i = \sum_{j=1, j\neq i}^{N} s(d_{ij})\ d_{ij} \tag{5.2}$$

Where $d_{ij} = \left| x_j - x_i \right|$ represents the distance among two grasshoppers *i* and *j*

The function *s* denotes the social forces which is given by [90],

$$s(r) = fe^{(-r/l)} - e^{-r} \tag{5.3}$$

Where *l* is the attractive length scale and *f* is the intensity of attraction.

The gravity component *G* is described as [90]

$$G_i = -g\hat{e}_g \tag{5.4}$$

Where $\hat{e}_g$ denotes a unity vector towards the center of the earth

g refers to the gravitational constant

The wind advection component A is written as follows [90]:

$$A_i = u\hat{e}_w \tag{5.5}$$

Where $\hat{e}_w$ indicates the unit vector in the wind direction

u denotes a constant drift

The updated form of Eq. (5.1) after Substituting the values of S, G and A can be rewritten as follows [90]:

$$X_i = \sum_{j=1, j\neq i}^{N} s\left(\left|x_j - x_i\right|\right)\frac{x_j - x_i}{d_{ij}} - g\hat{e}_g + ue_w \tag{5.6}$$

Where N denotes a total grasshoppers and $s(r)$ is given by Eq. (5.3).

For the convergence of algorithm to an optimum solution a modified version of Eq. (5.6) is given by [90]:

$$X_{i,d} = c\left(\sum_{j=1, j\neq i}^{N} c\frac{ub_d - lb_d}{2}\, s(\left|x_j^d - x_i^d\right|)\frac{x_j - x_i}{d}\right) + T_d \tag{5.7}$$

Where the decreasing coefficient c is used to reduce the repulsion region, comfort region and attraction region. T_d refers to d^{th} dimension of target grasshopper. The parameter c in Eq. (5.7) is reduced with iterations to balance the exploitation and exploration capabilities as follows [90]:

$$c = c_{max} - l\left[\frac{c_{max} - c_{min}}{L}\right] \tag{5.8}$$

Where l is a current iteration number,

c_{max} and c_{min} denotes the maximum and minimum value of c respectively,

L represents total iterations.

The steps of GOA algorithm are summarized as follows:

1. Initialize the parameters c_{min}, c_{max}, total iterations $it_{\max}$, number of decision variables K (d = 1, 2… K). Furthermore, generate a swarm of N grasshoppers X_i $(i = 1, 2 ... N)$ randomly in the upper and lower boundary $(x_{\max,d}\ ,\ x_{\min,d})$ of the search space.
2. Perform a fitness (F_i) calculation for every grasshopper and find the Target grasshopper X_{best}
3. Update the value of c using Eq. (5.8).
4. Normalize the distances among the grasshoppers within the range [1, 4] and use Eq. (5.7) to update the positions of all grasshoppers.
5. Check whether the updated solutions are within the range of decision variables, if not restrict them in the range $[x_{max}\ , x_{min}]$.
6. Calculate the new fitness F_i^{new} of every grasshoppers and update the target grasshopper as per new fitness.
7. If the total iterations are reached then report the target grasshopper X_{best} and stop else continue repeating the steps 3-6.

5.3 Lévy Flight

Lévy flight (LF) is a random walk in which the length of steps is determined by the Lévy distribution [144][145]. Lévy flight represents various phenomenon in the nature [146] and the food searching path of several animals also depicted by LF [147] [148]. In the recent years, a number of engineering problems have been solved using LF [146], [149] [150]. Furthermore, LF has been integrated with Bees algorithm [151], cuckoo search (CS) algorithm [57], particle swarm optimization algorithm [152] [162], Firefly Algorithm[153], and grey wolf optimization algorithm[154] to improve the solution diversity.

Lévy probability distribution is drawn in terms of a power-law formula as follows [144]:

$$L(s) \sim |S|^{-1-\beta} \tag{5.9}$$

where β is a Lévy distribution index and its value lies in the range of 0 to 2.

A Lévy distribution can be described as follows[145] [155]:

$$L(s,\gamma,\mu)=\begin{cases}\sqrt{\frac{\gamma}{2\pi}}\exp\left[-\frac{\gamma}{2(s-\mu)}\right]\frac{1}{(s-\mu)^{3/2}} & if\quad 0<\mu<s<\infty \\ 0 & if\quad s\le 0\end{cases} \tag{5.10}$$

where μ denotes a shift parameter and $\gamma > 0$ is a scale parameter

Lévy distribution described in terms of Fourier transform as follows [145] ,

$$F(k)=\exp[-\alpha|k|^{\beta}],\quad 0<\beta\le 2, \tag{5.11}$$

where α is skewness parameter or scale factor and β is Lévy index.

5.4 Modified Grasshopper Optimization Algorithm (MGOA)

Generally, for any population based metaheuristic algorithm to achieve an optimum performance a proper balance between exploitation and exploration of the search space is necessary. Although GOA is an efficient metaheuristic algorithm, it has some drawbacks. In the conventional GOA algorithm, the previous position of grasshopper, the positions remaining grasshoppers in the population and target grasshopper position determines the new position of any grasshopper [90]. However, grasshoppers get clustered around local optima after a certain number of iterations and it leads to loss of swarm diversity and the ability of algorithm to explore the solution space is deteriorated. It may cause algorithm to converge prematurely to a local solution. Another concern with GOA is that there is no provision to retain the best-so-far solution at each grasshopper index. This issue slow down the convergence of GOA and degrades the exploitation capability.

To overcome these shortcomings this chapter proposes a modified grasshopper optimization algorithm (MGOA) by incorporating three modifications into GOA algorithm. The Lévy flight is integrated with basic GOA to preserve the diversity of grasshopper swarm. Second, the threshold parameter is introduced to identify the inefficient search spaces and to redistribute the grasshoppers using LF. Lastly, the greedy selection operator from DE algorithm is employed to retain the better performing grasshoppers obtained so far at each index which provides rapid convergence to the global optimum.

In the proposed MGOA algorithm, a threshold parameter introduced to determine whether the current region of search space is efficient, which in turn helps in detecting stagnation at local optima condition. Therefore, a threshold value is determined and if target grasshopper fitness does not improve at the end of each iteration or if the improvement in fitness function is less than some predefined number, this threshold value is increased by 1. If the calculated threshold value reaches a predetermined value, the Lévy flight is used to redistribute the entire swarm of grasshoppers in the search space. When the threshold parameter reaches its predetermined value, this indicates that the current solution space is inefficient with a local optimum solution, which in turn shows the stagnation at local optima condition. Thus, the LF assists GOA in preserving the swarm diversity and enhancing global search ability to avoid entrapment in local optima.

The Lévy flight is used to generate new grasshoppers as follows [163] :

$$X^{new} = X + \alpha \oplus L\acute{e}vy(\beta) \tag{5.12}$$

Where α denotes the step size and considered as a random number for every dimension

$$X^{new} = X + random(size(K)) \oplus L\acute{e}vy(\beta) \tag{5.13}$$

The Ref. [163,164] [152]contains the scheme to create step size s , which is given by,

$$s = random(size(K)) \oplus L\acute{e}vy(\beta) \sim 0.01 \frac{u}{|v|^{1/\beta}} (X - X_{best}) \tag{5.14}$$

Where K denotes the number of dimensions, u and v follows a normal distribution, i.e. $u \sim N(0, \sigma_u^2),\ v \sim N(0, \sigma_v^2)$, Where σ_u and σ_v are taken as follows:

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta)\sin\left(\frac{\pi\beta}{2}\right)}{\Gamma\left[\left(\frac{1+\beta}{2}\right)\right]\beta 2^{(\beta-1)/2}} \right\}^{1/\beta} \quad , \quad \sigma_v = 1 \tag{5.15}$$

Where Γ denotes a Gamma function

To perform LF the value of s is added to the previous solution i.e. X which results in X^{new} as per Eq. (5.13). The fitness values of all updated grasshoppers are evaluated and the target is identified.

In basic GOA, there is no provision to retain the elite grasshopper found so far at each population index. To alleviate this issue, a greedy selection operator from Differential evolution algorithm [131] is used in the modified grasshopper optimization. The operator can be defined as follows:

$$X(it+1) = \begin{cases} X^{new} \;\; if \;\; f(X^{new}) \leq f(X) \\ X \qquad\qquad otherwise \end{cases} \tag{5.16}$$

As per this operator, the newly generated grasshopper position at grasshopper index *i* i.e. X_i^{new} will be accepted only if its fitness F_i^{new} is better than or equal to the fitness of old/previous grasshopper at same index (F_i) else X_i will be preserved in the swarm. Thus, this concept results in retaining the best grasshopper found so far at each index and improves the exploitation capability and convergence of the algorithm.

In the proposed MGOA algorithm, a predetermined threshold values is initialized and count is initialized to zero. The swarm of grasshoppers is generated and the fitness evaluation for each grasshopper is performed. The fitness of initially generated grasshoppers is used to determine the target grasshopper. After this the count values is checked for target and if the count reaches the predetermined value of threshold parameter, positions of grasshoppers are updated using LF else per Eq. (18) is used to update the grasshoppers positions. After updating the positions of grasshoppers, the fitness evaluation is performed for all the grasshoppers in swarm. The greedy selection scheme from DE is used to retain better grasshopper among previous grasshopper X and the newly generated grasshopper X^{new} and as per updated fitness values, the target is identified. If the target grasshopper fitness is improved the count is set to zero else if target fitness is not improved or improvement is less than some predefined number, the count is incremented by 1. These steps are repeated till termination criterion is satisfied. The flowchart of MGOA algorithm is shown in Fig. 5.1.

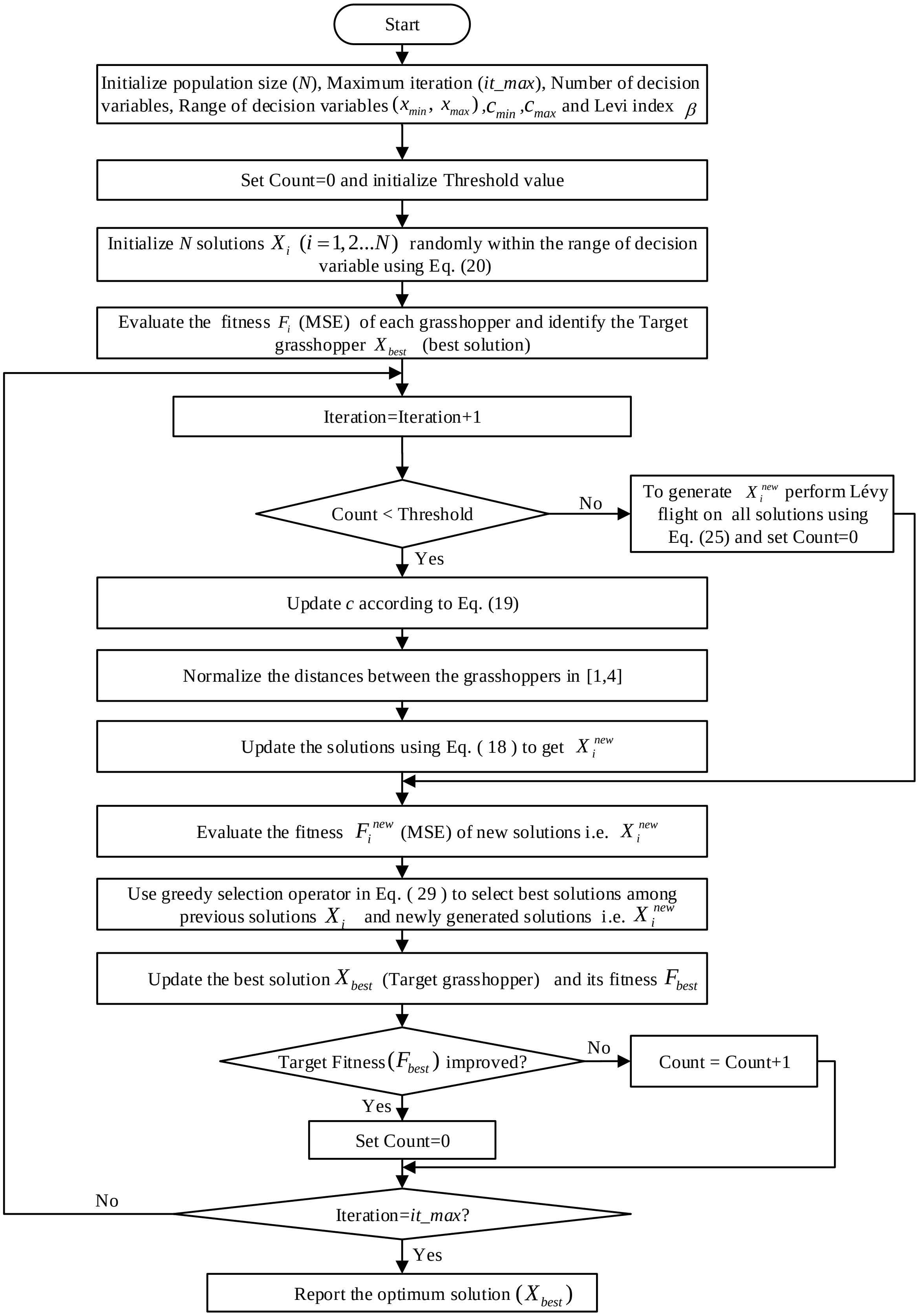


Fig. 5. 1. Flowchart of MGOA algorithm

5.5 Non-linear channel equalization using modified grasshopper optimization algorithm

In this section, the step by steps procedure to train FLANN based non-linear channel equalizer using the proposed modified grasshopper optimization algorithm is provided.

The steps are explained below:

Step 1: Compute the corrupted output of multi-path wireless channel $r(n)$ as per Eq. (2.3)

Step 2: Expand the input vector of FLANN $R(n)$ as per Eq. (2.8) to get its enhanced version $R^e(n)$

Step 3: Initialize N grasshoppers within upper and lower boundary of search space (i.e. within x_{min} and x_{max}). Furthermore, set count equal to zero and initialize threshold value.

Step 4: Determine the FLANN non-linear equalizer output $y(n)$ as per Eq. (2.12) and use this output for error calculation as per Eq (2.13). Furthermore, compute the fitness function (MSE) for grasshoppers as given below:

$$MSE\left(X\right)=\frac{1}{S}\sum_{n=1}^{S}e^2(n) \tag{5.17}$$

Step 5: If the count has reached the threshold value then perform LF on all grasshoppers using Eq. (5.13) to get X_i^{new} else update the grasshoppers using Eq. (5.7) to obtain X_i^{new}.

Step 6: Evaluate the fitness of all newly generated grasshoppers F_i^{new} and choose the better grasshopper at each population index using greedy selection operator of DE.

Step 6: Update the target grasshopper and its fitness.

Step 7: If there is no improvement in (in fitness of target grasshopper) MSE value or improvement in MSE is less some predefined value then increment the count by 1 else set count equal to zero.

Step 8: If the total iterations are reached then report the target grasshopper X_{best} and stop else continue repeating the steps 4-7.

5.6 Simulation Experiments

The simulations have been performed on PC with 8 GB RAM in a MATLAB R2013a.

5.6.1 Channels considered for simultions

The performance of a modified grasshopper optimization algorithms has been evaluated over four wireless communication channels. The corresponding channels are taken from the Refs. [11,19,37,122,137,165–167] and can be written as :

$$
\begin{aligned}
H_1(\mathrm{z}) &= 0.209 + 0.995z^{-1} + 0.209z^{-2} \quad &&: \text{Channel } 1 \\
H_2(\mathrm{z}) &= 0.260 + 0.930z^{-1} + 0.260z^{-2} \quad &&: \text{Channel } 2 \\
H_3(\mathrm{z}) &= 0.304 + 0.903z^{-1} + 0.304z^{-2} \quad &&: \text{Channel } 3 \\
H_4(\mathrm{z}) &= 0.341 + 0.876z^{-1} + 0.341z^{-2} \quad &&: \text{Channel } 4
\end{aligned} \tag{5.18}
$$

Furthermore, the following nonlinearities have been considered to introduce the non-linear distortion to the channel output and are taken from the references [19,37,137,165,167].

$$
\begin{aligned}
b(n) &= t(n) &&: \text{NL=0} \\
b(n) &= t(n) + 0.2\,t^2(n) - 0.1\,t^3(n) &&: \text{NL=1} \\
b(n) &= t(n) + 0.2t^2(n) - 0.1t^3(n) + 0.5\cos(\pi t(n)) &&: \text{NL=2}
\end{aligned} \tag{5.19}
$$

5.6.2 Sensitivity Analysis of the MGOA algorithm

In this section, simulation experiments are performed to analyze the sensitivity of MGOA to its key parameters. The non-linear channel with channel 2 and nonlinearity NL=1 is taken for simulation. The simulations are conducted to choose optimal values for number of grasshoppers (*N*), index of Levy distribution (β), data block size (S) and number of taps of equalizer (*M)* and the corresponding results are provided in Tables 5.1 to 5.4 and Figs. 5.2 to 5.5.

The effect of number of grasshopper on MSE is analysed by varying N from 5 to 35 and Fig. 5.2 and Table 5.1 shows that N=25 is sufficient to achieve optimum results. Moreover, Fig 5.3 and Table 5.2 illustrates that there is not significant reduction in MSE for values of the block size more than 200. It is clear from Fig 5.4 and Table 5.3 that the minimum MSE is obtained when number of taps are 4. Finally the index of Levy distribution is varied from 0.25 to 2 and Fig 5.5 and Table 5.4 depicts that β equal to 0.5 provides the least MSE.

Table 5. 1 Effect of number of grasshoppers (N) on MSE

Population Size (N)	MSE				
	Best	Worst	Median	Mean	Std. Dev.
5	0.0020	0.6169	0.0577	0.1546	0.1731
10	5.5532e-04	0.1310	0.0018	0.0098	0.0261
15	4.3734e-04	0.0018	7.8300e-04	8.4163e-04	3.2790e-04
20	3.9065e-04	0.0012	4.6128e-04	5.0282e-04	1.4760e-04
25	3.6636e-04	7.1749e-04	4.2504e-04	**4.4895e-04**	**8.8639e-05**
30	3.5731e-04	5.6167e-04	4.3886e-04	4.4066e-04	5.1189e-05
35	3.6030e-04	5.0784e-04	4.3951e-04s	4.3621e-04	4.3033e-05

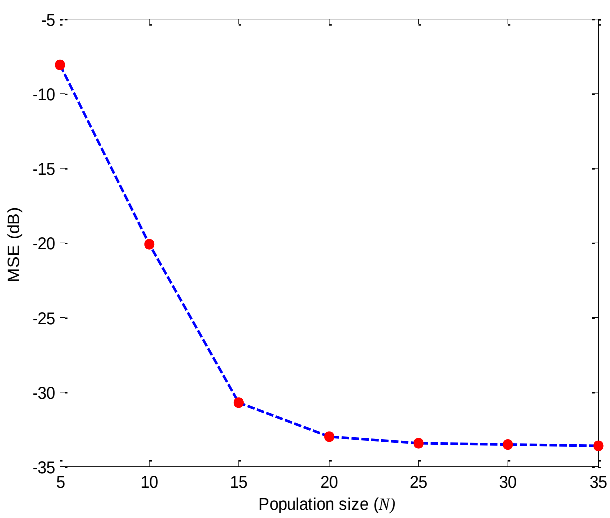


Fig. 5. 2. Effect of number of grasshoppers (N) on MSE

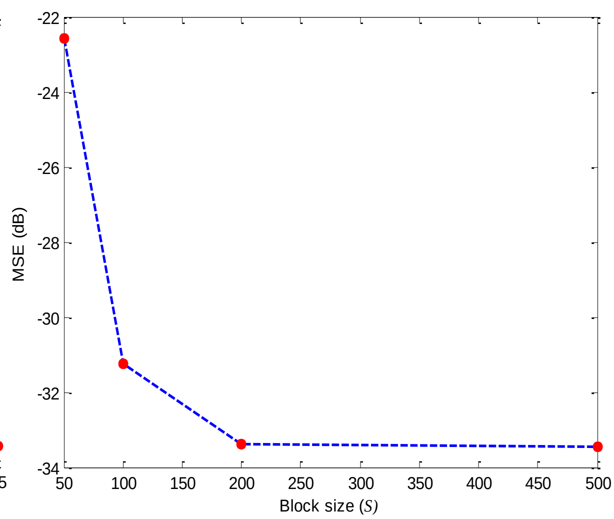


Fig. 5. 3. Effect for variation of data block size (S)

Table 5. 2 Effect for variation block size (S)

Block Size (S)	MSE				
	Best	Worst	Median	Mean	Std. Dev.
50	0.0051	0.0065	0.0055	0.0055	3.3926e-04
100	5.9412e-04	0.0011	7.2123e-04	7.4893e-04	1.3049e-04
200	3.6806e-04	7.1083e-04	4.4784e-04	**4.5785e-04**	**7.7255e-05**
500	3.6626e-04	6.1446e-04	4.3333e-04	4.5026e-04	6.6393e-05

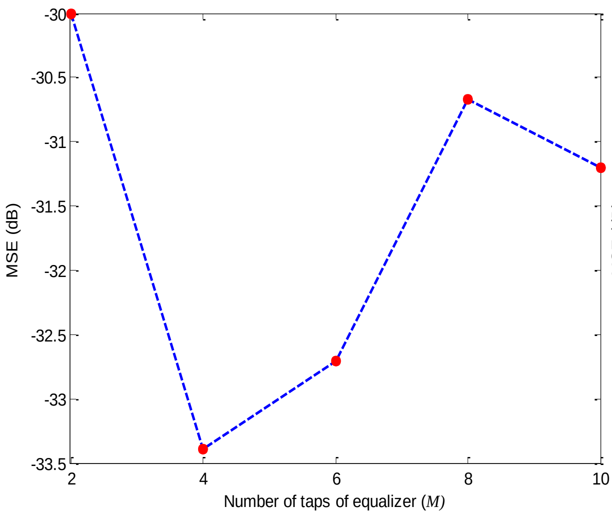


Fig. 5. 4. Effect of M on MSE

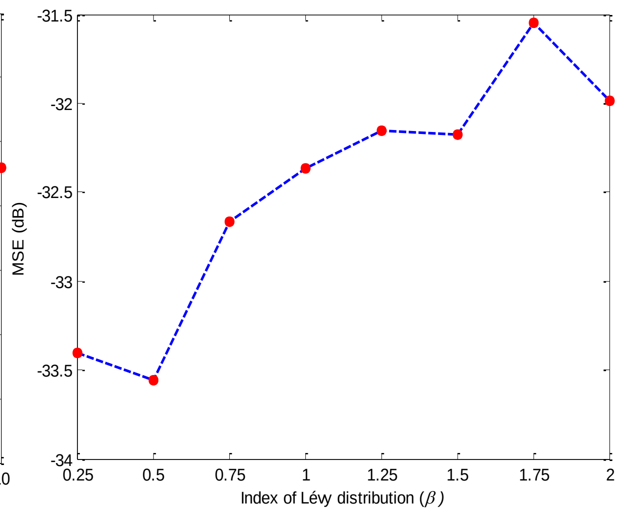


Fig. 5. 5. Effect of variation of Lévy index (β)

Table 5. 3 Effect of variation of *M*

No. of taps (M)	MSE				
	Best	Worst	Median	Mean	Std. Dev.
2	0.0010	0.0011	0.0010	0.0010	1.0435e-05
4	3.6806e-04	7.1083e-04	4.4784e-04	**4.5785e-04**	7.7255e-05
6	3.7767e-04	8.1960e-04	5.0993e-04	5.3623e-04	1.0880e-04
8	3.9989e-04	0.0016	7.7578e-04	8.5611e-04	3.2463e-04
10	4.0619e-04	0.0015	6.9439e-04	7.5745e-04	2.5878e-04

Table 5. 4 Effect of variation of Lévy index (β)

Lévy index (β)	MSE				
	Best	Worst	Median	Mean	Std. Dev.
0.25	3.6401e-04	5.8347e-04	4.4080e-04	4.5634e-04	6.4029e-05
0.5	3.5522e-04	7.5049e-04	4.2501e-04	**4.4102e-04**	8.4777e-05
0.75	3.7274e-04	9.1180e-04	5.3361e-04	5.4136e-04	1.3805e-04
1	4.2071e-04	8.8191e-04	5.4122e-04	5.8007e-04	1.3233e-04
1.25	4.1063e-04	0.0013	5.8038e-04	6.0923e-04	1.6423e-04
1.5	4.0266e-04	0.0010	5.7165e-04	6.0607e-04	1.5145e-04
1.75	4.1016e-04	0.0016	6.3696e-04	7.0080e-04	2.3311e-04
2	4.0771e-04	9.5361e-04	6.0821e-04	6.3288e-04	1.5455e-04

5.6.3 Performance Analysis of MGOA algorithm for channel Equalization

The performance of the proposed modified grasshopper optimization algorithm has been compared with GOA, sine cosine algorithm (SCA) and rat swarm optimizer (RSO) [168] algorithm for equalization of non-linear wireless communication channels. The training process of FLANN based non-linear equalizer has been performed for 500 iterations with block size of 200. Furthermore, to analyse the performance of modified grasshopper optimization algorithm for equalization of non-linear channels, the signal considered for transmission takes values either +1 or -1 with uniform distribution. The signal to noise ratio considered for AWGN introduced to wireless channel output is 20 dB. Table 5.5 provides the parameters taken for simulations.

5.6.3.1 MSE performance

In this section, the learning performance of the algorithms has been demonstrated over 4 non-linear wireless communication channels and results have been averaged over 30 runs.

Table 5. 5 Simulation Parameters

Algorithm	Parameter	Value
MGOA	Population size (N)	25
	Index of Lévy distribution (β)	0.5
GOA	Population size (N)	25
MFO	Population size (N)	25
	Random number (t)	[-1,1]
	b	1
SCA	Population size (N)	25
	r_1	*rand*
	r_2	2**pi*rand*
	r_3	2**rand*
	r_4	*rand*
	a	2

Non-linearity 1 (NL=1)

To analyse the MSE performance, this case considers a non-liner distortion from the non-linearity NL=1 (Eq. (33)) and the corresponding MSE curves are shown in Fig. 5.6. As evident from this figure, MGOA provides the lesser MSE and faster convergence over GOA, SCA and RSO algorithm. Moreover, MSE results in Table 5.6 shows the equalization capability of MGO based equalizer. Box plot diagrams [139] are used to show the distribution of data and the MSE box plots in Fig 5.7. confirms the superiority of MGOA from least values of the median and interquartile range achieved by it.

Table 5. 6 MSE results with non-linearity NL=1 for all four communication channels

Channel	Algorithms	MSE			
		Best	Worst	Mean	SD
Channel 1	MGOA	8.0652e-05	1.4920e-04	**1.0236e-04**	**1.5904e-05**
	GOA	8.4487e-05	7.0463e-04	2.0843e-04	1.1986e-04
	SCA	2.1634e-04	0.0027	6.6597e-04	5.1216e-04
	RSO	2.5196e-04	0.0124	0.0036	0.0037
Channel 2	MGOA	3.5522e-04	7.5049e-04	**4.4102e-04**	**8.4777e-05**
	GOA	5.0671e-04	0.0016	8.0838e-04	2.4882e-04
	SCA	7.8778e-04	0.0031	1.4250e-03	6.0215e-04
	RSO	8.6449e-04	0.0217	0.0059	0.0059
Channel 3	MGOA	0.0014	0.0021	**0.0016**	**1.4391e-04**
	GOA	0.0016	0.0050	0.0025	8.4880e-04
	SCA	0.0025	0.0099	0.0045	0.0019
	RSO	0.0026	0.0407	0.0163	0.0119
Channel 4	MGOA	0.0058	0.0082	**0.0063**	**6.5229e-04**
	GOA	0.0059	0.0128	0.0086	0.0016
	SCA	0.0079	0.0213	0.0141	0.0034
	RSO	0.0094	0.1620	0.0422	0.0323

Non-linearity 2 (NL=2)

This case takes into consideration a severe non-linear scenario with non-linearity NL=2 . The MSE convergence curves of all the algorithms with non-linearity NL=2 for four different non-linear channels is demonstrated in Fig. 5.8. This figure shows that MGOA escapes from local optima offering a higher convergence speed. As a result of which MGOA is able to achieve the lowest MSE among all the algorithms. Furthermore, MSE results in Table 5.7 validates the efficiency of the MGOA based equalizer. Figs. 5.6 and 5.8 illustrate that an increase in non-linearity (from NL=1 to NL=2) does not have any significant effect on the superiority of the proposed MGO when compared to others. Moreover, the MSE box plots in Fig. 5.9 further validates the superiority of MGOA.

Table 5. 7 MSE results with non-linearity NL=2 for all four communication channels

Channel	Algorithm	MSE			
		Best	Worst	Mean	SD
Channel 1	MGOA	0.0011	0.0024	**0.0013**	**2.6125e-04**
	GOA	0.0011	0.0054	0.0023	0.0012
	SCA	0.0021	0.0058	0.0039	9.9434e-04
	RSO	0.0048	0.0394	0.0116	0.0076
Channel 2	MGOA	0.0032	0.0052	**0.0039**	**5.1607e-04**
	GOA	0.0035	0.0132	0.0068	0.0025
	SCA	0.0065	0.0169	0.0092	0.0025
	RSO	0.0086	0.1002	0.0215	0.0200
Channel 3	MGOA	0.0124	0.0191	**0.0143**	**0.0021**
	GOA	0.0127	0.0334	0.0206	0.0055
	SCA	0.0177	0.0449	0.0283	0.0063
	RSO	0.0270	0.1422	0.0571	0.0296
Channel 4	MGOA	0.0413	0.0549	**0.0449**	**0.0029**
	GOA	0.0413	0.0820	0.0552	0.0099
	SCA	0.0546	0.1294	0.0732	0.0158
	RSO	0.0873	0.2280	0.1367	0.0332

Statistical Analysis:

In this section, a Wilcoxon’s rank-sum test [169] is conducted at 5% significance level and corresponding *p* values are noted in Table 5.8. As can be noticed from results, *p*-values obtained for competing algorithms are very much less than 0.05 which indicates the statistical significance of the results obtained by MGOA based equalizer.

Table 5. 8 Results of Wilcoxon rank-sum test for MSE results of Table 5.6 and Table 5.7

Channel	Nonlinearity	Algorithms			
		MGOA	GOA	SCA	RSO
Channel 1	NL=1	N/A	1.6980e-08	3.0199e-11	3.0199e-11
	NL=2	N/A	5.0912e-06	3.3384e-11	3.0199e-11
Channel 2	NL=1	N/A	2.6099e-10	3.0199e-11	3.0199e-11
	NL=2	N/A	1.1023e-08	3.0199e-11	3.0199e-11
Channel 3	NL=1	N/A	1.6947e-09	3.0199e-11	3.0199e-11
	NL=2	N/A	3.0103e-07	4.5043e-11	3.0199e-11
Channel 4	NL=1	N/A	7.7725e-09	3.3384e-11	3.0199e-11
	NL=2	N/A	8.8411e-07	3.3384e-11	3.0199e-11

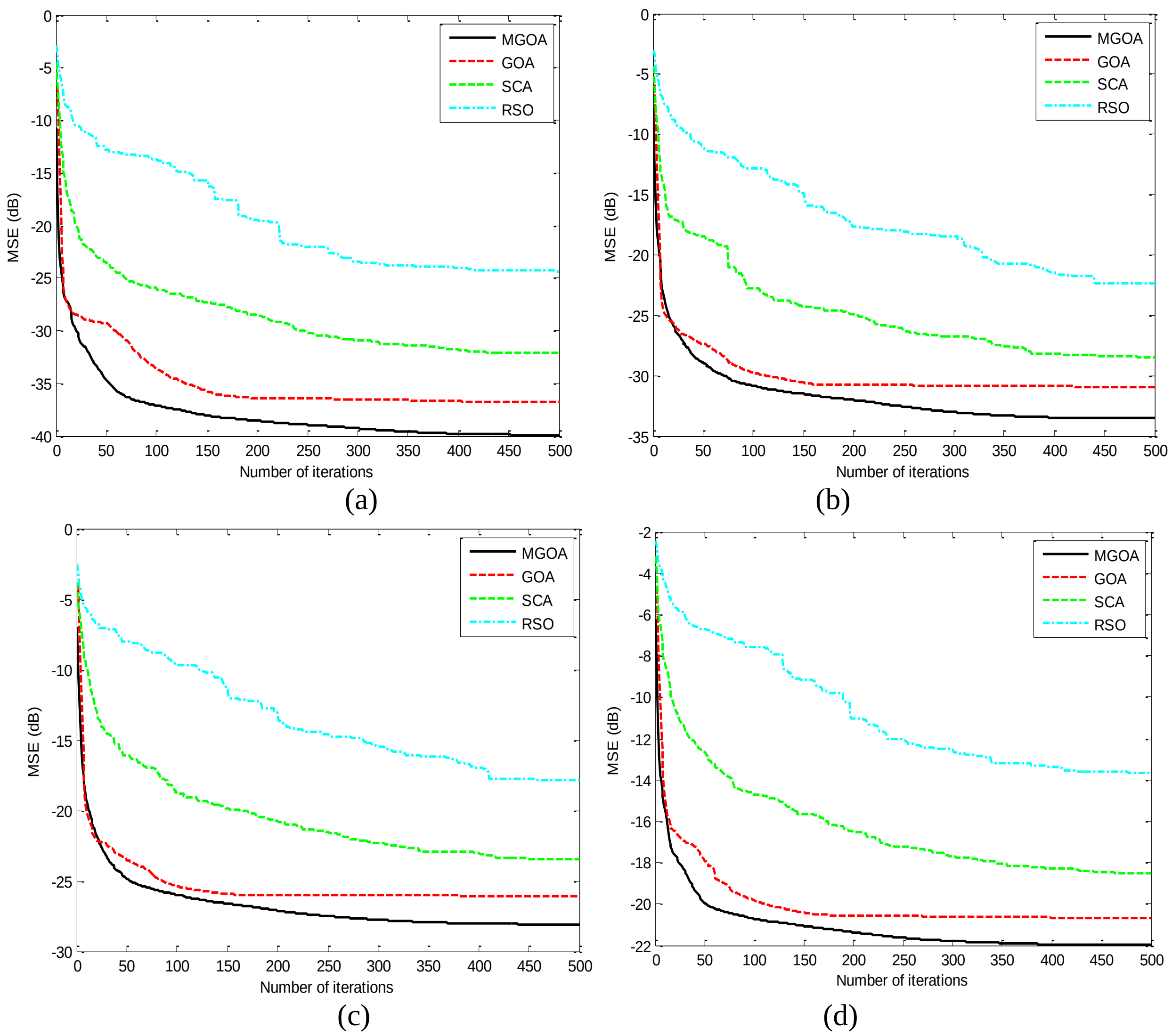


Fig. 5. 6. Learning curves with non-linearity NL=1 for (a) Channel 1 (b) Channel 2 (c) Channel 3 and (d) Channel 4

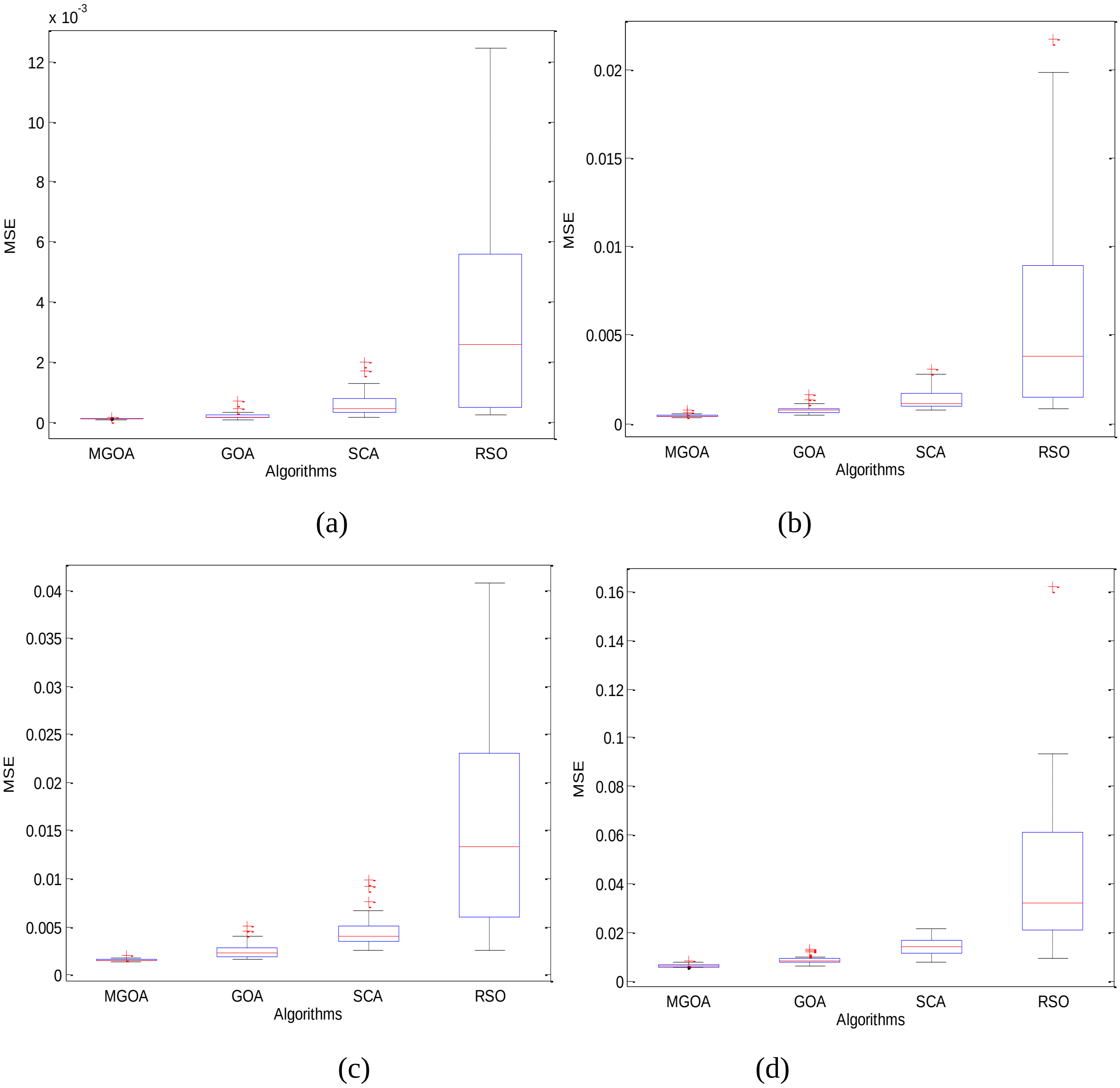


Fig. 5. 7**.** MSE Box plots with non-linearity NL=1 for (a) Channel 1 (b) Channel 2 (c) Channel 3 (d) Channel 4

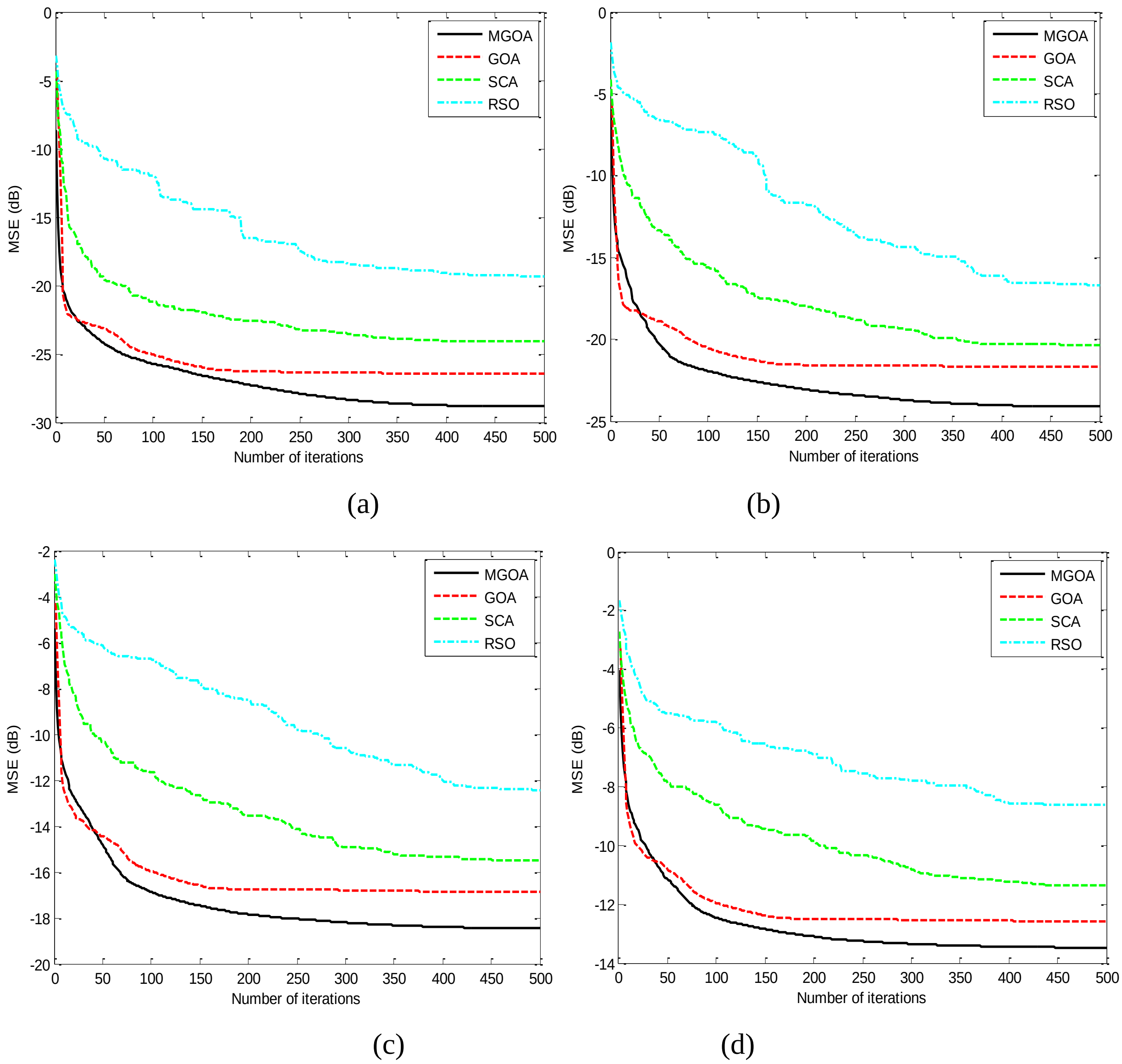


Fig. 5. 8. Learning curves with non-linearity NL=2 for (a) Channel 1 (b) Channel 2 (c) Channel 3 and (d) Channel 4

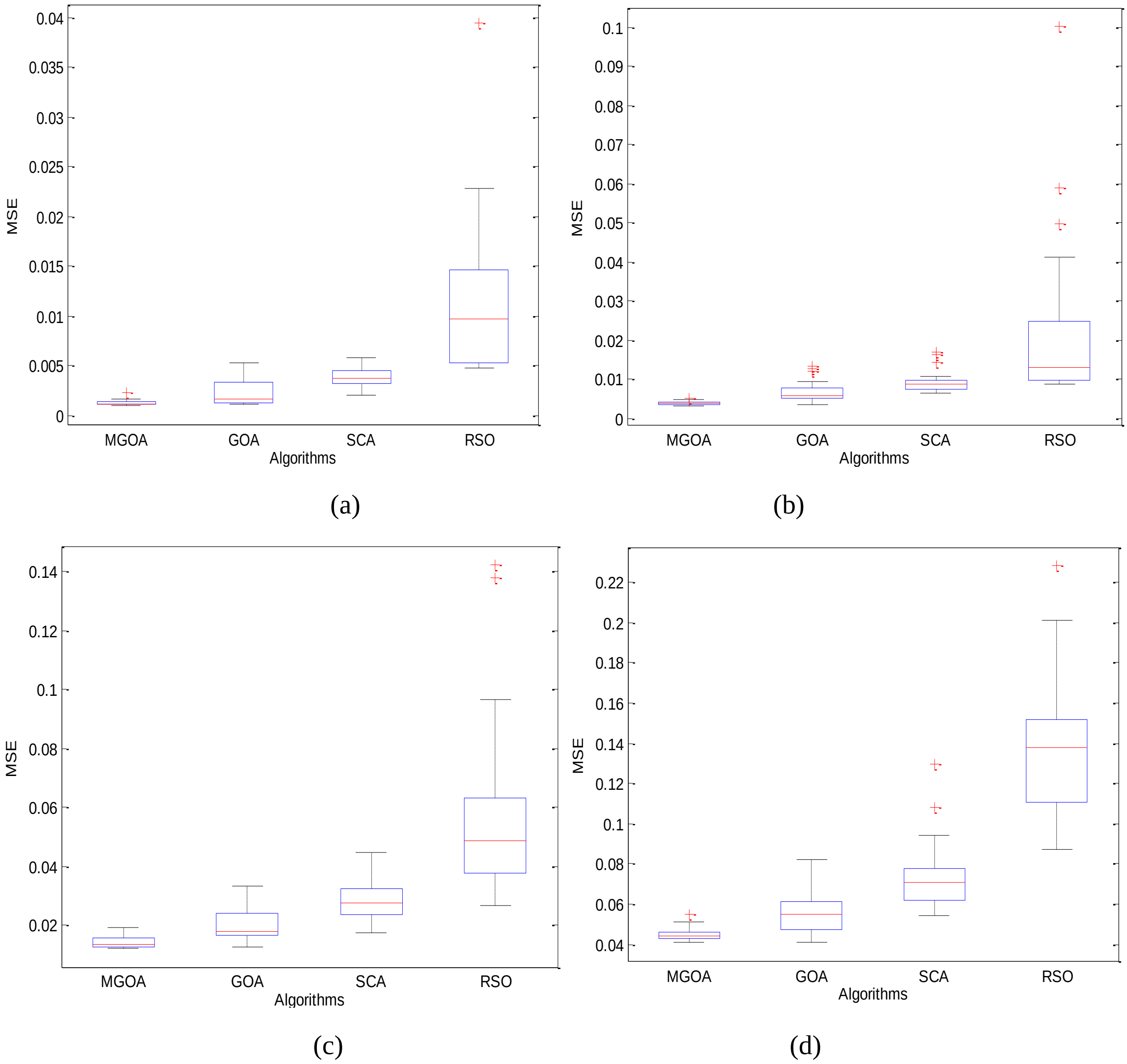


Fig. 5. 9. MSE Box plots with non-linearity NL=2 for (a) channel 1 (b) channel 2 (c) channel 3 and (d) channel 4

5.6.3.2 BER performance

The BER performance of the proposed modified grasshopper optimization algorithm has been evaluated in this section over 4 non-linear channels.

Non-linearity 1 (NL=1)

The effect of non-linearity NL=1 on the BER performance of MGO and other algorithms is examined and plotted in Fig. 5.10. Approximately 1.25 dB improvement in SNR is achieved by MGOA over GOA algorithm for channel 1 at BER of 1.117e-04 and more than 2 dB for the remaining algorithms. Moreover, MGOA provides similar BER performance for Channel 2, Channel 3 and Channel 4.

Non-linearity 2 (NL=2)

Fig. 5.11 shows the effect of non-linear distortion from the non-linearity NL=2 on the BER performance of all algorithms. As can be seen from Fig. 5.11(a) a gain of about 1.3 dB is made by the MGOA when compared to the GOA and more than 2 dB in comparison with other algorithms at a BER of 1.89e-04. Furthermore, Figs. 5.10 and 5.11 demonstrate that despite increase in non-linearity from NL=1 to NL=2 MGOA is perfroming consistently.

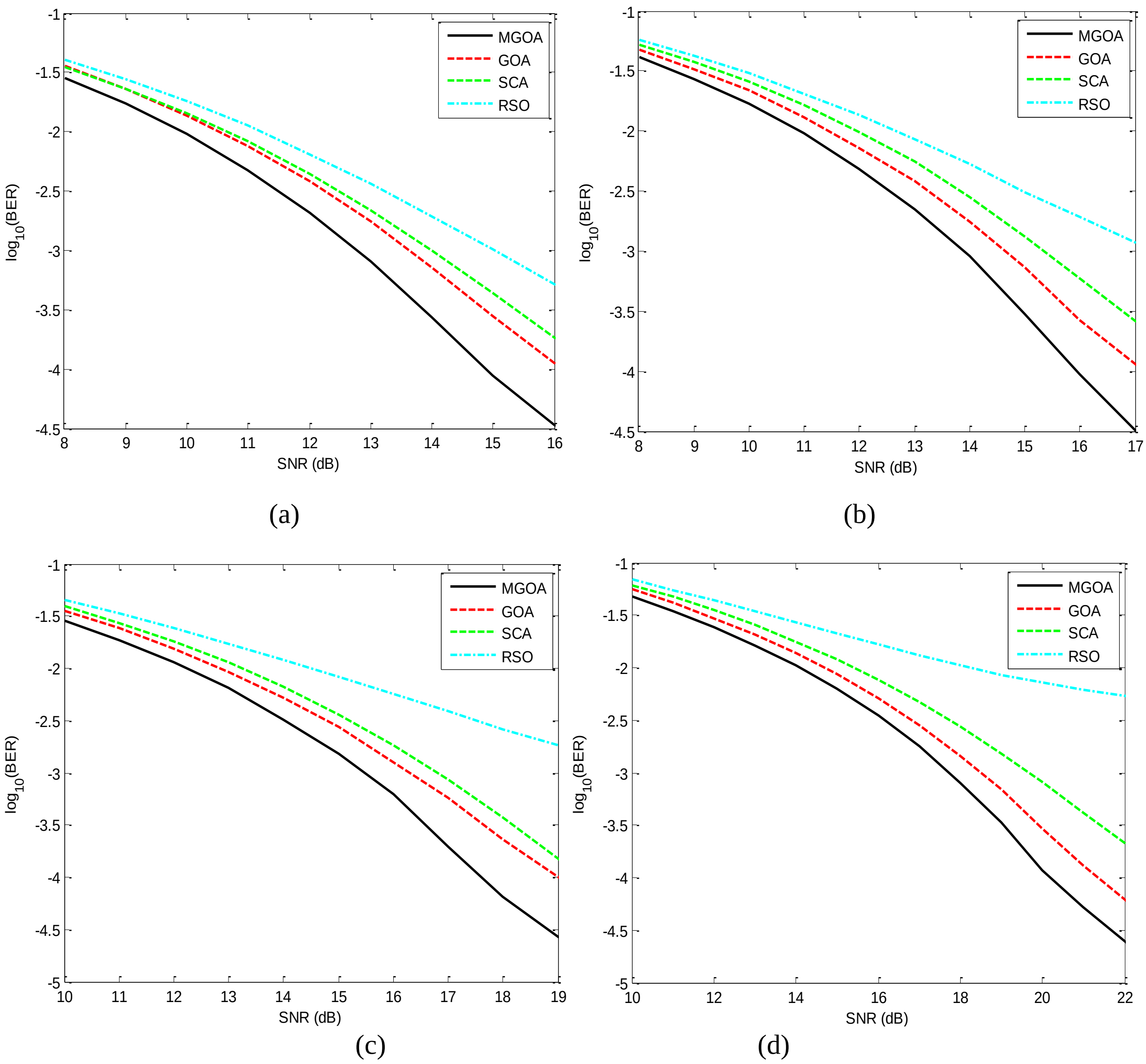
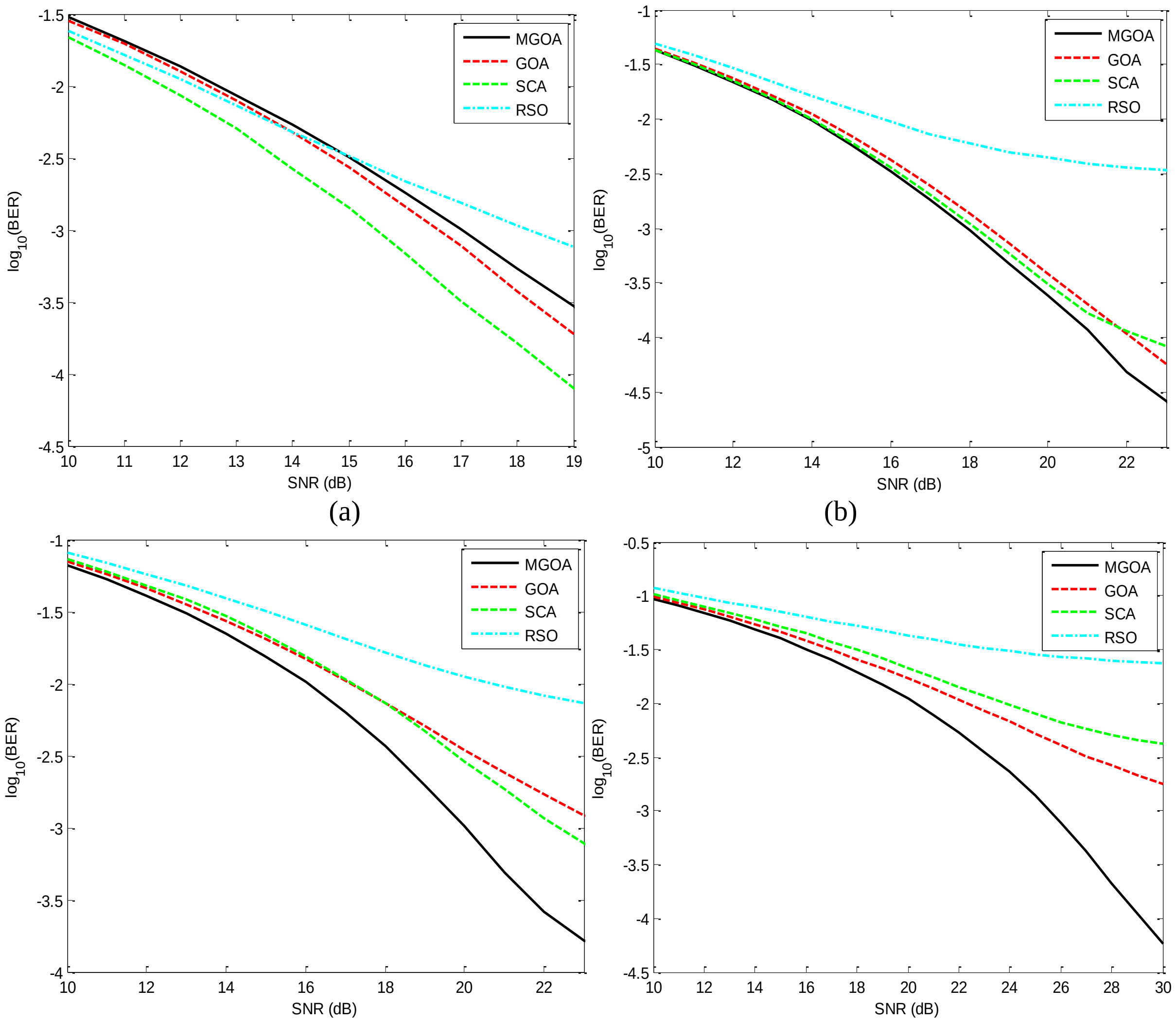


Fig. 5. 10**.** BER performance with non-linearity NL=1 for (a) channel 1 (b) channel 2 (c) channel 3 (d) channel 4



(c) (d)

Fig. 5. 11. BER performance with non-linearity NL=2 for (a) channel 1 (b) channel 2 (c) channel 3 (d) channel 4

5.7 Conclusion

In this chapter, a Modified Grasshopper Optimization Algorithm (MGOA) is proposed for equalization of non-linear wirless communication channels. The superiority of the proposed MGOA based equalizer is illustrated over other metaheuristic algorithms such as GOA, SCA and RSO . The simulation results on four non-linear channels demonstrate the efficiency of the proposed MGOA algorithm in terms of BER and MSE. Furthermore, the statistical validity of the results provided by MGOA is confirmed through Wilcoxon rank-sum test.

Chapter 6

A neural network based approach for joint channel estimation and equalization of Universal Filtered Multi-Carrier (UFMC) System

6.1 Introduction

In this chapter, we have proposed a joint channel estimation and data detection for Universal Filtered Multi-carrier (UFMC) system using deep feed-forward neural network. In conventional universal filtered multicarrier system to recover the transmitted symbols, channel estimation and data detection are carried out as a two different processes. However, the conventional approaches involves the use of pilots for every frame. Furthermore, in the pilot based channel estimation approach it requires to explicitly model the channel using the available channel observations which may not be accurate always. To overcome these drawbacks, this work proposes a joint channel estimation and data detection approach for UFMC using deep neural network. The proposed approach provides better BER performance when compared to existing methods.

6.2 Related Work:

Universal Filtered Multi-carrier (UFMC) system has been proposed in the year 2013 [170]. The benefits of Filter bank Multicarrier (FBMC) and orthogonal frequency division multiplexing (OFDM) are integrated in UFMC [171]. The Ref. [172] contains comprehensive comparison of UFMC, OFDM and FBMC . The pilot aided channel estimation for UFMC is carried out in [171] using the conventional channel estimation methods used for channel estimation in OFDM system. Furthermore, channel estimation under AWGN and Rayleigh fading channel have been carried out in [173] for different QAM modulated signals. Moreover, the channel estimation for power domain NOMA-UFMC have been performed in [174]. However, conventional channel estimation techniques involves estimation of channel coefficients to recover the transmitted data. Therefore, the recent use of deep neural network in wireless communication system by various researchers [175–179] motivated us to use neural network based approach for joint channel estimation and data detection in UFMC system.

6.3 Universal Filtered Multi-carrier (UFMC) System

Fig. 6.1 shows the system model for universal filtered multicarrier system. The description of UFMC system in this section is taken from [180][172] [181]. In UFMC system, *B* sub bands are used to divide the total available wireless channel bandwidth. If there are total *N* subcarriers then each sub-band consist of *N/B* subcarriers. The data symbols which need to be transmitted are first mapped to 4 QAM signal and then converted from serial to parallel form to perform the IFFT operation. The output of IFFT block is given to FIR filter c_i (i=1,2,3……*B*). The filtered output o_i of i^{th} sub-band is given as follows [180]:

$$\begin{aligned} o_i(n) &= c_i * x_i \\ &= \sum_{m=0}^{L-1} c_i(\mathrm{m}) * x_i(n-m) \quad n = 0,1,2....,N+L-1 \end{aligned} \tag{6.1}$$

The filtered output of all *B* sub-bands is summed up as follows [180]:

$$r\,(n) = \sum_{i=1}^{B} o_i(n) \tag{6.2}$$

Finally, the UFMC signal is transmitted through a wireless channel of impulse response *h* and the received signal *y* is given by [180],

$$y(n) = r(n) * h(n) + q(n) \tag{6.3}$$

$$y(n) = \left(\sum_{i=1}^{B} c_i * x_i \right) * h(n) + q(n) \tag{6.4}$$

Where *h(n)* denotes the channel impulse response and *q(n)* is noise

At the receiver, zero padding is done on the received UFMC signal to perform the 2*N*-point FFT operation. The signal after performing 2N-point FFT operation is given by [180],

$$Y(k) = \frac{1}{\sqrt{N}} \sum_{m=0}^{N+L-2} y(m) e^{-j2\pi mn/2N}, \quad k=0,1,....2N\text{-}1 \tag{6.5}$$

If single tap channel is considered the Eq. (6.5) can be written as [180]:

$$Y(n) = H(n)\sum_{i=1}^{B}\tilde{X}_i(n)Ci(n) + Q(n) \tag{6.6}$$

Where $H(n)$ is $2N$ point FFT of $h(n)$

$Q(n)$ is 2N point FFT of additive white Gaussian noise $q(n)$

$\tilde{X}_i(n)$ is the 2N point FFT of x_i

After 2N-point FFT operation channel estimation is performed using the pilots and the estimated channel coefficients are used to detect the data.

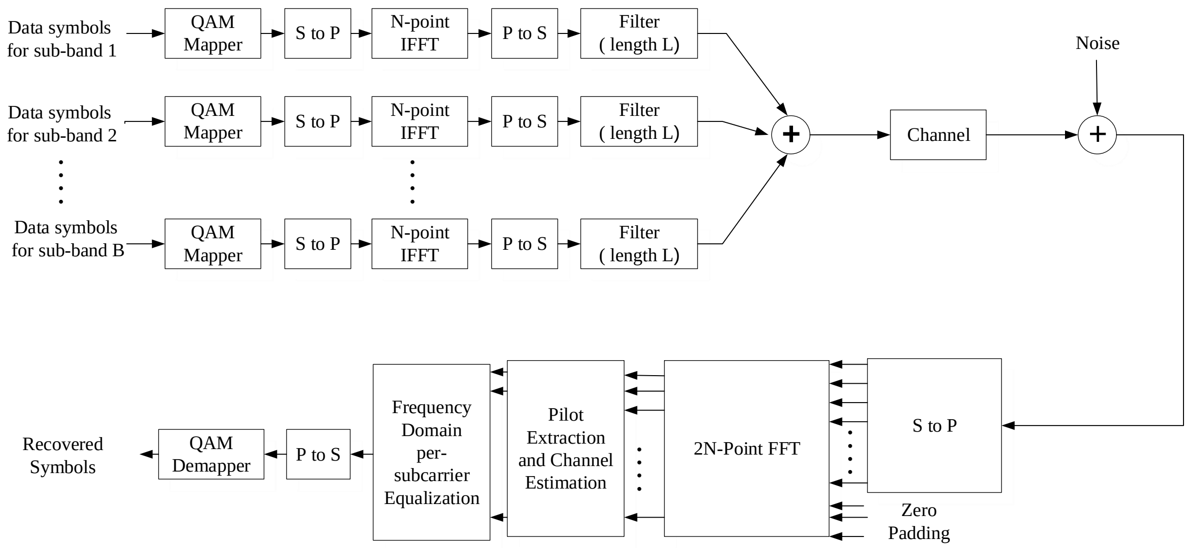


Fig. 6. 1 UFMC System [180][172]

6.4 Neural Networks

Neural networks are the supervised machine learning models and have demonstrated interesting results in many engineering applications [22][23] [24]. The detailed description of neural networks can be found in the Refs. [182][25][27]. The basic structure of deep neural network used in this study is shown in Fig. 6.2.

6.5 Neural Network based approach for joint channel estimation and data detection in UFMC system

In multicarrier system to recover the transmitted symbols, channel estimation and data detection are carried out as a two different processes. In traditional channel estimation approaches such as least square (LS) and Minimum mean square error (MMSE) estimation, the transmitted symbols are recovered by using the estimates of the channel coefficients. However, the process of channel estimation need the transmission of pilots along with the data. In this study, the joint channel estimation and data detection are carried out jointly using the deep neural network. The deep neural network is trained offline with channel observations as training data and after completion of training the trained model is utilized directly to detect the transmitted symbols. Fig. 6.2 shows the proposed approach for joint channel estimation and data detection in UFMC system.

During the training process of deep neural network, we have used Mean square error (MSE) as the cost function and which is given as follows:

$$MSE = \frac{1}{N}\sum_{j=1}^{N}\left(\hat{X}(n) - X(n)\right)^2 \tag{6.8}$$

Where $\hat{X}(n)$ denotes the estimate of the transmitted symbol recovered by the neural network and $X(n)$ is the actual transmitted symbol.

After completion of training the neural network model is deployed for data detection. It is noted that once the training process of DNN gets completed it does not need the training symbols.

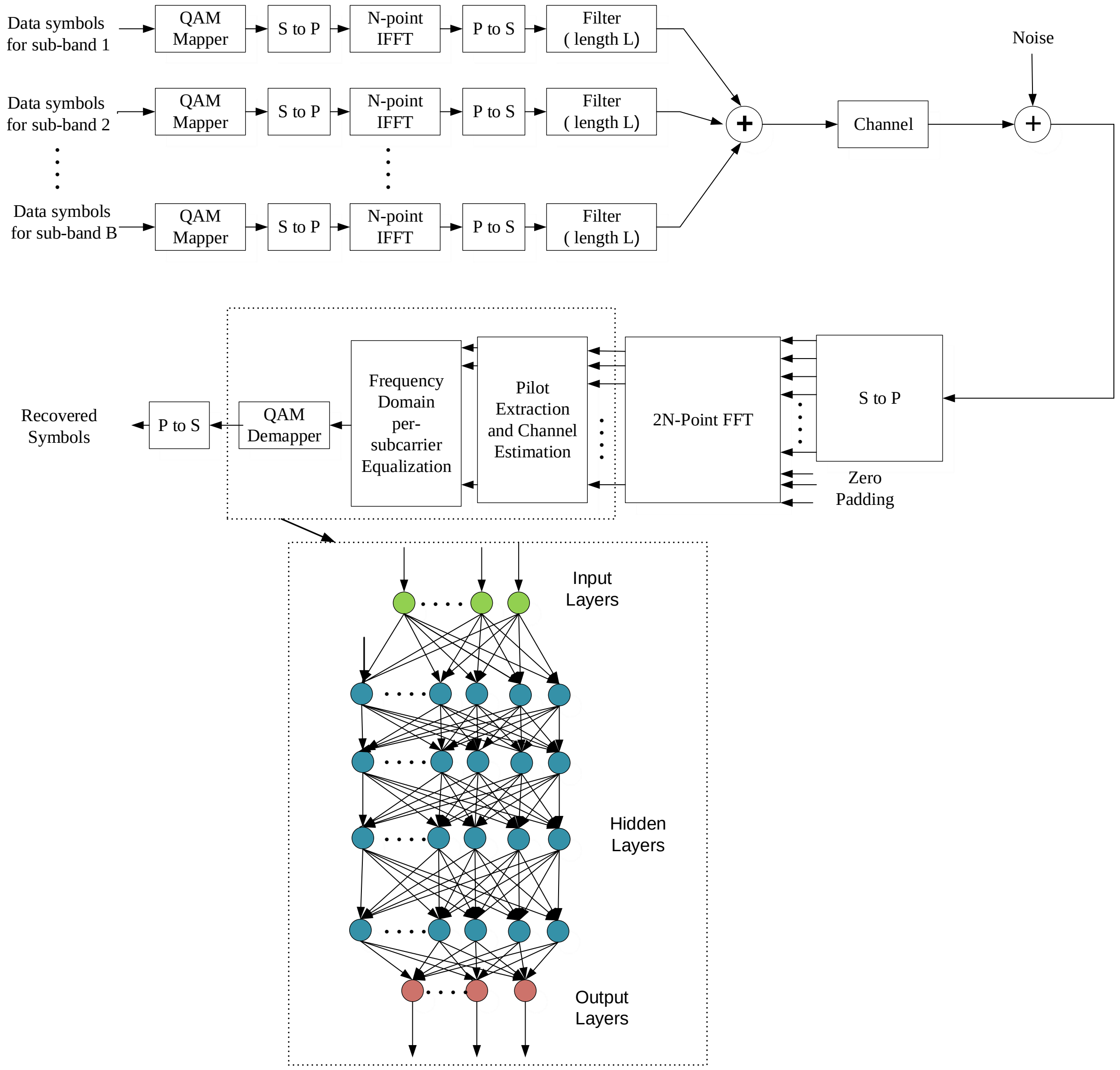


Fig. 6. 2 Neural Network based approach for joint channel estimation and signal detection in UFMC system

6.6 Simulation Results

In order demonstrate the performance of the proposed approach for UFMC, an extensive simulation experiments have been performed. The channel model considered for simulations in this study is Rayleigh fading channel. The transmitted symbols are with 4 QAM modulation scheme. The data obtained from the channel observations have been used to train the deep neural network. The BER performance of the proposed approach have been compared with conventional channel estimation approaches i.e. LS and MMSE.

The fig. 6.3 shows that BER performance varies with SNR used during training. As can be seen from the results the SNR of 8 dB during training provides the better BER performance.

The performance of proposed approach is superior when compared to LS and MMSE based approach which can be confirmed from the Fig, 6.4 and Fig. 6.5. Furthermore, when the number of taps of channel increases the BER degrades for all the approaches which can be confirmed from Fig. 6.5. For this channel also proposed approach is providing the significantly better performance.

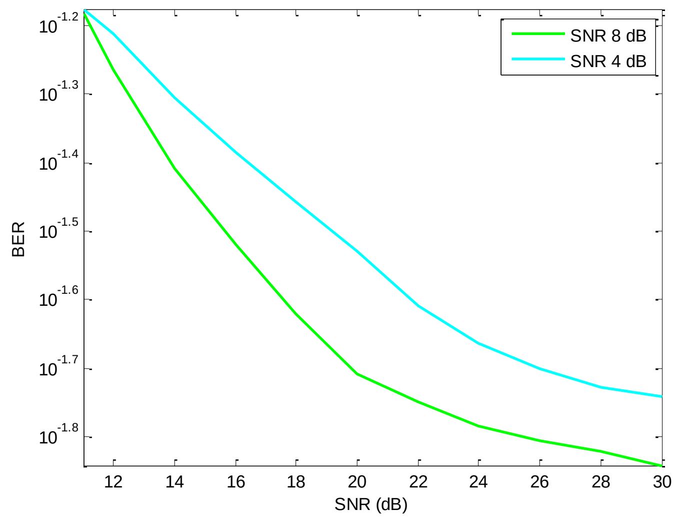


Fig 6.3 BER performance when DNN is trained at different SNR

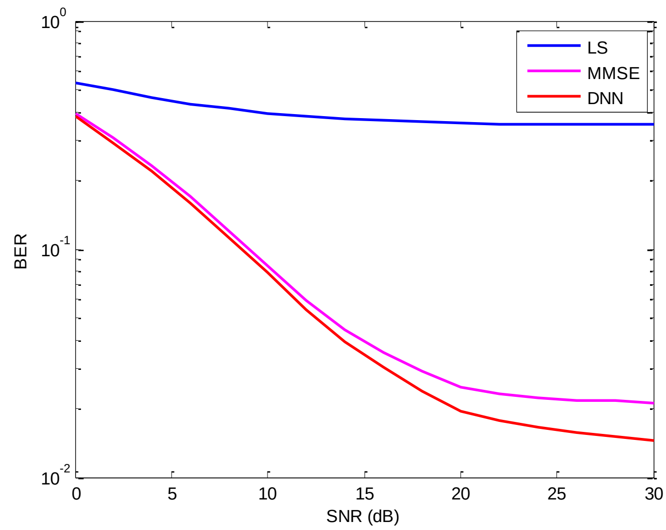


Fig. 6.4 BER performance for Rayleigh fading channel with 5 taps

6.7 Conclusion

In this chapter a neural network based approach have been proposed for joint channel estimation and data detection in UFMC system. The simulations confirmed that the proposed approach performs better than the existing methods**.** In future it is worth to investigate the performance of advanced versions of deep neural networks such as convolutional neural network for data detection in UFMC.

Chapter 7

Conclusions and Future Scope

7.1 Conclusions

The chapter wise conclusion of this thesis is summurised as follows:

In **Chapter 1**, introduction to the channel equalization in a wireless communication system, motivation, and contributions of the thesis have been explained.

In **Chapter 2**, basic system model of non-linear channel equalization is provided along with necessary mathematical equations. Furthermore, introduction about machine learning is given by explaining the basic concepts related to the neural networks. Finally, overview of metaheuristic algorithms and comprehensive review of the literature available in the area of channel equalization is also have been discussed.

In **Chapter 3**, a scheme has been proposed for training the Functional link artificial neural network based non-linear channel equalizer by using Cuckoo Search Algorithm (CSA).. Three non-linear channels were taken for simulations to validate the superiority of the CSA based training scheme and the results have been compared with recent and well-established algorithms. The simulations proved that CSA based training method offers improved performance in terms of MSE and BER when compared to existing algorithms**.** The robustness of the cuckoo search algorithm (CSA) based training scheme has been shown by considering the BER performance in a burst error scenario and it is observed that the scheme significantly outperforms the compared algorithms by effectively handling the burst errors. The performance of the proposed scheme has been validated for a wide range of signal-to-noise ratio (SNR 10 to 30 dB) values through simulation studies and it is observed that the scheme outperforms the other algorithms in poor SNR conditions as well. Moreover, the Wilcoxon rank-sum test proved that the proposed approach provided statistically significant results in comparison with competing approaches.

An efficient JAYA algorithm with Levy flight (JAYALF) is proposed in **Chapter 4.** To evaluate the performance of the JAYALF algorithm for non-linear channel equalization, three non-linear channels were taken for simulations. The simulation studies showed that JAYALF based equalizer provides superior performance than other equalizers in terms of convergence speed, steady-state MSE and BER. Finally, statistical test confirmed that the proposed algorithm provided statistically significant results in comparison with competing approaches.

In **Chapter 5**, a Modified Grasshopper Optimization Algorithm (MGOA) is proposed for equalization of wirless communication channels. The superiority of the proposed MGOA based equalizer is illustrated over other metaheuristic algorithms. The simulation results on four non-linear channels demonstrate the efficiency of the proposed MGOA algorithm in terms of BER and MSE. Furthermore, the statistical validity of the results obtained from MGOA is confirmed through Wilcoxon rank-sum test.

In **Chapter 6** a neural network based approach have been proposed for joint channel estimation and data detection in UFMC system. The simulations confirmed that the proposed approach performs better than the existing methods.

7.2 Future Scope

The research work carried out in this thesis can be extended in the future in different ways. The proposed metahheuristic algorithms and machine learning based approaches can used for data detection in MIMO applications. The data colletced from the real wireless communication channels can be used to train the channel equalizers to have a optimal perfromance in the real time environment. Moreover, the proposed metahheuristic algorithms and machine learning based approaches can be tested in the real-time environment using available hardware platform. The proposed approaches can be applied to other problems in wireless communication.

List of Publications

[1] Kishor Kisan Ingle, Ravi Kumar Jatoth, An Efficient JAYA Algorithm with Lévy Flight for Non-linear Channel Equalization, **Expert Systems with Applications**. 145 (2020) 112970. doi:10.1016/j.eswa.2019.112970.

(**SCIE Indexed**, **Elsevier, Impact Factor: 6.954**)

[2] Kishor Kisan Ingle, Ravi Kumar Jatoth, A New Training Scheme for Neural Network based Non-linear Channel Equalizers in Wireless Communication System using Cuckoo Search Algorithm. **AEU – International Journal of Electronics and Communications** 138(2021) 153371. https://doi.org/10.1016/j.aeue.2020.153371.

(**SCIE Indexed**, **Elsevier, Impact Factor: 3.183**)

[3] Kishor Kisan Ingle, Ravi Kumar Jatoth "Non-linear Channel Equalization using the modified grasshopper optimization" **Applied Soft Computing**

(**Received revision for this paper and currently revising, SCIE Indexed**, **Elsevier, Impact Factor: 6.725**)

Bibliography

[1] J.C. Patra, P.K. Meher, G. Chakraborty, Nonlinear channel equalization for wireless communication systems using Legendre neural networks, Signal Processing. 89 (2009) 2251–2262. doi:10.1016/j.sigpro.2009.05.004.

[2] S. Haykin, Adaptive Filter Theory, Third Edit, Prentice-Hall, 1996.

[3] B. Widrow, S.D. Stearns, Adaptive Signal Processing, Prentice Hall, Englewood Cliffs, NJ, 1985.

[4] S.U.H. Qureshi, Adaptive Equalization, Proc. IEEE. 73 (1985) 1349–1387. doi:10.1109/PROC.1985.13298.

[5] S. Chen, Y. Wu, S. McLaughlin, Genetic algorithm optimization for blind channel identification with higher order cumulant fitting, IEEE Trans. Evol. Comput. 1 (1997) 259–265. doi:10.1109/4235.687886.

[6] J.G. Proakis, M. Salehi, Digital Communications, Fifth Edit, McGraw-Hill, New York, 2008.

[7] J.C. Patra, W.B. Poh, N.S. Chaudhari, A. Das, Nonlinear channel equalization with QAM signal using Chebyshev artificial neural network, Proc. Int. Jt. Conf. Neural Networks. 5 (2005) 3214–3219. doi:10.1109/IJCNN.2005.1556442.

[8] S. Chen, G.J. Gibson, C.F.N. Cowan, P.M. Grant, Adaptive equalization of finite non-linear channels using multilayer perceptrons, Signal Processing. 20 (1990) 107–119. doi:10.1016/0165-1684(90)90122-F.

[9] G.J. Gibson, S. Siu, C.F.N. Cowan, Multi-layer perceptron structures applied to adaptive equalizers for data communications, ICASSP, IEEE Int. Conf. Acoust. Speech Signal Process. 2 (1989) 1183–1186. doi:10.1109/ICASSP.1989.266645.

[10] G.J. Gibson, S. Siu, C.F.N. Cowan, The Application of Nonlinear Structures to the Reconstruction of Binary Signals, IEEE Trans. Signal Process. 39 (1991) 1877–1884. doi:10.1109/78.91157.

[11] J.C. Patra, R.N. Pal, R. Baliarsingh, G. Panda, Nonlinear channel equalization for QAM signal constellation using artificial neural networks, IEEE Trans. Syst. Man, Cybern. Part B Cybern. 29 (1999) 262–271. doi:10.1109/3477.752798.

[12] D.E. Rumelhart, G.E. Hinton, R.J. Williams, Learning Internal Representations By Error Propagation, in: D.E. Rumelhart, J.L. McClelland (Eds.), Parallel Distrib. Process. Explor. Microstruct. Cogn., MIT Press, Cambridge, MA, 1986: pp. 318–362.

[13] M. Gori, A. Tesi, On the Problem of Local Minima in Backpropagation, IEEE Trans. Pattern Anal. Mach. Intell. 14 (1992) 76–86. doi:10.1109/34.107014.

[14] Y. Lee, S.-H. Oh, M.W. Kim, An Analysis of Premature Saturation in Back Popagation Learning, Neural Networks. 6 (1993) 719–728. doi:10.1016/S0893-6080(05)80116-9.

[15] V.K. Ojha, A. Abraham, V. Snášel, Metaheuristic design of feedforward neural networks: A review of two decades of research, Eng. Appl. Artif. Intell. 60 (2017) 97–116.

doi:10.1016/j.engappai.2017.01.013.

[16] S. Mirjalili, S.M. Mirjalili, A. Lewis, Let a biogeography-based optimizer train your Multi-Layer Perceptron, Inf. Sci. (Ny). 269 (2014) 188–209. doi:10.1016/j.ins.2014.01.038.

[17] S. Panda, P.K. Mohapatra, S. Prasada Panigrahi, A new training scheme for neural networks and application in non-linear channel equalization, Appl. Soft Comput. J. 27 (2015) 47–52. doi:10.1016/j.asoc.2014.10.040.

[18] G. Das, P.K. Pattnaik, S.K. Padhy, Artificial Neural Network trained by Particle Swarm Optimization for non-linear channel equalization, Expert Syst. Appl. 41 (2014) 3491–3496. doi:10.1016/j.eswa.2013.10.053.

[19] S.J. Nanda, N. Jonwal, Robust nonlinear channel equalization using WNN trained by symbiotic organism search algorithm, Appl. Soft Comput. J. 57 (2017) 197–209. doi:10.1016/j.asoc.2017.03.029.

[20] V.G. Gudise, G.K. Venayagamoorthy, Comparison of Particle Swarm Optimization and Backpropagation as Training Algorithms for Neural Networks, Proc. 2003 IEEE Swarm Intell. Symp. SIS'03 (Cat. No.03EX706). (2003) 110–117. doi:10.1109/SIS.2003.1202255.

[21] W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity, Bull. Math. Biophys. 5 (1943) 115–133. doi:10.1007/BF02478259.

[22] A. Gorin, R.J. Mammone, Introduction To The Special Issue On Neural Networks For Speech Processing, IEEE Trans. Speech Audio Process. 2 (1994) 113–114. doi:10.1109/89.260355.

[23] J.-N. Hwang, S.-Y. Kung, M. Niranjan, J.C. Principe, The past, present, and future of neural networks for signal processing, IEEE Signal Process. Mag. 14 (1997) 28–48. doi:10.1109/79.637299.

[24] A. Krizhevsky, I. Sutskever, G.E. Hinton, ImageNet classification with deep convolutional neural networks, Commun. ACM. 60 (2017) 84–90. doi:10.1145/3065386.

[25] M. Nielsen, Neural Networks and Deep Learning, Determ. Press. (2015). http://neuralnetworksanddeeplearning.com.

[26] M. Allander, Channel Equalization using Machine Learning for Underwater Acoustic Communications, Master Sci. Thesis Electr. Eng. Linköping Univ. (2020).

[27] A.K. Jain, J. Mao, K.M. Mohiuddin, Artificial neural networks: a tutorial, Computer (Long. Beach. Calif). 29 (1996) 31–44. doi:10.1109/2.485891.

[28] A.K. Sahoo, Adaptive Nonlinear System Identification and Channel Equalization Using Functional Link Artificial Neural Network, A Master Technol. Thesis Submitt. Dep. Electron. Commun. Eng. Natl. Inst. Technol. Rourkela. (2007).

[29] F. Rosenblatt, The perceptron: A probabilistic model for information storage and organization in the brain, Psychol. Rev. 65 (1958) 386–408. doi:10.1037/h0042519.

[30] Y.-H. Pao, Adaptive pattern recognition and neural networks, Addison-Wesley, Reading, MA, 1989.

[31] Y.-H. Pao, Y. Takefuji, Functional-Link Net Computing: Theory, System Architecture,

and Functionalities, Computer (Long. Beach. Calif). 25 (1992) 76–79. doi:10.1109/2.144401.

[32] H. Zhao, J. Zhang, Functional link neural network cascaded with Chebyshev orthogonal polynomial for nonlinear channel equalization, Signal Processing. 88 (2008) 1946–1957. doi:10.1016/j.sigpro.2008.01.029.

[33] J.C. Patra, R.N. Pal, A functional link artificial neural network for adaptive channel equalization, Signal Processing. 43 (1995) 181–195. doi:10.1016/0165-1684(94)00152-P.

[34] J.C. Patra, W.C. Chin, P.K. Meher, G. Chakraborty, Legendre-FLANN-based nonlinear channel equalization in wireless communication system, Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern. (2008) 1826–1831. doi:10.1109/ICSMC.2008.4811554.

[35] H. Zhao, X. Zeng, X. Zhang, J. Zhang, Y. Liu, T. Wei, An adaptive decision feedback equalizer based on the combination of the FIR and FLNN, Digit. Signal Process. A Rev. J. 21 (2011) 679–689. doi:10.1016/j.dsp.2011.05.004.

[36] R.K. Jatoth, M.S.B.S. VADDADI, S.S.V.K.K. Anoop, An Intelligent Functional Link Artificial Neural Network for Channel Equalization, in: Proc. 8th WSEAS Int. Conf. Signal Process. Robot. Autom., 2009: pp. 240–245. http://dl.acm.org/citation.cfm?id=1558916.1558958.

[37] H. Zhao, J. Zhang, Adaptively combined FIR and functional link artificial neural network equalizer for nonlinear communication channel, IEEE Trans. Neural Networks. 20 (2009) 665–674. doi:10.1109/TNN.2008.2011481.

[38] T.-T. Lee, J.-T. Jeng, The Chebyshev-Polynomials-Based Unified Model Neural Networks for Function Approximation, IEEE Trans. Syst. MAN, Cybern. B Cybern. 28 (1998) 925–935. doi:10.1109/3477.735405.

[39] J.C. Patra, R.N. Pal, B.N. Chatterji, G. Panda, Identification of Nonlinear Dynamic Systems Using Functional Link Artificial Neural Networks., IEEE Trans. Syst. Man. Cybern. B. Cybern. 29 (1999) 254–262. doi:10.1109/3477.752797.

[40] A. Gotmare, R. Patidar, N. V. George, Nonlinear system identification using a cuckoo search optimized adaptive Hammerstein model, Expert Syst. Appl. 42 (2015) 2538–2546. doi:10.1016/j.eswa.2014.10.040.

[41] J.C. Patra, A.C. Kot, Nonlinear Dynamic System Identification Using Chebyshev Functional Link Artificial Neural Networks, IEEE Trans. Syst. Man, Cybern. Part B, Cybern. 32 (2002) 505–511. doi:10.1109/TSMCB.2002.1018769.

[42] S. Mall, S. Chakraverty, Application of Legendre Neural Network for solving ordinary differential equations, Appl. Soft Comput. J. 43 (2016) 347–356. doi:10.1016/j.asoc.2015.10.069.

[43] N. V. George, A. Gonzalez, Convex combination of nonlinear adaptive filters for active noise control, Appl. Acoust. 76 (2014) 157–161. doi:10.1016/j.apacoust.2013.08.005.

[44] I.H. Osman, G. Laporte, Metaheuristics: A bibliography, Ann. Oper. Res. 63 (1996) 513–623. doi:10.1007/bf02125421.

[45] X.-S. Yang, Nature-Inspired Metaheuristic Algorithms, LUNIVER Press. Second Ed. (2010).

[46] R. Zhang, X. Jiang, R. Li, Improved decomposition-based multi-objective cuckoo search algorithm for spectrum allocation in cognitive vehicular network, Phys. Commun. 34 (2019) 301–309. doi:10.1016/j.phycom.2018.06.003.

[47] T. Çavdar, PSO tuned ANFIS equalizer based on fuzzy C-means clustering algorithm, AEU - Int. J. Electron. Commun. 70 (2016) 799–807. doi:10.1016/j.aeue.2016.03.006.

[48] A. Aggarwal, T.K. Rawat, D.K. Upadhyay, Design of optimal digital FIR filters using evolutionary and swarm optimization techniques, AEU - Int. J. Electron. Commun. 70 (2016) 373–385. doi:10.1016/j.aeue.2015.12.012.

[49] P.J. van Laarhoven, E.. Aarts, Simulated Annealing: Theory and Applications, Springer. (1987). doi:10.1007/978-94-015-7744-1.

[50] H. Szu, R. Hartley, Fast simulated annealing, Phys. Lett. A. 122 (1987) 157–162. doi:10.1016/0375-9601(87)90796-1.

[51] J.H. Holland, Adaptation in Natural and Artificial Systems: an Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence., Univ. Michigan Press. (1975).

[52] D.E. Goldberg, J.H. Holland, Genetic Algorithms and Machine Learning, Mach. Learn. 3 (1988) 95–99. doi:10.1023/A:1022602019183.

[53] R. Eberhart, J. Kennedy, New optimizer using particle swarm theory, Proc. Sixth Int. Symp. Micro Mach. Hum. Sci. (1995) 39–43. doi:10.1109/mhs.1995.494215.

[54] K.M. Passino, Biomimicry of Bacterial Foraging for Distributed Optimization and Control, IEEE Control Syst. Mag. 22 (2002) 52–67. doi:10.1109/MCS.2002.1004010.

[55] J.R. Zhang, J. Zhang, T.M. Lok, M.R. Lyu, A hybrid particle swarm optimization-back-propagation algorithm for feedforward neural network training, Appl. Math. Comput. 185 (2007) 1026–1037. doi:10.1016/j.amc.2006.07.025.

[56] C.F. Juang, A Hybrid of Genetic Algorithm and Particle Swarm Optimization for Recurrent Network Design, IEEE Trans. Syst. Man, Cybern. Part B Cybern. 34 (2004) 997–1006. doi:10.1109/TSMCB.2003.818557.

[57] X.-S. Yang, S. Deb, Cuckoo Search via Lévy Flights, in: 2009 World Congr. Nat. Biol. Inspired Comput. NABIC 2009 - Proc., 2009: pp. 210–214. doi:10.1109/NABIC.2009.5393690.

[58] X.-S. Yang, S. Deb, Cuckoo search: recent advances and applications, Neural Comput. Appl. 24 (2014) 169–174. doi:10.1007/s00521-013-1367-1.

[59] X.-S. Yang, Cuckoo Search and Firefly Algorithm: Overview and Analysis, in: X.-S. Yang (Ed.), Cuckoo Search Firefly Algorithm,Studies Comput. Intell., Springer, 2014: pp. 1–26. doi:10.1007/978-3-319-02141-6_1.

[60] I. Pavlyukevich, Lévy flights, non-local search and simulated annealing, J. Comput. Phys. 226 (2007) 1830–1844. doi:10.1016/j.jcp.2007.06.008.

[61] P. Civicioglu, E. Besdok, A conceptual comparison of the Cuckoo-search, particle swarm optimization, differential evolution and artificial bee colony algorithms, Artif. Intell. Rev. 39 (2013) 315–346. doi:10.1007/s10462-011-9276-0.

[62] S. Agrawal, R. Panda, S. Bhuyan, B.K. Panigrahi, Tsallis entropy based optimal

multilevel thresholding using cuckoo search algorithm, Swarm Evol. Comput. 11 (2013) 16–30. doi:10.1016/j.swevo.2013.02.001.

[63] S. Agrawal, R. Panda, An Efficient Algorithm for Gray Level Image Enhancement Using Cuckoo Search, in: B.K. Panigrahi, S. Das, P.N. Suganthan, P.K. Nanda (Eds.), Swarm, Evol. Memetic Comput. SEMCCO 2012. Lect. Notes Comput. Sci., Springer, 2012: pp. 82–89. doi:10.1007/978-3-642-35380-2_11.

[64] M. Kumar, T.K. Rawat, Optimal design of FIR fractional order differentiator using cuckoo search algorithm, Expert Syst. Appl. 42 (2015) 3433–3449. doi:10.1016/j.eswa.2014.12.020.

[65] A. Sikander, P. Thakur, R.C. Bansal, S. Rajasekar, A novel technique to design cuckoo search based FOPID controller for AVR in power systems, Comput. Electr. Eng. 70 (2018) 261–274. doi:10.1016/j.compeleceng.2017.07.005.

[66] D.N. Vo, P. Schegner, W. Ongsakul, Cuckoo search algorithm for non-convex economic dispatch, IET Gener. Transm. Distrib. 7 (2013) 645–654. doi:10.1049/iet-gtd.2012.0142.

[67] S. Araghi, A. Khosravi, D. Creighton, Intelligent cuckoo search optimized traffic signal controllers for multi-intersection network, Expert Syst. Appl. 42 (2015) 4422–4431. doi:10.1016/j.eswa.2015.01.063.

[68] A.P. Patwardhan, R. Patidar, N. V. George, On a cuckoo search optimization approach towards feedback system identification, Digit. Signal Process. 32 (2014) 156–163. doi:10.1016/j.dsp.2014.05.008.

[69] R.V. Rao, Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems, Int. J. Ind. Eng. Comput. 7 (2016) 19–34. doi:10.5267/j.ijiec.2015.8.004.

[70] R. Venkata Rao, Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems, Int. J. Ind. Eng. Comput. 7 (2016) 19–34. doi:10.5267/j.ijiec.2015.8.004.

[71] S.P. Singh, T. Prakash, V.P. Singh, M.G. Babu, Analytic hierarchy process based automatic generation control of multi-area interconnected power system using Jaya algorithm, Eng. Appl. Artif. Intell. 60 (2017) 35–44. doi:10.1016/j.engappai.2017.01.008.

[72] S.O. Degertekin, L. Lamberti, I.B. Ugur, Discrete sizing/layout/topology optimization of truss structures with an advanced Jaya algorithm, Appl. Soft Comput. J. 79 (2019) 363–390. doi:10.1016/j.asoc.2019.03.058.

[73] W. Warid, H. Hizam, N. Mariun, N.I. Abdul-Wahab, Optimal power flow using the Jaya algorithm, Energies. 9 (2016). doi:10.3390/en9090678.

[74] M. Aslan, M. Gunduz, M.S. Kiran, JayaX: Jaya algorithm with xor operator for binary optimization, Appl. Soft Comput. J. 82 (2019). doi:10.1016/j.asoc.2019.105576.

[75] S. Basak, S. Banerjee, Adaptive Jaya Optimization Technique for Economic Load Dispatch Considering Valve Point Effect, in: 2019 Devices Integr. Circuit, IEEE, Kalyani, India, 2019: pp. 104–107. doi:10.1109/devic.2019.8783691.

[76] Z. Ding, J. Li, H. Hao, Structural damage identification using improved Jaya algorithm based on sparse regularization and Bayesian inference, Mech. Syst. Signal Process. 132

(2019) 211–231. doi:10.1016/j.ymssp.2019.06.029.

[77] C. Pradhan, C.N. Bhende, Online load frequency control in wind integrated power systems using modified Jaya optimization, Eng. Appl. Artif. Intell. 77 (2019) 212–228. doi:10.1016/j.engappai.2018.10.003.

[78] U. Raut, S. Mishra, An improved Elitist–Jaya algorithm for simultaneous network reconfiguration and DG allocation in power distribution systems, Renew. Energy Focus. 30 (2019) 92–106. doi:10.1016/j.ref.2019.04.001.

[79] P. Singh, H. Chaudhary, A Modified Jaya Algorithm for Mixed-Variable Optimization Problems, J. Intell. Syst. (2018). doi:10.1515/jisys-2018-0273.

[80] Z. Yang, Y. Guo, Q. Niu, H. Ma, Y. Zhou, L. Zhang, A Novel Binary Jaya Optimization for Economic/Emission Unit Commitment, in: 2018 IEEE Congr. Evol. Comput., IEEE, 2018. doi:10.1109/CEC.2018.8477660.

[81] S. Mishra, P.K. Ray, Power Quality Improvement Using Photovoltaic Fed DSTATCOM Based on JAYA Optimization, IEEE Trans. Sustain. Energy. 7 (2016) 1672–1680. doi:10.1109/TSTE.2016.2570256.

[82] R. V. Rao, K.C. More, J. Taler, P. Ocłoń, Dimensional optimization of a micro-channel heat sink using Jaya algorithm, Appl. Therm. Eng. 103 (2016) 572–582. doi:10.1016/j.applthermaleng.2016.04.135.

[83] N. Kumar, I. Hussain, B. Singh, B.K. Panigrahi, Rapid MPPT for Uniformly and Partial Shaded PV System by Using JayaDE Algorithm in Highly Fluctuating Atmospheric Conditions, IEEE Trans. Ind. Informatics. 13 (2017) 2406–2416. doi:10.1109/TII.2017.2700327.

[84] S. Ghavidel, A. Azizivahed, L. Li, A hybrid Jaya algorithm for reliability–redundancy allocation problems, Eng. Optim. 50 (2018) 698–715. doi:10.1080/0305215X.2017.1337755.

[85] R.V. Rao, A. Saroj, Economic optimization of shell-and-tube heat exchanger using Jaya algorithm with maintenance consideration, Appl. Therm. Eng. 116 (2017) 473–487. doi:10.1016/j.applthermaleng.2017.01.071.

[86] R.V. Rao, G.G. Waghmare, A new optimization algorithm for solving complex constrained design optimization problems, Eng. Optim. 49 (2017) 60–83. doi:10.1080/0305215X.2016.1164855.

[87] K. Abhishek, V.R. Kumar, S. Datta, S.S. Mahapatra, Application of JAYA algorithm for the optimization of machining performance characteristics during the turning of CFRP (epoxy) composites: comparison with TLBO, GA, and ICA, Eng. Comput. 33 (2017) 457–475. doi:10.1007/s00366-016-0484-8.

[88] K. Yu, J.J. Liang, B.Y. Qu, X. Chen, H. Wang, Parameters identification of photovoltaic models using an improved JAYA optimization algorithm, Energy Convers. Manag. 150 (2017) 742–753. doi:10.1016/j.enconman.2017.08.063.

[89] R.V. Rao, A. Saroj, Multi-objective design optimization of heat exchangers using elitist-Jaya algorithm, Energy Syst. 9 (2018) 305–341. doi:10.1007/s12667-016-0221-9.

[90] S. Saremi, S. Mirjalili, A. Lewis, Grasshopper Optimisation Algorithm : Theory and application, Adv. Eng. Softw. 105 (2017) 30–47. doi:10.1016/j.advengsoft.2017.01.004.

[91] A. Asghar, H. Hossam, F. Ibrahim, S. Mirjalili, An efficient hybrid multilayer perceptron neural network with grasshopper optimization, Soft Comput. (2018). doi:10.1007/s00500-018-3424-2.

[92] A.A. El-fergany, Electrical characterisation of proton exchange membrane fuel cells stack using grasshopper optimiser, IET Renew. Power Gener. Res. 12 (2018) 9–17. doi:10.1049/iet-rpg.2017.0232.

[93] I. Aljarah, H. Saadeh, A.M.A. Hossam, F. Mohammad, A.H. Seyedali, Simultaneous Feature Selection and Support Vector Machine Optimization Using the Grasshopper Optimization Algorithm, Cognit. Comput. 10 (2018) 478–495. doi:10.1007/s12559-017-9542-9.

[94] M. Mafarja, I. Aljarah, A. Asghar, A.I. Hammouri, H. Faris, A.M. Al-zoubi, S. Mirjalili, Knowle dge-Base d Systems Evolutionary Population Dynamics and Grasshopper Optimization approaches for feature selection problems, Knowledge-Based Syst. 145 (2018) 25–45. doi:10.1016/j.knosys.2017.12.037.

[95] A. Zakeri, A. Hokmabadi, Efficient feature selection method using real-valued grasshopper optimization algorithm, Expert Syst. Appl. 119 (2019) 61–72. doi:10.1016/j.eswa.2018.10.021.

[96] N. Rajput, V. Chaudhary, H.M. Dubey, M. Pandit, Optimal generation scheduling of thermal System using biologically inspired Grasshopper Algorithm, in: 2017 2nd Int. Conf. Telecommun. Networks (TEL-NET, 2017: pp. 3–8.

[97] S. Łukasik, P.A. Kowalski, M. Charytanowicz, P. Kulczycki, Data Clustering with Grasshopper Optimization Algorithm, in: 2017 Fed. Conf. Comput. Sci. Inf. Syst., 2017: pp. 71–74. doi:10.15439/2017F340.

[98] D. Potnuru, A.S.L. V Tummala, Implementation of Grasshopper Optimization Algorithm for Controlling a BLDC Motor Drive, Springer Singapore, n.d. doi:10.1007/978-981-13-0514-6.

[99] J. Wu, H. Wang, N. Li, P. Yao, Y. Huang, Z. Su, Distributed trajectory optimization for multiple solar-powered UAVs target tracking in urban environment by Adaptive Grasshopper Optimization Algorithm, Aerosp. Sci. Technol. 70 (2017) 497–510. doi:10.1016/j.ast.2017.08.037.

[100] A.A. Ewees, M. Abd, E.H. Houssein, Improved grasshopper optimization algorithm using opposition-based learning, Expert Syst. Appl. 112 (2018) 156–172. doi:10.1016/j.eswa.2018.06.023.

[101] S. Arora, P. Anand, Chaotic grasshopper optimization algorithm for global optimization, Neural Comput. Appl. 2 (2018). doi:10.1007/s00521-018-3343-2.

[102] A. Saxena, S. Shekhawat, R. Kumar, Application and Development of Enhanced Chaotic Grasshopper Optimization Algorithms, Model. Simul. Eng. 2018 (2018). doi:10.1155/2018/4945157.

[103] S. Zahra, M. Seyedali, M. Shahrzad, S. Hossam, I. Aljarah, Grasshopper optimization algorithm for multi-objective optimization problems, Appl Intell. (2018). doi:10.1007/s10489-017-1019-8.

[104] A. Tharwat, E.H. Houssein, M.M. Ahmed, A.E. Hassanien, T. Gabel, MOGOA algorithm for constrained and unconstrained multi-objective optimization problems, Appl Intell. 48

(2018) 2268–2283. doi:10.1007/s10489-017-1074-1 MOGOA.

[105] R.W. Lucky, Automatic Equalization for Digital Communication, Bell Syst. Tech. J. 44 (1965) 547–588. doi:10.1002/j.1538-7305.1965.tb01678.x.

[106] J.M. Cioffi, T. Kailath, An Efficient, Recursive-Least-Squares, Fractionally Spaced Equalizer using Intersymbol Interpolation, in: IEEE Int. Conf. Commun. 1984, ICC ’84, IEEE, 1984: pp. 369–372.

[107] D.T.M. Slock, On the Convergence Behavior of the LMS and the Normalized LMS Algorithms, IEEE Trans. Signal Process. 41 (1993) 2811–2825. doi:10.1109/78.236504.

[108] A.T. Al-Awami, A. Zerguine, L. Cheded, A. Zidouri, W. Saif, A new modified particle swarm optimization algorithm for adaptive equalization, Digit. Signal Process. 21 (2011) 195–207. doi:10.1016/j.dsp.2010.05.001.

[109] S. Chen, G.J. Gibson, C.F.N. Cowan, P.M. Grant, Reconstruction of binary signals using an adaptive radial basis function equaliser, EURASIP Signal Process. 22 (1991) 77–93.

[110] S. Chen, G.J. Gibson, C.F.N. Cowan, Adaptive channel equalisation using a polynomial-perceptron structure, IEE Proceedings, Part I Commun. Speech Vis. 137 (1990) 257–264. doi:10.1049/ip-i-2.1990.0036.

[111] K. Burse, R.N. Yadav, S.C. Shrivastava, Channel equalization using neural networks: A review, IEEE Trans. Syst. Man Cybern. Part C Appl. Rev. 40 (2010) 352–357. doi:10.1109/TSMCC.2009.2038279.

[112] S. Chen, B. Mulgrew, P.M. Grant, A Clustering Technique for Digital Communications Channel Equalization Using Radial Basis Function Networks, IEEE Trans. Neural Networks. 4 (1993) 570–590. doi:10.1109/72.238312.

[113] J.C. Patra, R.N. Pal, Functional link artificial neural network-based adaptive channel equalization of nonlinear channels with QAM signal, Proc. IEEE Int. Conf. Syst. Man Cybern. 3 (1995) 2081–2086. doi:10.1109/icsmc.1995.538086.

[114] S.E. Fahlman, An Empirical Study of Learning Speed in Back-propagation Networks, Tech. Rep. C. Carnegie-Mellon Univ. (1988). http://repository.cmu.edu/cgi/viewcontent.cgi?article=2799&context=compsci.

[115] S. Panda, A. Sarangi, S.P. Panigrahi, A new training strategy for neural network using shuffled frog-leaping algorithm and application to channel equalization, AEU - Int. J. Electron. Commun. 68 (2014) 1031–1036. doi:10.1016/j.aeue.2014.05.005.

[116] M. Krishnan, S. Yun, Y.M. Jung, Improved clustering with firefly-optimization-based mobile data collector for wireless sensor networks, AEU - Int. J. Electron. Commun. 97 (2018) 242–251. doi:10.1016/j.aeue.2018.10.014.

[117] Y.C. Kuyu, F. Vatansever, A new intelligent decision making system combining classical methods, evolutionary algorithms and statistical techniques for optimal digital FIR filter design and their performance evaluation, AEU - Int. J. Electron. Commun. 70 (2016) 1651–1666. doi:10.1016/j.aeue.2016.10.004.

[118] S. Chen, Y. Wu, S. McLaughlin, Genetic Algorithm Optimization for Blind Channel Identification with Higher Order Cumulant Fitting, IEEE Trans. Evol. Comput. 1 (1997) 259–265. doi:10.1109/4235.687886.

[119] S. Chen, Y. Wu, Maximum Likelihood Joint Channel and Data Estimation Using Genetic Algorithms, IEEE Trans. Signal Process. 46 (1998) 1469–1473. doi:10.1109/78.668813.

[120] S. Han, W. Pedrycz, C. Han, Nonlinear Channel Blind Equalization Using Hybrid Genetic Algorithm with Simulated Annealing, Math. Comput. Model. 41 (2005) 697–709. doi:10.1016/j.mcm.2004.05.006.

[121] A.T. Al-Awami, W. Saif, A. Zerguine, A. Zidouri, L. Cheded, An Adaptive Equalizer Based On Particle Swarm Optimization Techniques, 2007 9th Int. Symp. Signal Process. Its Appl. ISSPA 2007, Proc. (2007). doi:10.1109/ISSPA.2007.4555436.

[122] A.A. Al-shaikhi, A.H. Khan, A.T. Al-awami, A. Zerguine, A Hybrid Particle Swarm Optimization Technique for Adaptive Equalization, Arab. J. Sci. Eng. 44 (2019) 2177–2184. doi:10.1007/s13369-018-3387-8.

[123] S.J. Nanda, G. Panda, B. Majhi, Development of Novel Digital Equalizers for Noisy Nonlinear Channel using Artificial Immune System, 2008 IEEE Reg. 10 Colloq. 3rd Int. Conf. Ind. Inf. Syst. (2008). doi:10.1109/ICIINFS.2008.4798337.

[124] S. Pandey, R. Patidar, N. V. George, Design of a Krill Herd Algorithm based Adaptive Channel Equalizer, 2014 IEEE Int. Symp. Intell. Signal Process. Commun. Syst. ISPACS 2014. (2014) 257–260. doi:10.1109/ISPACS.2014.7024463.

[125] T.S.D. Singh, A. Chatterjee, MMSE design of nonlinear Volterra equalizers using artificial bee colony algorithm, Meas. J. Int. Meas. Confed. 46 (2013) 210–219. doi:10.1016/j.measurement.2012.06.008.

[126] B. Majhi, G. Panda, A. Choubey, On the Development of a New Adaptive Channel Equalizer using Bacterial Foraging Optimization Technique, 2006 Annu. India Conf. INDICON. (2006). doi:10.1109/INDCON.2006.302761.

[127] T.-J. Su, J.-C. Cheng, C.-J. Yu, An adaptive channel equalizer using self-adaptation bacterial foraging optimization, Opt. Commun. 283 (2010) 3911–3916. doi:10.1016/j.optcom.2010.06.007.

[128] G.R. Patra, S. Maity, S. Sardar, S. Das, Nonlinear Channel Equalization for Digital Communications Using DE-Trained Functional Link Artificial Neural Networks, in: S. Aluru, S. Bandyopadhyay, U.V. Catalyurek, D.P. Dubhashi, P.H. Jones, M. Parashar, B. Schmidt (Eds.), Contemp. Comput. IC3 2011. Commun. Comput. Inf. Sci., 2011: pp. 403–414. doi:10.1007/978-3-642-22606-9_41.

[129] Y. Shi, R. Eberhart, A Modified Particle Swarm Optimizer, in: 1998 IEEE Int. Conf. Evol. Comput. Proceedings. IEEE World Congr. Comput. Intell., 1998: pp. 2675–2679. doi:10.1109/ICEC.1998.699146.

[130] J. Kennedy, R. Eberhart, Particle Swarm Optimization, in: Proc. ICNN'95 - Int. Conf. Neural Networks, 1995: pp. 1942–1948. doi:10.1109/ICNN.1995.488968.

[131] R. Storn, K. Price, Differential Evolution – A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, J. Glob. Optim. 11 (1997) 341–359. doi:10.1023/A:1008202821328.

[132] S. Mirjalili, SCA: A Sine Cosine Algorithm for solving optimization problems, Knowledge-Based Syst. 96 (2016) 120–133. doi:10.1016/j.knosys.2015.12.022.

[133] S. Mirjalili, S.M. Mirjalili, A. Lewis, Grey Wolf Optimizer, Adv. Eng. Softw. 69 (2014)

46–61. doi:10.1016/j.advengsoft.2013.12.007.

[134] S. Mirjalili, Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems, Neural Comput. Appl. 27 (2016) 1053–1073. doi:10.1007/s00521-015-1920-1.

[135] S. Mirjalili, A. Lewis, The Whale Optimization Algorithm, Adv. Eng. Softw. 95 (2016) 51–67. doi:10.1016/j.advengsoft.2016.01.008.

[136] X.-S. Yang, S. Deb, Engineering optimisation by cuckoo search, Int. J. Math. Model. Numer. Optim. 1 (2010) 330–343. doi:10.1504/IJMMNO.2010.035430.

[137] H. Zhao, X. Zeng, J. Zhang, T. Li, Y. Liu, D. Ruan, Pipelined functional link artificial recurrent neural network with the decision feedback structure for nonlinear channel equalization, Inf. Sci. (Ny). 181 (2011) 3677–3692. doi:10.1016/j.ins.2011.04.033.

[138] A.A. Al-Shaikhi, A.H. Khan, A.T. Al-Awami, A. Zerguine, A Hybrid Particle Swarm Optimization Technique for Adaptive Equalization, Arab. J. Sci. Eng. 44 (2019) 2177–2184. doi:10.1007/s13369-018-3387-8.

[139] D.F. Williamson, R.A. Parker, J.S. Kendrick, The Box Plot : A Simple Visual Method to Interpret Data, Ann. Intern. Med. 110 (1989) 916–921. doi:https://doi.org/10.7326/0003-4819-110-11-916.

[140] S. García, D. Molina, M. Lozano, F. Herrera, A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 Special Session on Real Parameter Optimization, J. Heuristics. 15 (2009) 617–644. doi:10.1007/s10732-008-9080-4.

[141] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (2011) 3–18. doi:10.1016/j.swevo.2011.02.002.

[142] F. Wilcoxon, Individual Comparisons by Ranking Methods, Biometrics Bull. 1 (1945) 80–83. doi:https://doi.org/10.2307/3001968.

[143] T.M.A. Taher, A. Al-Banna, ADAPTIVE EQUALIZATION IN THE PRESENCE OF BURST ERRORS, US Pat. No. US 8,385,400 B2. (2013).

[144] A. V. Chechkin, R. Metzler, J. Klafter, V.Y. Gonchar, Introduction to the Theory of Lévy Flights, Anomalous Transp. Found. Appl. (2008) 129–162. doi:10.1002/9783527622979.ch5.

[145] X.S. Yang, S. Deb, Multiobjective cuckoo search for design optimization, Comput. Oper. Res. 40 (2013) 1616–1624. doi:10.1016/j.cor.2011.09.026.

[146] Y. Chen, Research and simulation on Levy Flight model for DTN, Proc. - 2010 3rd Int. Congr. Image Signal Process. CISP 2010. 9 (2010) 4421–4423. doi:10.1109/CISP.2010.5647905.

[147] A.M. Edwards, R.A. Phillips, N.W. Watkins, M.P. Freeman, E.J. Murphy, V. Afanasyev, S. V. Buldyrev, M.G.E. Da Luz, E.P. Raposo, H.E. Stanley, G.M. Viswanathan, Revisiting Lévy flight search patterns of wandering albatrosses, bumblebees and deer, Nature. 449 (2007) 1044–1048. doi:10.1038/nature06199.

[148] G.M. Viswanathan, V. Afanasyev, S. V. Buldyrev, E.J. Murphy, P.A. Prince, H.E. Stanley, Lévy flight search patterns of wandering albatrosses, Nature. 381 (1996) 413–415. doi:10.1038/381413a0.

[149] I. Rhee, M. Shin, S. Hong, K. Lee, S.J. Kim, S. Chong, On the levy-walk nature of human mobility, IEEE/ACM Trans. Netw. 19 (2011) 630–643. doi:10.1109/TNET.2011.2120618.

[150] G. Terdik, T. Gyires, Lévy flights and fractal modeling of internet traffic, IEEE/ACM Trans. Netw. 17 (2009) 120–129. doi:10.1109/TNET.2008.925630.

[151] W.A. Hussein, S. Sahran, S.N.H.S. Abdullah, Patch-Levy-based initialization algorithm for Bees Algorithm, Appl. Soft Comput. 23 (2014) 104–121. doi:10.1016/j.asoc.2014.06.004.

[152] H. Hakli, H. Uğuz, A novel particle swarm optimization algorithm with Levy flight, Appl. Soft Comput. J. 23 (2014) 333–345. doi:10.1016/j.asoc.2014.06.034.

[153] X.-S. Yang, Firefly Algorithm, Lévy Flights and Global Optimization, Springer London, London, 2010. doi:10.1007/978-1-84882-983-1_15.

[154] A.A. Heidari, P. Pahlavani, An efficient modified grey wolf optimizer with Lévy flight for optimization tasks, Appl. Soft Comput. J. 60 (2017) 115–134. doi:10.1016/j.asoc.2017.06.044.

[155] X.-S. Yang, Engineering optimization: An introduction with metaheuristic applications, First edit, John Wiley & Sons, New Jersey, 2010.

[156] S. Das, P.N. Suganthan, Differential evolution: A survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (2011) 4–31. doi:10.1109/TEVC.2010.2059031.

[157] S. Mirjalili, Moth-flame optimization algorithm: A novel nature-inspired heuristic paradigm, Knowledge-Based Syst. 89 (2015) 228–249. doi:10.1016/j.knosys.2015.07.006.

[158] S. Mirjalili, The ant lion optimizer, Adv. Eng. Softw. 83 (2015) 80–98. doi:10.1016/j.advengsoft.2015.01.010.

[159] E.H. Satorius, S.T. Alexander, Channel Equalization Using Adaptive Lattice Algorithms, IEEE Trans. Commun. COM-27 (1979) 899–905. doi:10.1109/TCOM.1979.1094477.

[160] H. Nenavath, R.K. Jatoth, Hybridizing sine cosine algorithm with differential evolution for global optimization and object tracking, Appl. Soft Comput. J. 62 (2018) 1019–1043. doi:10.1016/j.asoc.2017.09.039.

[161] C.M. Topaz, A.J. Bernoff, S. Logan, W. Toolson, A model for rolling swarms of locusts, Eur. Phys. J. Spec. Top. 157 (2008) 93–109. doi:10.1140/epjst/e2008-00633-y.

[162] R. Jensi, G.W. Jiji, An enhanced particle swarm optimization with levy flight for global optimization, Appl. Soft Comput. J. 43 (2016) 248–261. doi:10.1016/j.asoc.2016.02.018.

[163] X.S. Yang, S. Deb, Multiobjective cuckoo search for design optimization, Comput. Oper. Res. 40 (2013) 1616–1624. doi:10.1016/j.cor.2011.09.026.

[164] Engineering Optimization an Introduction with Metaheuristic Applications, n.d.

[165] H. Zhao, J. Zhang, Functional link neural network cascaded with Chebyshev orthogonal

polynomial for nonlinear channel equalization, Signal Processing. 88 (2008) 1946–1957. doi:10.1016/j.sigpro.2008.01.029.

[166] J.C. Patra, W.C. Chin, P.K. Meher, G. Chakraborty, Legendre-FLANN-based nonlinear channel equalization in wireless communication system, Conf. Proc. - IEEE Int. Conf. Syst. Man Cybern. (2008) 1826–1831. doi:10.1109/ICSMC.2008.4811554.

[167] J.C. Patra, P.K. Meher, G. Chakraborty, Nonlinear channel equalization for wireless communication systems using Legendre neural networks, Signal Processing. 89 (2009) 2251–2262. doi:10.1016/j.sigpro.2009.05.004.

[168] G. Dhiman, M. Garg, A. Nagar, V. Kumar, M. Dehghani, A novel algorithm for global optimization: Rat Swarm Optimizer, J. Ambient Intell. Humaniz. Comput. 12 (2021) 8457–8482. doi:10.1007/s12652-020-02580-0.

[169] F. Wilcoxon, Individual Comparisons by Ranking Methods, Biometrics Bull. 1 (1945) 80. doi:10.2307/3001968.

[170] V. Vakilian, T. Wild, F. Schaich, S. Ten Brink, J.F. Frigon, Universal-filtered multi-carrier technique for wireless systems beyond LTE, 2013 IEEE Globecom Work. GC Wkshps 2013. (2013) 223–228. doi:10.1109/GLOCOMW.2013.6824990.

[171] X. Wang, T. Wild, F. Schaich, S. Ten Brink, Pilot-aided channel estimation for universal filtered multi-carrier, 2015 IEEE 82nd Veh. Technol. Conf. VTC Fall 2015 - Proc. (2016). doi:10.1109/VTCFall.2015.7391089.

[172] F. Schaich, T. Wild, Waveform contenders for 5G - OFDM vs. FBMC vs. UFMC, in: ISCCSP, IEEE, 2014: pp. 457–460.

[173] V. Durga, S. Anuradha, On Channel Estimation in Universal Filtered Multi-Carrier (UFMC) System, Prog. Electromagn. Res. Symp. 2019-June (2019) 3708–3713. doi:10.1109/PIERS-Spring46901.2019.9017701.

[174] A. Singh, K. Krishna Naik, C.R.S. Kumar, Novel Pilot-Aided Channel Estimation Scheme for Power Domain NOMA-UFMC System in Fading Scenarios, 1st Int. Conf. Range Technol. ICORT 2019. 2 (2019). doi:10.1109/ICORT46471.2019.9069626.

[175] H. Ye, G.Y. Li, B.F. Juang, Power of Deep Learning for Channel Estimation and Signal Detection in OFDM Systems, IEEE Wirel. Commun. Lett. 7 (2018) 114–117. doi:10.1109/LWC.2017.2757490.

[176] Z. Qin, H. Ye, G.Y. Li, B.H.F. Juang, Deep learning in physical layer communications, IEEE Wirel. Commun. 26 (2019) 93–99. doi:10.1109/MWC.2019.1800601.

[177] H. Ye, G.Y. Li, Initial results on deep learning for joint channel equalization and decoding, IEEE Veh. Technol. Conf. 2017-Septe (2018) 1–5. doi:10.1109/VTCFall.2017.8288419.

[178] B. Shamasundar, A. Chockalingam, A DNN Architecture for the Detection of Generalized Spatial Modulation Signals, IEEE Commun. Lett. XX (2020) 1–1. doi:10.1109/lcomm.2020.3018260.

[179] T.O. Shea, S. Member, J. Hoydis, An Introduction to Deep Learning for the Physical Layer, 3 (2017) 563–575.

[180] X. Wang, T. Wild, F. Schaich, S. ten Stephan, Channel Estimation and Equalization for

5G Wireless Communication Systems,Master Thesis, Universitat Stuttgart, 2014.

[181] F. Schaich, T. Wild, Y. Chen, Waveform contenders for 5G - Suitability for short packet and low latency transmissions, IEEE Veh. Technol. Conf. 2015-Janua (2014) 5–9. doi:10.1109/VTCSpring.2014.7023145.

[182] J. Schmidhuber, Deep Learning in neural networks: An overview, Neural Networks. 61 (2015) 85–117. doi:10.1016/j.neunet.2014.09.003.