Hindawi
International Transactions on Electrical Energy Systems
Volume 2023, Article ID 1572996, 18 pages
https://doi.org/10.1155/2023/1572996

Research Article

Finite Impulse Response Filter Design Using Fuzzy Logic-Based Diversity-Controlled Self-Adaptive Differential Evolution

K. Mohaideen Abdul Kadhar,[1] **S. Rengarajan**,[2] **S. Tamilselvi**,[2] **N. Karuppiah**,[3] **Praveen Kumar Balachandran**,[3] **A. Thamilmaran**,[4] **C. Dhanamjayulu**,[4] **and Baseem Khan**[5,6]

[1]*Centre for Computational Imaging and Machine Vision, Department of Electronics and Communication Engineering, Sri Eswar College of Engineering, Coimbatore, India*
[2]*Department of Electrical and Electronics Engineering, Sri Sivasubramaniya Nadar College of Engineering, Chennai, India*
[3]*Department of Electrical and Electronics Engineering, Vardhaman College of Engineering, Hyderabad, India*
[4]*School of Electronics Engineering, Vellore Institute of Technology, Vellore, India*
[5]*Department of Electrical and Computer Engineering, Hawassa University, Hawassa 05, Ethiopia*
[6]*Department of Electrical and Electronic Engineering Technology, Faculty of Engineering and the Built Environment, University of Johannesburg, Johannesburg, South Africa*

Correspondence should be addressed to C. Dhanamjayulu; dhanamjayulu.c@vit.ac.in and Baseem Khan; baseem_khan04@yahoo.com

Received 19 December 2022; Revised 24 May 2023; Accepted 26 June 2023; Published 13 July 2023

Academic Editor: Sheng Du

The design of finite impulse response (FIR) filters involves the estimation of effective filter coefficients, making the designed filter exhibit infinite stopband attenuation and have a flat-shaped passband. The few conventional filter design methods such as impulse response truncation (IRT) and windowing technique exhibit undesirable characteristics owing to the Gibbs phenomenon, thus making them unsuitable for various practical complexities. This research work employs the fuzzy logic-based diversity-controlled self-adaptive differential evolution algorithm (FLDCSaDE) for the design of FIR band stop (BS) and high pass (HP) filters. In order to validate the results of the proposed technique, various population-based evolutionary computing techniques such as the covariance matrix adaptation evolution strategy (CMAES), differential evolution (DE), self-adaptive differential evolution (SaDE), and Jaya algorithm have also been applied for determining the effective filter coefficients. The performance of the various algorithms has been analysed and compared based on the parameters such as stopband attenuation, passband attenuation, and ripples. The simulation results show that the FLDCSaDE algorithm outperforms other evolutionary algorithms having 4% and 1.5% lower ripples than the SaDE algorithm for high pass and band stop filters, respectively. Experimental results depict that the performance of the fuzzy approach causes positioning and tracking accuracy obtained to be improved by 27% and the corresponding false positive rate (FPR) is substantially reduced to 0.11 from the mean amplitude value obtained from the fuzzy approach in the frequency response. The frequency response obtained from the FLDCSaDE algorithm is close to the ideal response of the BS and HP FIR filters.

1. Introduction

A signal can be termed as a physical parameter that changes with space, time, or any other quantity. In general, signals are responsible for the movement of information which sprouts in almost every field of science and technology. Signals are categorized into two sections, namely, discrete and continuous-time signals, depending on the nature of space and time. Given that most of the inherently occurring signals are continuous in nature (analog), an effective filtering method of analog signals is a necessity. One such method of filtering is through digital signal processing (DSP).

Filters are the most extensively used systems for processing signals that are digital in nature. This can be employed to obtain the needed output spectral attributes by altering the input signal spectrum. Digital filters play a critical role in the communication and processing of signals. The main advantage of digital filters is higher reliability, accuracy, and lower tolerances with lower noises leading to increased efficiency. They are robust to changes in characteristics which make them suitable for variable multiple applications such as voice signal synthesis and analysis [1], image processing [2], removing random noise from seismic data [3], and rectifying many biomedical signals [4] such as EEG, MRIs, and ECGs. Digital filters can be further classified into IIR and FIR filters based on the response. FIR filters exhibit a linear phase which is inherently stable, nonrecursive in nature, and less sensitive in utilizing bounded word duration effects. The above-mentioned attributes make FIR filters more desirable despite the requirement of numerous coefficients as compared to infinite impulse response (IIR) filters where the memory requirement is high.

There are several traditional techniques available with the notion of designing FIR filters, among which the windowing approach is the most preferred technique. Selection of a suitable window function such as Hamming, Hanning, and Kaiser is dependent on several factors with respect to the desired frequency response such as variation in stopband attenuation, transition in width, and ripples in stopband and passband [5]. Frequency domain convolutions carried out in the windowing technique results in tapered rectangular edges, leading to ripples in passband and stopband and in turn limit the performance of the given FIR filter. The obtained ripple is not uniformly distributed across a given rectangular window but in turn follows a pattern in which ripples decay as we move away from the path along the discontinuous points according to the side-lobe pattern observed from the window. Tolerance to given ripple behaviour can be observed by allowing more freedom to the stated ripple, thus reducing the complexity and improving the robustness of the given filter. However, in this way, the order of the filter gets reduced and it affects the transition from passband to stopband, thus deteriorating signal strength by interference [6].

In contrast to the traditional method, procedures involved in the digital FIR filter designing process could be conceived to be an optimization problem [7], alongside the intention to minimize the error function, which basically indicates an inconsistency in the filter designed from the intended response [8]. Classical methods such as the least square and gradient-based methods, optimizing L_1 and L_2 norms, can provide better passband response and high stopband attenuation, with minimal ripples [9, 10]. However, these gradient-based methods can result only in local optimal solution and cannot support multiobjective optimization problems. Furthermore, these gradient descent methods have difficulties in handling large dimension problems, due to the irregularities in the approximate gradients.

Due to the drawbacks of classical gradient-based methods, in recent times, many researchers have started to implement evolutionary and swarm-based algorithms to get global optimal solutions in designing digital FIR filters. The use of evolutionary algorithms and intelligent swarm techniques are very well suited for handling non-differentiable and multiobjective functions [11]. It is important in laying the foundation for understanding filter design with the use of different possible solution algorithms, namely, the search algorithms. Both the single objective and multiobjective formulations have their own set of advantages, wherein the single objective problems have lower constraints and complexity and the multiobjective formulation has better accuracy. The adaptation of these formulations depends on the nature of the problem in consideration [12]. Many evolutionary and swarm techniques are developed in designing the FIR filter, and they are discussed as follows.

The evolutionary process-based algorithms such as simulated annealing and genetic algorithms are implemented in FIR filter design [13]. In 2015, the simulated annealing method was employed to design the FIR filter with the objective of attaining a desired magnitude response of the filter [14]. Moreover, various variants of the genetic algorithms (GAs) such as real-coded GA and hybrid GA are also implemented to design the FIR filter [9, 13, 15].

The swarm intelligence-based metaheuristic algorithms, one of which is called particle swarm optimization (PSO), was developed in the year 2016 [16]. Its greatest advantage is its simplicity in implementation and conceptual level, leading to highly utilized computational algorithms for the filter design. References [17–20]. The linear phase FIR filter has been designed using PSO and GA by considering the feasible passband and stopband frequencies and the size of the passband and stopband ripples [21]. Craziness-based PSO was implemented in designing the 28^{th} and 36^{th} order band stop filter [22, 23]. The cat swarm optimization algorithm is applied to design the linear phase of the FIR filter to meet the desired frequency response characteristics [24]. The linear phase multiband stop filter for the order of 40, 48, and 58 is developed using an improved cuckoo search PSO. The adaptive cuckoo search algorithm (ACSA) is also developed to design a digital FIR filter design [25].

Further research through the years led to the development of much more refined techniques for estimating the filter coefficients. The algorithms so found had to inherently displace the issue of global search space and replace it with the local search mechanism. One such algorithm was developed in 2005 and was called the covariance matrix adaptation evolution strategy (CMAES). It generates renewed population members by sampling from the probability distribution of a given search space. The use of correlations between filter coefficients accelerates the convergence process because of the absence of derivatives [26]. However, when compared to differential evolution (DE), the CMAES algorithm suffers from the nonstationary error that increases the noise component in the filter which if present in the magnitude response may remain undetected, thus decreasing the efficiency of the filter. The use of eigenvectors and covariance matrix to find the frequency response of the given filter increases the time complexity [27], especially for higher order matrix where the optimization of the filter coefficients becomes a complicated process.

The Jaya algorithm solves the abovementioned problem as it is free of gradient, free of algorithm-specific parameters, and the time complexity for filter optimization is much lesser for this algorithm. In terms of reducing mixed noise or ripples in the filter, Jaya outperformed other algorithms as shown in reference [28]. However, the convergence rate is not as efficient when compared to CMAES as its performance benchmark is lesser when applied to optimizing filter coefficients.

Furthermore, the FIR filter design has been designed by improving the candidate solution iteratively based on the evolutionary process using DE, proposed in 2020 [29]. The major advancement in DE is finding the true global minimum of a model search space from the coefficient set regardless of the initial parameter values [30], thus making the initialization process independent of the delay. In contrast to GA, the DE algorithm uses mutation operation, thus utilizing fewer control parameters for filter responses, namely, the magnitude responses for stopband and passband regions [31]. Low pass digital FIR filter parameters are identified using the DE algorithm by minimizing the least mean square function [32]. The FIR filter has been designed using DE in consideration of different word lengths, and the same is implemented using Cadence RTL Compiler (UMC 90 nm technology) [33]. However, there are inherent disadvantages of the decreasing search spaces because of the diversity of the possible movements in exploration. The performance of DE suffers due to the predefined control parameters (CO and F) and mutation strategy. If the problem differs, mutation strategy and control parameters are to be fine-tuned for obtaining the consistent optimization performances. Due to this drawback of DE, many adaptive DE algorithms such as self-adaptive DE [34], DE [35], opposition-based DE [36], composite DE [37], random neighbours-based DE [38], and neighbourhood mutation and opposition-based learning (NBOLDE) [39] have been proposed recently. The nuances of the different optimization algorithms are summarized in Table 1. The basic understanding of the processes and parameters to determine the optimal filter coefficients were analysed. The DE algorithm had an interesting approach to improving candidate solution and minimizing it using elitist replacement. This methodology of DE and SaDE is combined to avoid using the predefined parameters to improve the performance.

This paper proposes fuzzy logic-based diversity-controlled self-adaptive differential evolution (FLDCSaDE) for the first time to design the FIR filter by overcoming the accuracy and performance limitations from the previous literature. The superiority of the FLDCSaDE algorithm over other algorithms is analysed in terms of its ripple content, performance, attenuation, and others. The algorithm overcomes the inherent problems of the evolutionary algorithm when designing the filter, namely, the dependency among the filter parameters, improper initialization of control parameters, and premature convergence. The abovediscussed DE, CMAES, Jaya, SaDE, and FLDCSaDE algorithms are used in constrained and unconstrained real-life problems as their parameter requirements are much lesser than other algorithms, and hence, their computation makes their limitations negligible. FLDCSaDE employs fuzzy logic mainly for two purposes. One is to keep the diversity in the population, and the other is to obtain weights that would help narrow down the search space, thus finding the optimal filter parameters at a much faster rate. The designed filter possesses minimum magnitude error and ripples in passband and stopband. The general design procedure involved in the design of the FIR filter using the evolutionary algorithm is discussed in Figure 1.

Although the conventional SaDE algorithm is used to generate success and failure rates based on four mutation strategies [40], it requires a total of K_n strategies for choosing an optimal value for updating the CO value. The use of fuzzy logic makes it feasible to update CO values based on the diversity of the population, thus surpassing the performance of SaDE and removing numerous types of noise and harmonics. Proper tuning of filter coefficients dynamically gives a way for better performance and at the same time improves the trade-off between adaptation and robustness. In case of improper tuning, the parameters increase the computational effort and complexity [39]. The core notion of employing local search in the FLDCSaDE algorithm helps strike a balance between the two important driving forces imperative for any optimization algorithm which is exploitation and exploration. It is not required to look for optimal filter coefficients in each iteration since the search can be applied to any randomly chosen individual in the given population, and if the solution obtained is better than the one obtained in the previous generation, the individual is included in the given solution set, thus enhancing the time complexity. This helps obtain feasible FIR filter coefficients having an execution time much lesser than that of SaDE and lower ripple content, and thus, higher efficiency shows that this algorithm can be applied to obtain the optimal FIR filter response.

2. Methodology

The methodology for the implementation of the design of the finite impulse response (FIR) filter has been listed as per the sequence in the following section:

(1) Determination of objective function.

(2) Comparison of optimization algorithms.

(3) Evaluating the best fit.

(4) Integrating the abovementioned algorithm to high pass (HP) and band stop (BS) filter.

(5) Introducing constraints and determining the default initial values.

(6) Finding the errors (both stop and passband) and amplitude. Repeating step 5 until the errors are minimized.

This research paper is organized beginning with the formulation of the filter design, objective function, and constraint parameters in Section 3. It is followed by an in-

Table 1: Review of recent literature works.

S. No.	Author	Title	Source	Findings
1	Rawat TK, 2014	Digital signal processing	1st ed. Oxford University Press	The book gives the basic understanding of the filter parameters and their impact on the frequency response
2	Reddy KS, Sahoo SK. 2015	An approach for FIR filter coefficient optimization using differential evolution algorithm	AEU-International Journal of Electronics and Communications	The filter design was performed using DE algorithm where the coefficients were compared for various word lengths
3	Aggarwal, Apoorva and Rawat, Tarun and Upadhyay, Dharmendra. 2018	Design of optimal digital FIR filters using evolutionary and swarm optimization techniques	AEU-International Journal of Electronics and Communications	The employed optimization techniques are compared based on the execution time, convergence test and optimal parameters obtained
4	Manolis Georgioudakis, Vagelis Plevris, 2020	On the performance of differential evolution variants in constrained structural optimization	Procedia manufacturing, Volume 44	The constrained optimization problem analysis applications of five DE variants on the real-world applications from the convergence history

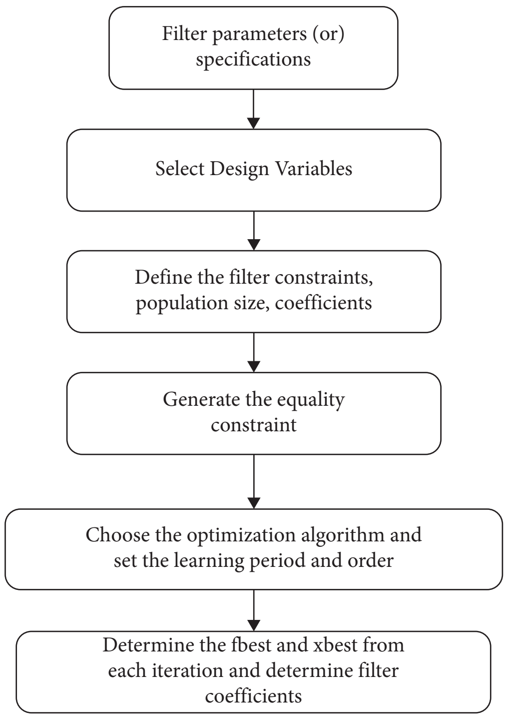

FIGURE 1: FIR filter design using the optimization algorithm.

depth analysis of the various evolutionary algorithms having performance similar to the proposed FLDCSaDE, namely, DE, CMAES, Jaya, and SaDE in Section 4. Section 5 discusses the implementation of FLDCSaDE for the FIR filter design. Based on the simulations, the results obtained are tabulated for various evolutionary algorithms in Section 5, followed by the analysis and conclusion at the end.

3. FIR Filter Design

FIR-based filters are adopted for many practical applications because of their properties such as linear phase and inherent stability over IIR-based filters. At the forefront of designing an N^{th} order FIR high pass filter which is optimal, the desired response of the filter $H(e^{j\omega})$ is compared with the ideal feedback $H_d(e^{j\omega})$, specified as follows:

$$H_d\left(e^{j\omega}\right) = \begin{cases} 0, & \omega \in \left(0, \omega_c\right) \text{ stopband}, \\ 1, & \omega \in \left(0, \omega_c\right) \text{ passband}, \end{cases} \tag{1}$$

where the cut-off frequency of the desired filter is given by ω_c. Utilizing the inverse discrete time-based Fourier transform (IDTFT) response of the given ideal filter and multiplying the result with a window function gives the impulse or delta response obtained for the filter $h(n)$. Taking the discrete-time Fourier transform (DTFT) for the given impulse response $h(n)$ into account, the desired filter response $H(e^{j\omega})$ for the given FIR filter could be attained as follows:

$$H\left(e^{j\omega}\right) = \sum_{n=1}^{N} h(n)e^{-j\omega n}, \tag{2}$$

where the order of the given filter is given by N. Basically, for a linear phase type-I filter having coefficients which are symmetric and have bounded length, $h(n) = h(N-1-n)$, where $0 \le n \le N-1$, have the magnitude response, which is given by the following equation:

$$H_r\left(e^{j\omega}\right) = h[M] + 2\sum_{n=0}^{M-1} h(n)\cos((M-n)\omega), \tag{3}$$

where $H_r(e^{j\omega})$ is the function with a real-valued response and $M = (N-1)/2$. Through the process of limiting $H_r(e^{j\omega})$ with the response obtained for the ideal filter, the error function $E(\omega)$ can be found. By the defined L_2 norm given as the standard for designing a high pass filter, the error function $E(\omega)$ can be stated as follows:

$$|E(\omega)|_2 = \int_0^{\pi} \left|H_r\left(e^{j\omega}\right) - H_d\left(e^{j\omega}\right)\right|^2 d\omega. \tag{4}$$

This paper aims at deriving the response for the stated filter through the utilization of the aforementioned error function $E(\omega)$ as an objective function in the optimization problem. The main objective is to obtain a series of optimized symmetric coefficients that minimize the error function of the filter and to use those coefficients in designing a linear phase type-1 FIR filter for band stop and high pass modes. The error function can be minimized by adopting any evolutionary algorithm to find the optimal filter coefficients for the design of the FIR band stop and high pass modes.

4. Evolutionary Algorithms

This section deals with the steps involved in the optimization process of various evolutionary algorithms applied in this paper.

4.1. Covariance Matrix Adaptation Evolution Strategy (CMAES). Nikolaus Hansen developed an efficient evolution algorithm used as an optimizing tool for continuous search spaces with real objectives [41]. This algorithm was termed as the covariance matrix adaptation evolution strategy (CMAES). It is a derivative-free stochastic approach used specifically for nonlinear problems [42], unlike other methods which are being based on derivatives failing to yield a solution due to various reasons such as improper search with sharp turns, numerous breaks in continuity, and local optimal solutions. The CMAES algorithm is continuous in nature and can be used for various applications involving complex optimization numerical problems at a search region which is continuously nonconvex and nonlinear. It utilizes multivariate distribution samples $\mathbf{N}(\mathbf{m}, \mathbf{C})$ which are constructed with the help of its mean value and its covariance matrix; it is symmetric, positive, and definite in nature during the process of optimization. These samples can be taken into account for obtaining renewed members of the population. The covariance matrix and the mean involving the given set of samples point out to the limits in the region of $\mathbf{m} \in \mathbf{R^n}$ and $\mathbf{C} \in \mathbf{R^{n\times n}}$, respectively. In the previous generation, by updating the value of "m," which corresponds to the translational displacement of the distribution, the possibility of obtaining the optimal solution is maximized. The covariance matrix can be represented geometrically, with a distinct shape such as that of an isodensity ellipsoid. The shape of this geometric representation, whose axis has length, corresponds to the respective eigenvalues of the covariance matrix.

The algorithm uses two different adaptation strategies, namely, CMA and step size adaptation. The numerous steps used during the implementation of the algorithm are specified as follows:

Step 1. Sampling population

The search positions are found through the process of sampling a distribution consisting of multiple variables. For every search point belonging to the generation "g," we find the step size ($\sigma^{(g)}$), covariance matrix $C^{(g)}$, and the mean value $m^{(g)}$. From the abovementioned process, the new variables or individuals obtained are again sampled at the subsequent generation $(g+1)$ as follows:

$$X_k^{(g+1)} = m^{(g)} + \sigma^{(g)} Ns\left(0, C^{(g)}\right) \text{for } k = 1, \ldots, Ns, \tag{5}$$

where $X_k^{(g+1)}$ belongs to the $g+1$ generation and is the k^{th} offspring belonging to the generation $g+1$; population size is denoted by Ns.

Step 2. Recombination along with the selection process

After completion of the sampling process, the new mean $m^{(g+1)}$ is obtained by selecting the top μ (weighted average) samples having the highest fitness among the given total population size (Ns). The new mean becomes the weighted average of μ samples with the weight parameter w_i and is given by the following equation:

$$m^{(g+1)} = \sum_{i=1}^{\mu} w_i X_{i:\,Ns}^{(g+1)}, \tag{6}$$

where $\mu \leq Ns$ denotes the original population size, and subsequently, by equating $w_i = 1/\mu$, we calculate the mean for all μ samples and $X_{i:\,Ns}^{(g+1)}$ denotes that among the given (Ns) sampling points, it is the i^{th} best ranked individual.

Step 3. Adaptation of the covariance matrix C

There are two more reliable but complex methods to update C, which are as follows:

(i) Using Polyak's average to estimate the rank-μ update of C:

It is based on updating the value of C using the previous history,

$$\begin{aligned}\mathbf{C}^{(\mathbf{g+1})} &= (\mathbf{1} - \mathbf{c}_{\mathbf{cov}})\mathbf{C}^{(\mathbf{g})} + \mathbf{c}_{\mathbf{cov}} \left\{ \sum_{\mathbf{i=1}}^{\boldsymbol{\mu}} \mathbf{w}_{\mathbf{i}} \left(\frac{\mathbf{X}_{\mathbf{i:\,Ns}}^{(\mathbf{g+1})} - \mathbf{m}^{(\mathbf{g})}}{\boldsymbol{\sigma}^{(\mathbf{g})}} \right) \right. \\ &\quad \left. \cdot \left(\frac{\mathbf{X}_{\mathbf{i:\,Ns}}^{(\mathbf{g+1})} - \mathbf{m}^{(\mathbf{g})}}{\boldsymbol{\sigma}^{(\mathbf{g})}} \right)^{\mathbf{T}} \right\} \\ &= (\mathbf{1} - \mathbf{c}_{\mathbf{cov}})\mathbf{C}^{(\mathbf{g})} + (\mathbf{c}_{\mathbf{cov}}) C_{\lambda}^{(g+1)}.\end{aligned} \tag{7}$$

(ii) Utilizing the evolution path–cumulation:

The second way is using an evolution path (p_c^g), to log symbol information, (P_c) can be calculated using the standard conditions defined initially $C^{(0)} = \mathrm{I}, p_c^{(0)} = 0, \sigma^{(0)} = 0.25\,(x_{t,\max} - x_{t,\min})$, as

$$\mathbf{p}_{\mathbf{c}}^{(\mathbf{g+1})} = (\mathbf{1} - \mathbf{c}_{\mathbf{c}})\mathbf{p}_{\mathbf{c}}^{(\mathbf{g})} + \sqrt{\boldsymbol{\mu}_{\mathbf{eff}} \mathbf{c}_{\mathbf{c}} (\mathbf{2} - \mathbf{c}_{\mathbf{c}})} \left(\frac{\mathbf{m}^{(\mathbf{g+1})} - \mathbf{m}^{(\mathbf{g})}}{\boldsymbol{\sigma}^{(\mathbf{g})}} \right). \tag{8}$$

We can use P_c to update the covariance matrix C:

$$\mathbf{C}^{(\mathbf{g+1})} = (\mathbf{1} - \mathbf{c}_{\mathbf{cov}})\mathbf{C}^{(\mathbf{g})} + \mathbf{c}_{\mathbf{cov}} \mathbf{p}_{\mathbf{c}}^{(\mathbf{g+1})} \mathbf{p}_{\mathbf{c}}^{(\mathbf{g+1})^{\mathbf{T}}}, \tag{9}$$

$$\text{Also, } f(x) = \frac{1}{2}(x - x^*)^T H (x - x^*). \tag{10}$$

(iii) Combining two methods:

Using the updated values of the covariance matrix from the two previous methods, we combine the values to derive the renewed formula for updating the final CMA for the covariance matrix $(C^{(g+1)})$ by using equations (7) and (10), with $\mu_{\text{cov}} \geq 1$, rank-one update and weighting between rank-μ:

$$\begin{aligned}\mathbf{C}^{(\mathbf{g+1})} &= (\mathbf{1} - \mathbf{c}_{\mathbf{cov}})\mathbf{C}^{(\mathbf{g})} + \frac{\mathbf{c}_{\mathbf{cov}}}{\boldsymbol{\mu}_{\mathbf{cov}}} \mathbf{p}_{\mathbf{c}}^{(\mathbf{g+1})} \mathbf{p}_{\mathbf{c}}^{(\mathbf{g+1})^{\mathbf{T}}} \\ &\quad + \mathbf{c}_{\mathbf{cov}} \left(\mathbf{1} - \frac{\mathbf{1}}{\boldsymbol{\mu}_{\mathbf{cov}}} \right) \sum_{\mathbf{i=1}}^{\boldsymbol{\mu}} \mathbf{w}_{\mathbf{i}} \left(\frac{\mathbf{X}_{\mathbf{i:\,Np}}^{(\mathbf{g+1})} - \mathbf{m}^{(\mathbf{g})}}{\boldsymbol{\sigma}^{(\mathbf{g})}} \right) \\ &\quad \cdot \left(\frac{\mathbf{X}_{\mathbf{i:\,Np}}^{(\mathbf{g+1})} - \mathbf{m}^{(\mathbf{g})}}{\boldsymbol{\sigma}^{(\mathbf{g})}} \right)^{\mathbf{T}}.\end{aligned} \tag{11}$$

The eigen composition of the covariance matrix obeys the following:

$C^{(g)} = B^{(g)}(D^{(g)})^2(B^{(g)})^T$; $\mu_{\text{eff}} = 1/(\sum_{i=1}^{\mu} w_i^2)$ is given to the variance for the given optimized mass; $c_c = 4/(n+4)$ is denoted as the learning ratio which can be used to find the combined effective step for the evolution path; μ_{cov} defined a constraint for rank-μ update and rank-one weighting.

$c_{\text{cov}} = \left\{ 1/\mu_{\text{cov}} 2/(n+\sqrt{2})^2 + (1 - 1/\mu_{\text{cov}}) \min(1, (2\mu_{\text{cov}} - 1)/(n+2)^2 + \mu_{\text{cov}}) \right\}$ gives the learning rate of the updated covariance matrix $C^{(g)}$, $B^{(g)} = n * n$ orthogonal matrix, and $D^{(g)} = n * n$ diagonal matrix. During the usage of population sizes for small offspring, the rank-one update is found to be inefficient as it minimizes the evaluations of the functions [43].

Step 4. Controlling the given step-size

For the purpose of updating the global step size $(\sigma^{(g)})$, we utilize the relation between the mean trajectory denoted by $(p_\sigma^{(g)})$ [44].

The evolution path for adaptive step size is computed by

$$\mathbf{p}_\sigma^{(g+1)} = (1 - \mathbf{c}_\sigma)\mathbf{p}_\sigma^{(g)} + \left\{ \sqrt{(\mathbf{c}_\sigma(2 - \mathbf{c}_c)\boldsymbol{\mu}_{\mathrm{eff}})}\,\mathbf{B}^{(g)}\mathbf{D}^{(g)^{-1}}\mathbf{B}^{(g)^T}\left(\frac{\mathbf{m}^{(g+1)} - \mathbf{m}^{(g)}}{\boldsymbol{\sigma}^{(g)}}\right)\right\}, \tag{12}$$

where, $\mathbf{c}_\sigma$ is defined as the learning for the given step size which is found from P_σ, the path movement. Step size ($\boldsymbol{\sigma}^{(g)}$) is linked to the conjugate evolution path ($\mathbf{p}_\sigma^{(g)}$) and is given as follows:

$$\boldsymbol{\sigma}^{(g+1)} = \boldsymbol{\sigma}^{(g)} \exp\left(\frac{\mathbf{c}_\sigma}{\mathbf{d}_\sigma}\left(\frac{\|\mathbf{p}_\sigma^{(g+1)}\|}{\mathbf{E}\|\mathbf{N}(\mathbf{0},\mathbf{I})\|} - \mathbf{1}\right)\right), \tag{13}$$

where I is an Identity matrix; $N(0, I)$ denotes the normal distribution with unity covariance matrix and zero mean. $E||N(0, I)||$ defines the expectation of the Euclidean normal distribution of $N(0, I)$ with the distributed random vector initialized to zero $p_\sigma^{(0)} = 0$ initially. $c_\sigma = 10/(n + 20)$ denotes the timer horizon of evolution in the backward path and also $c_\sigma = \mu_{\mathrm{eff}} + 2/n + \mu_{\mathrm{eff}} + 3$; the step-size damping factor is given by $d_\sigma = 1 + 2\max(0, \sqrt{\mu_{\mathrm{eff}} - 1/n + 1}) + c_\sigma$.

4.2. Jaya Algorithm. The Jaya algorithm is a new population-based metaheuristic optimization algorithm that is used to determine the optimal subset of features to improve the performance of the classification process. It combines the features of evolutionary algorithms and swarm-based intelligence. The Jaya algorithm has the tendency to move to the best, i.e., it is closer to success, and avoids the worst solution obtained in the iteration [45]. This nature makes this algorithm victorious, and hence, the name Jaya is defined. It is derived from a Sanskrit word meaning "*victory*" [46]. The advantage of the algorithm over other metaheuristic algorithms is that it requires only a few control parameters such as the maximum number of generations, population size, and the number of design variables that are common for all algorithms, and it is independent of algorithm-specific parameters; therefore, it does not require extensive tuning, so we can avoid unwanted convergence and reduce the computational costs.

The primary objective of this algorithm is to minimize/maximize an objective function $f(x)$.

The following steps are followed in the implementation of the Jaya algorithm:

Step 1: initializing the decision variable (x_i), population size (N_p), and iteration number (T). The decision variable is initialized by a value between the lower and upper bound ranges such that $x_i \in [X_i^{\min}, X_i^{\max}]$.

Step 2: creating a job preference vector $V_{k,l} = \{v_{1,k,l}, v_{2,k,l}, \ldots, v_{i,k,l}, \ldots, v_{n,k,l}\}$, where $V_{k,l}$ is an n-dimensional vector which represents a sequence of jobs in the kth schedule at the lth iteration and $v_{i,k,l}$ is the preference value assigned to an ith job in the kth schedule at the lth iteration. The preference value is randomly generated with a uniform random number distribution as follows:

$$v_{i,k,l} = 1 + \mathrm{rand}\,(0,1) * (n - 1), \tag{14}$$

where rand is a uniform function that generates a random number between 0 and 1 and

$$i \in \{1,2,3,\cdots,n\}, k \in \{1,2,3,\cdots,N_p\}, l \in \{1,2,3,\cdots,N_p\}. \tag{15}$$

Step 3: the objective function $f(x_i)$ for each solution is calculated, and the job preference vector is sorted in the ascending order based on the objective function values [47]. The minimum and the maximum objective functions are determined, and their corresponding best preference vectors and the worst preference vectors are $V_{k,\mathrm{lmin}}$ and $V_{k,\mathrm{lmax}}$, respectively.

Step 4: Updating the preference values of all vectors based on the preference values of ($V_{k,\mathrm{lmin}}$ and $V_{k,\mathrm{lmax}}$), respectively.

$$\begin{aligned} V_{i,k,l} = v_{i,k,l} + r_{i,k,l} * \left(v_{i,k,\mathrm{lmin}} - |v_{i,k,l}|\right) \\ - r_{2,i,l} * \left(v_{i,k,\mathrm{lmax}} - |v_{i,k,l}|\right), \end{aligned} \tag{16}$$

where $V_{i,k,l}$ is the preference value for the ith job in the kth schedule at the lth iteration.; $V_{i,k,\mathrm{lmin}}$ and $V_{i,k,\mathrm{lmax}}$ are the preference values of the ith job in kth schedule with the minimum and maximum values of the objective function.; and $r1$ and $r2$ are random values generated with the uniform distribution function U [0,1], which are used to achieve the right balance between the exploration and exploitation processes [48].

Step 5: converting the values to a new preference value and identifying the corresponding new objective function value

Step 6: comparing the values of the new schedule with the previous schedule and updating it with the better solution

Step 7: repeating the abovementioned process until an optimal solution is obtained and the current solution is replaced by the optimal solution

The viability and efficiency of the Jaya algorithm make it suitable for real-world problems such as feature selection, image processing, designing PID controllers, and many other applications [49–51]. The algorithm is used to optimize multiple-objective cases such as (i) minimizing the total operating cost, (ii) minimizing the system loss, and (iii) minimizing voltage deviation. In addition, the algorithm is investigated to satisfy the multiobjective cases such as (i) minimizing the total cost and system loss, (ii) minimizing the total cost and voltage deviation, and (iii) minimizing the system loss and voltage deviation.

4.3. Self-Adaptive Differential Evolution (SaDE). In the SaDE algorithm, the population is initialized randomly with NS (population size) target vectors, strategy probability, number of strategies available, and learning period (LP). Fitness for the population is evaluated for the parent selection process. The mutation operator is enforcing a very objective vector in the existing period to bring about the evolution vector. The crossover process is enforced on every combination of objective vectors and uses the respective mutated vector to create a preliminary vector combination. For offspring generation, the SaDE algorithm mainly calculates (i) strategy probability and (ii) assigns control parameters to the process of obtaining preliminary vectors for each objective vector. The choice of three control constants such as crossover rate (CO), mutation rate (M), and population size (NS) in successive generations highlights the SaDE algorithm from the conventional DE algorithm by providing randomness [52], which in turn helps enhance the measure of searching for exploitation and exploration.

Thus, the strategy candidate pool has been divided into four different strategies; the first three strategies use the binomial-crossover operator, and the fourth strategy generates a trial vector without a solitary crossover.

(1) DE/rand/1/bin (ST1)

$$U_{i,j=}\begin{cases} x_{r1,j} + M * \left(x_{r2,j} - x_{r3,j}\right), & \text{if rand } [0,1) < \text{COorj} = j_{\text{rand}}, \\ x_{i,j}, & \text{Otherwise.} \end{cases} \tag{17}$$

(2) DE/rand-to-best/2/bin (ST2)

$$U_{i,j} = \begin{cases} x_{i,j} + M * \left(x_{\text{best},j} - x_{i,j}\right) + M * \left(x_{r1,j} - x_{r2,j}\right) + M * \left(x_{r3,j} - x_{r4,j}\right), & \text{if rand}[0,1) < \text{COorj} = j_{\text{rand}}, \\ x_{i,j}, & \text{Otherwise.} \end{cases} \tag{18}$$

(3) DE/rand-to-best/2/bin (ST3)

$$U_{i,j} = \begin{cases} x_{r1,j} + M * \left(x_{r2,j} - x_{r3,j}\right) + M * \left(x_{r4,j} - x_{r5,j}\right), & \text{if rand}[0,1) < \text{COorj} = j_{\text{rand}}, \\ x_{i,j}, & \text{Otherwise.} \end{cases} \tag{19}$$

(4) DE/current-to-rand/1 (ST4)

$$U_{i,j=}X_{i,g} + k * \left(X_{r1,g} - X_{i,g}\right) + M * \left(X_{r2,g} - X_{r3,g}\right). \tag{20}$$

In the SaDE technique, $x_{r1}, x_{r2}, x_{r3}, x_{r4}$, and x_{r5} are the distinct solutions for the current population. Population size (NS) is not needed to be tuned, and common values should be attempted based on the intricacy of the application under consideration. The scaling factor, M, is firmly linked to the concurrent speed. Proper selection of CO results in enhancing the efficiency of the optimization, while incorrectly selected values may worsen the efficiency. Hence, for a given problem, there is a gradual adjustment in the prior values so that they are in the range and have entered the next generation successfully. CO is usually spread across a spectrum with a mean (COmk) and a standard deviation of 0.1 analogous to the kth strategy. In the beginning, (COmk) is assigned to be 0.5 for fundamental LP generation and is repeated for all strategies. Subsequently, after LP iterative generations, (COmk) is assigned to be the median of the successful CO values. Post evaluation of the latest obtained objective vectors, CO variables in (COmk) belonging to the previously defined generation are substituted with the numeric values found in the most recent generation for the k^{th} outcome.

A single outcome for obtaining the objective vector is taken from one of the efficient strategies assigned to each target vector in order to obtain an objective vector. From the knowledge of previous candidate solutions, the SaDE algorithm adapts to the strategies for generating objective vectors and related control parameter settings by oneself during evolution. As a result, a better and more appropriate outcome for obtaining is found and the various arguments are found by adjusting the control parameters to match at various stages during the optimization problem [53]. The selected control strategy is then enforced on the analogous output vector for creating an objective vector. After iteratively passing through the generations and calculating all the obtained objective vectors, the numeric value of objective vectors obtained in a particular strategy that can be passed and discarded in the subsequent generation is kept in the pass and fail memories, respectively, within the given learning period. The chances of obtaining a unique outcome shall be modified after each iterative generation with respect to the passed and failed ones.

5. FIR Filter Design Implementation Using FLDCSaDE

5.1. Fuzzy Logic-Based Diversity Controlled Self-Adaptive Differential Evolution (FLDCSaDE). This methodology of diversity control is elicited through CO adaptation and executed by means of a fuzzy system using a feedback loop. To bring about obligatory changes in CO, we calculated the mean for the Gaussian or normal distribution of the crossover rate (COm) on the basis of diversity. Implementation of these changes in the algorithm helps enhance the exploration attribute. Diversity in the population is controlled by changing the rate of crossover based on the necessity of the evolutionary process using the fuzzy system. To strike an equality between exploitation and exploration, there exist numerous techniques for obtaining appropriate validity in the evolutionary process and for diversity calculations. In the FLDCSaDE algorithm, for the purpose of obtaining the diversity among the population, we use the "distance-to-average-point" [54] estimate, which is given as follows:

$$\text{div}_g(\text{Pop}) = \frac{1}{|L| * \text{NS}} * \sum_{i=1}^{\text{NS}} \sqrt{\sum_{j=1}^{D} \left(x_{ij} - \overline{x_j}\right)^2}, \tag{21}$$

where the required dimension for the problem is given by D, the size of the population is given by NS for a population Pop, and the diagonal length in the search space defined in the range of $S \subseteq R^D$ is given by $|L|$. The search area is defined by $\sqrt{\sum (x_{\max} - x_{\min})^2}$ where every search variable x lies at bounded limits given by $x_{\min} < x < x_{\max}$; $\overline{x_j}$ denotes j^{th} value for the given average point $\overline{x}$, whereas x_{ij} indicates the j^{th} numeric value for the i^{th} independent individual. The given "distance-to-average" measure needs the size of the population as a constraint, the limits across which the variable used can be searched, and the problem amplitude in terms of its dimensions. Thus, the aforementioned approach is found to be very effective in tackling complicated problems involving dynamic design characteristics [55].

The fuzzy approach used in the system is interpreted by putting together a unique mapping, extending from the specified input onto the output with the help of fuzzy logic [56]. This approach elucidates the numeric value defined for the input vector and with the help of a few laid-down regulations allocates a numeric value for the given output vector. Given COm adaptation in the DCSaDE strategy usually demands two components mentioned as follows:

$$\text{COm}_k = \text{COm}_{s,k} + \text{COm}_{d,k}, \tag{22}$$

where $\text{COm}_{s,k}$ value is acquired by using the SaDE algorithm that usually takes the median defined for every $\text{COm}_{d,k}$ constituent involving diversity, and CO numeric value defined during the past generations of linear programming optimization LP is taken into consideration. Initially, the $\text{COm}_{d,k}$ value was obtained by using a set of if-then statements. If the diversity is above a certain level, then $\text{COm}_{d,k}$ is gradually decreased by a fixed factor; otherwise, it is gradually increased by a fixed factor to improve the overall crossover rate. The problem with this method was determination of the factor by which to increase or decrease the $\text{COm}_{d,k}$ value. Larger values resulted in quicker settling to the maximum or minimum value of COm, which resulted in poor diversity control. Also, fine tuning of the COm value was not possible with the if-then statements.

The fuzzy system-based adaptation causes a gradual change leading to better evolutionary process. The fuzzy system usually takes into consideration the numeric value of COm immediately besides any user mediation based on the problem attributes. The COm value should necessarily be carried out in accordance with the numerous stages in the evolution as well.

During the various stages of evolution, the crossover rate should be suitably modified based on the search space. The fuzzy system helps in the effective tuning process for the numeric value of COm, at the instant of many levels in the evolutionary development. The adoption of COm can be carried out in a similar fashion for a feedback-based fuzzy controller, where the feedback is stated using the diversity deviation involving the current population. Diversity deviation or error is measured by involving deviation of population diversity of the current generation with respect to a reference or desired diversity level. Therefore, for COm_k, variables which are predefined are adapted on the basis of fuzzy logic.

In the given task, a SISO (single input and single output) fuzzy logic-based system developed by Mamdani is used. The input used for the fuzzy system involves deviation in the diversity at g^{th} generation Δerr_g is defined as given deviation in the diversity among the individuals of the population div_g present at g^{th} generation and the diversity taken as a reference (ref).

$$\Delta\text{err}_g = \text{ref} - \text{div}_g. \tag{23}$$

The output obtained FS is interpreted as the deviation of the mean for the given normal distribution at the specified numeric value of the crossover rate (ΔCOm_k^g) under the k^{th} strategy for the g^{th} generation. For the given generation, the COm_k^g is obtained through

$$\text{COm}_k^g = \text{COm}_k^{g-1} + \Delta\text{COm}_k^g. \tag{24}$$

Fuzzy variables for FS are established from output and input. Subfunctions of these variables are classified into Positive (P), Negative (N), Zero (Z), Large Positive (LP), and Large Negative (LN). All of these are specified with the help of functions of triangular membership. For obtaining defuzzification, we use the centroid approach. Towards the completion of every generation, calculation is carried out for obtaining reference diversity (ref) and current population diversity. The deviation among the two terms is sent for the feedback control in the FS loop. The FS makes note of all the important implementation changes in numeric value of COm_k numeric value using the assistance of rule set, obtained from the SaDE approach for revising the crossover rate in the upcoming generation. This combined with success history-based adaptations of CO will be able to adopt the control parameter value depending on the search space of the functions and the stage of evolution during the evolutionary process. The fuzzy approach is summarized in Figure 2. The corresponding false positive rate (FPR) can be computed based on the ripple content which contributes towards false decision outputs. Lower the ripple obtained, the better is the frequency response and in-turn performance thus obtained.

5.2. Implementation of FLSaDE for FIR Filter Design. The general steps involved in the implementation of the abovementioned algorithm are summarized as follows. The objective or fitness function used to minimize the error function is given in equation (4) for both band stop and high pass filters. The abovementioned function is evaluated after each iteration in order to derive optimal coefficients for the given FIR filter. The target error value (tolerance) that should be reached is set to 0.001, and the target objective value is given to be 1.000001. The equality constraint for the given frequency response $h(n)$ is bounded and given by $h(n) = h(N-1-n)$ where $0 \leq n \leq N-1$, where N is the order of the given FIR filter.

The various values of function parameters such as upper and lower bound limits and size of the given population are selected such that they are used in finding and generating optimal coefficients of the given filter for minimizing the error function of high pass and band stop filters.

Step 1: values are initialized to the population size, $N_s = 20$. The lower and upper bound limits for the coefficients of the given filter are assigned −1 and +1, respectively, for both band stop and high pass filters. The number of iterations $N = 20$ for improving the accuracy of the given search space. The values of control parameters are initialized using the formulation rate $M = 0.5$ and the crossover rate $\mathrm{CO} = 0.5$.

Step 2: we evaluate the strategy probability to the 4 strategies listed in the SaDE algorithm. The learning period is set to 50, and the coefficients are divided into lower bound and upper bound values. The fuzzy interference system (FIS) parameters are initialized. In the initialization, the inputs of the system are made fuzzy, followed by the fuzzy operators being applied to obtain the output.

Step 3: the parent population is evaluated. The best value and its id are computed for the given generation. In order to rotate the given population to randomize, the following steps are performed: first, the old population is saved, and then the current index of the pointer array is found. Second, the vector's locations are shuffled, and their indices are rotated.

Step 4: we group the given population into strategies, and their diversity is calculated by finding the error, length of the FIS, and mean CO values. From the mean, the optimal CO and M are computed. For the given bounded coefficients, boundary conditions are checked. After the given parent population is shuffled, the child population is selected from the parent population. If the selected child is better than the previous generations, it is added to the list. For the given population, local search is applied for finding filter coefficients.

Step 5: each function call is performed for each individual iteration, and the parameters are passed in column form. The number of design variables is set to 11. The maximum function evaluation is performed for 200 generations, and the population in each generation is updated based on the worst and best solutions obtained.

Step 6: for each iteration (or run), we find the optimal frequency (fbest) and filter coefficients (xbest). The same procedure is repeated for all the algorithms used in the design of the FIR filter. Finally, the iterations are repeated until the tolerance error and target objective value are reached. After 20 iterations, the one with the minimum fbest value is treated as the best member of the given generation. The values corresponding to the best are the optimal coefficients for the given algorithm.

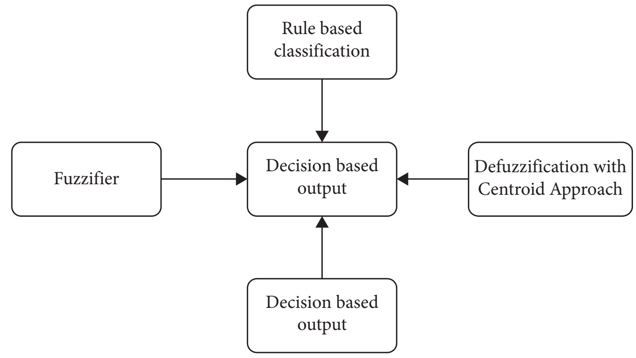

Figure 2: Approach towards the fuzzy algorithm.

6. Simulation Results and Analysis

6.1. High Pass Filter (HPF). A comparison of the high pass filters designed using different evolutionary algorithms (EA) is made with regard to the set of attenuation values and the filter frequency response. The optimum filter coefficients for the algorithms, namely, SaDE, FLDCSaDE, CMAES, Jaya, and DE, are depicted through Table 2.

The ideal filter order is determined by the ability to achieve higher attenuation between stopband and passband, a flatter passband resulting in significantly minimal ripples, and narrow transition band. While lower order filters have minimal roll-off and complexity, they have a larger transition band, leading to irregular filtering action. In comparison to lower order filters, higher order filters also have disadvantages of their own. They have higher complexity which increases in the powers of 2 as we double the order. Hence, there is a need to determine the ideal filter order which encompasses the advantages of both the lower and higher order filters. In this paper, we have utilized the order of the filter with 21 coefficients and 20 iterations. We find that this 21st order filter offers a lower transition band than other higher order filters. Moreover, the filter gain obtained is higher and the error is minimal, making the filter closer to the ideal response [57]. The frequency response for the optimized filter coefficients of each algorithm for both band Stop and high pass filters is obtained through the freqz function in MATLAB which is used to find the frequency response of the given digital filter as shown in Figure 3. We measure the attenuation values for the given filter by taking the absolute values of the given frequency response, converting the values to decibels, and utilizing the peaks obtained in stopband and passband to find the corresponding attenuation [8]. Also, for finding ripples, the absolute values are taken as they are not converted to decibels. The stopband and passband attenuation analogous to the distinct algorithms mentioned above are listed as follows. The abovementioned algorithms form the basis for the algorithms discussed in the paper. The performance of the various algorithms is discussed in the latter sections. The following algorithms are found to be superior among all the other genetic and swarm optimization algorithms due to the fact that they are much more recent, and the limitations of the previous algorithms have been rectified in the following algorithms.

In this paper, a high pass filter is designed to demonstrate the effectiveness of optimization techniques. The specifications for the high pass filter are order of the filter $N = 21$, cut-off frequency $\omega_c = 0.43$, and the number of samples: 1000. The objective function in equation (4) is computed for each and every step until optimal solution is obtained. The limits assigned for the filter coefficients are found to lie within the range of −1 to +1. We analyze the parameters obtained from the frequency response of a high pass FIR filter for their performance, utilizing various evolutionary algorithms (EA), most importantly the stopband ripple, passband, and stopband attenuation. These parameters are tabulated in Table 3.

Based upon the magnitude response in dB obtained from the graphical representation of various algorithms such as SaDE, FLDCSaDE, Jaya, CMAES, and DE in Figure 3, we observe that the plot clearly depicts SaDE and FLDCSaDE having the highest negative magnitude in dB for a given normalized frequency. For more accurate visualization, we use stopband attenuation (A_{stop}) as recorded in Table 4 since it is based on the mean value from each iteration, thus making it a stable value obtained in any iteration rather than a single highest peak, followed by a subsequent lower value. The filters are mainly characterized by their stopband attenuation [58, 59]. From Table 3, SaDE and FLDCSaDE have the highest negative stopband attenuation of −23.9720 dB and −20.2928 dB, respectively, compared to Jaya (−15.8515 dB), CMAES (−14.1167 dB), and DE (−14.4799). However, from their mean value in Table 4, we find FLDCSaDE to outperform other algorithms in the region around the stopband, thus indicating the highest stable attenuation throughout the response. Moreover, we find that the passband attenuation is the lowest for FLDCSaDE (0.4330 dB), indicating that it takes the minimum time for computation. Lower time correlates to a lower number of parameters involved in the filter design. It is seen from the plot that all the algorithms produce almost the same magnitude of overshoot, but among them, FLDCSaDE delivers the minimum overshoot at the point where the ideal filter contains discontinuity [60]. CMAES and Jaya generate the maximum. SaDE as well as DE produces an intermediate overshoot [61, 62]. It shows that FLDCSaDE type filters yield a level and smooth passband response; that is, they have a flat response at the passband. These considerations are studied and passed onto the calculation of the variance, highest, and mean values of the ripples obtained at the passband as in Table 5. Since the variance of the ripples in the passband for all the algorithms is close to 0, it states that the passband ripples of the algorithms for 20 runs are very close to each other. The lower value of stopband ripple from Table 6 for FLDCSaDE indicates that the distortion and aliasing in the waveform are comparatively lower than the other algorithms.

6.2. Band Stop Filter (BSF). In this section, we adopt a similar approach as in the design of a high pass FIR filter. One of the differences in execution is that we have two crossover frequencies for the band stop filter when compared to one for the high pass filter. It is basically the inverse of the band pass filter, and these second order reject filters are designed to provide higher attenuation at and near the single crossover frequency with little attenuation at all other frequencies as shown in Figure 4. For any standard band stop filter, the highest frequency attenuated is about 10 to 100 times more than that obtained from the lowest frequency of attenuation. The conventional band stop filter is designed because of its lower interference and increased performance; also, the amplitude of the reference level of the sideband is comparatively reduced, thus producing a wider stopband. The band stop filter coefficients for various evolutionary algorithms are listed in Table 7.

The design specifications taken into consideration for the states' band stop filter are as follows: order for the given filter $N = 20$, cut-off frequency $\omega_{c1} = 0.38$, $\omega_{c2} = 0.73$, and number of samples: 1000. The objective function in equation (4) has to be estimated after each step in order to come up with the solution which is most effective. The limits assigned for the filter coefficients are found to lie within the range of −1 to +1. Similar to the high pass filter, we analyze the parameters of the frequency response of the band stop FIR filter for finding their performance, utilizing various evolutionary algorithms (EA), most importantly the stopband ripple, passband, and stopband attenuation observed and tabulated in Table 8 [62].

From the normalized magnitude response representation in graphical form in Figure 4, for various algorithms such as SaDE, FLDCSaDE, Jaya, CMAES, and DE, we find that although the maximum filter stopband attenuation from Table 8 is favourable for DE with −26.0818 dB, the passband attenuation is high (1.03995 dB). Taking the mean value in Table 9 into consideration for the stopband attenuation, we find that FLDCSaDE has the highest negative value of −18.86844 dB, as seen from Table 9, which shows that it has the most stable response with the highest magnitude. Moreover, the response of DE is skewed, indicating a sole

Table 2: Optimized high pass filter coefficients of order 21.

Algorithm	Fbest	Xbest
DE	0.0324	−0.01894, 0.03103, 0.0283, 0.00673, −0.03601, −0.0023, 0.07345, 0.14259, −0.1024, −0.3348, 0.5614
CMAES	0.2142	−0.1521, 0.0971, 0.0620, −0.0272, −0.1055, −0.0204, 0.3003, −0.3187, 0.0991, 0.0262−0.0305
Jaya	0.138794	0.0680, −0.0960, −0.0206, 0.0480, −0.0473, 0.1299, −0.0868, −0.2067, 0.3875, −0.1402, −0.0420
SaDE	0.03359	−0.0299, 0.01694, 0.0316, −0.0111, −0.0474, −0.02359, 0.07580, 0.0797, −0.07563, −0.30701, 0.57366
FLDCSaDE	0.0303	0.0252, −0.0149, −0.0389, −0.0005, 0.0516, 0.0281, −0.0619, −0.0833, 0.0686, 0.3104, −0.5709

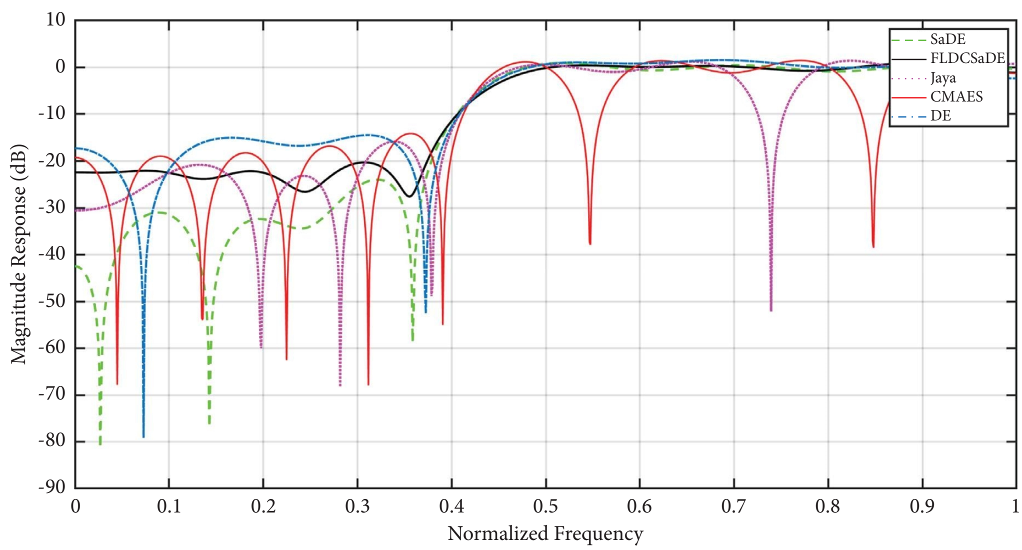

Figure 3: High pass filter normalized frequency response of order 21.

Table 3: Comparative study of the high pass filter after 20th iteration.

Algorithm used	Lowest stopband amplitude (IndB)	Highest passband amplitude (IndB)
DE	−14.4799	1.5653
CMAES	−14.1167	1.4737
Jaya	−15.8515	1.2363
SaDE	−23.9720	0.9697
FLDCSaDE	−20.2928	0.4330

Table 4: Statistical study of the high pass filter after 20th iteration.

Algorithm	Stopband amplitude (in dB)		
	Mean	Variance	Standard deviation
DE	−16.1164	11.8321	3.4398
CMAES	−18.3858	10.4221	3.2283
Jaya	−14.6156	8.8489	2.9747
SaDE	−20.8871	11.4938	3.3902
FLDCSaDE	−22.0034	11.6958	3.4199

Table 5: Quantitative study of the high pass filter after 20th iteration.

Algorithm	Normalized passband ripple		
	Highest	Mean	Variance
DE	1.1975	1.1867	0.0097
CMAES	1.1894	1.2311	0.0011
Jaya	1.1789	1.3140	0.0045
SADE	1.1181	1.1964	0.0037
FLDCSaDE	1.1075	1.2172	0.0034

Table 6: Qualitative study of the high pass filter after 20th iteration.

Algorithm	Normalized stopband ripple	
	Highest	Average
DE	0.1772	0.2582
CMAES	0.1127	0.1768
Jaya	0.0778	0.2391
SaDE	0.0695	0.1078
FLDCSaDE	0.0239	0.1381

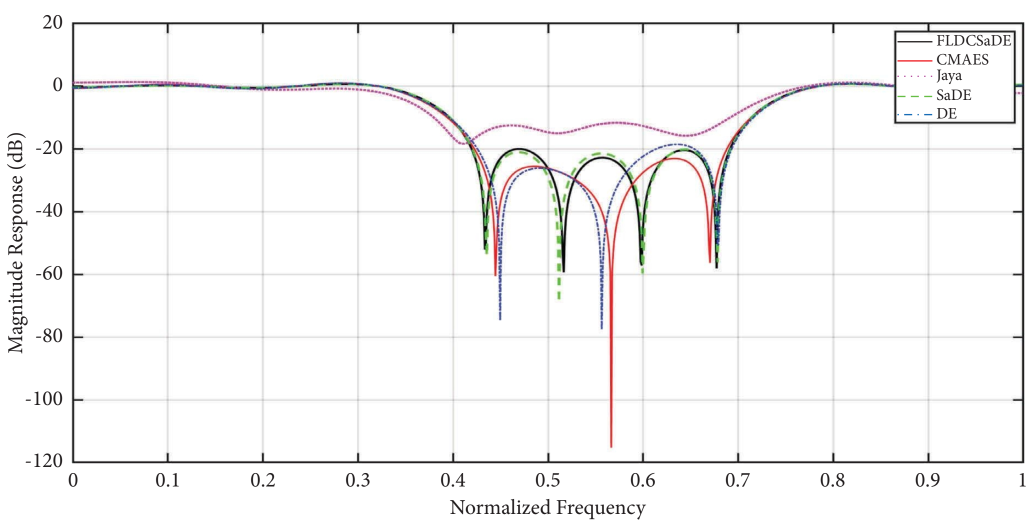

Figure 4: Normalized frequency response for the band stop filter of order.

peak followed by a decaying response. Also, the passband attenuation of the FLDCSaDE is much less than the other algorithms 0.7434 dB, as noticed from Table 10, indicating that the algorithms perform efficiently when applied to either high pass or band stop filters. It is also proposed that the maximum passband ripple for the FLDCSaDE algorithm has a difference of 0.0040 dB with SaDE, 0.6766 dB with Jaya, 0.3026 dB with DE, and 0.2887 dB with CMAES. The mean value of the passband ripple is the lowest for the FLDCSaDE algorithm. Hence, the FLDCSaDE algorithm yields a level and smooth passband response; that is, it has a flat response at the passband. The stopband ripples of all the algorithms in Table 11 except for the Jaya algorithm are very close to each other, and therefore, these algorithms offer high attenuation. Some ripples are observed in the stopband for FLDCSaDE, suggesting that the gain obtained is irregular in the passband region. But the fact that the average value of the ripple in Table 11 is the minimum for the FLDCSaDE algorithm suggests that it has better convergence and a stable response compared to other algorithms.

Table 7: Optimized band stop filter coefficients of order 21.

Algorithm	Fbest	Xbest
DE	0.06214	−0.044382922593895, 0.005565291287194, −0.026315118123262, 0.024595586053397, 0.095609804223588, −0.080856920285804 −0.057625047812984, 0.016096059609101, −0.051098700458515, 0.529308963218684, 0.119807451302831
CMAES	0.06099	−0.045028169563770, 0.000602384998817, −0.023479982438493, 0.025142123236134, 0.089557035881411, −0.077270386217179 −0.059455057949346, 0.015133595780733, −0.048279285580282, 0.534616581064309, 0.108894656326628
Jaya	0.06063	0.001543333372969, −0.059457334194842, 0.003071292790779, 0.067038663689822, −0.010341472683587, 0.032572719095202 −0.099354765943969, −0.089836547432283, 0.261725305613091, 0.050855503693469, 0.652463415791425
SaDE	0.05653	0.006611886394233, −0.069104937084887, 0.016483676921158, 0.056346409141704, −0.009597752715976, 0.036425353480383 −0.097369116871479, −0.104488659309710, 0.264127567920871, 0.055966719269367, 0.648674480040057
FLDCSaDE	0.05640	0.006250468833997, −0.068605876087016, 0.015304200365945, 0.055441396890893, −0.008882275363904, 0.036309763558918 −0.099435949129020, −0.103839433324579, 0.267507239328361, 0.056701712740939, 0.649395723694243

Table 8: Comparative study of the band stop filter after 20th iteration.

Algorithm	Lowest stopband amplitude (dB)	Highest passband amplitude (dB)
DE	−26.0818	1.03995
CMAES	−25.5195	1.0261
Jaya	−12.4252	1.4140
SaDE	−21.4021	0.7434
FLDCSaDE	−22.7314	0.7374

Table 9: Statistical study of the band stop filter after 20th iteration.

Algorithm	Stopband amplitude (in dB)		
	Mean	Variance	Standard deviation
DE	−16.6030	14.8547	3.85418
CMAES	−17.2205	12.0425	3.47022
Jaya	−15.7026	5.56584	2.35920
SaDE	−18.8248	14.9776	3.87008
FLDCSaDE	−18.8684	13.9893	3.74022

Table 10: Quantitative study of the band stop filter after 20th iteration.

Algorithm	Normalized passband ripple		
	Highest	Mean	Variance
DE	1.1272	1.20078	0.007394
CMAES	1.1254	1.21114	0.002519
Jaya	1.1768	1.30965	0.002302
SADE	1.0977	1.18124	0.002534
FLDCSaDE	1.0954	1.19008	0.003816

Table 11: Qualitative study of the band stop filter after 20th iteration.

Algorithm	Normalized stopband ripple	
	Highest	Average
DE	0.0496	0.27779
CMAES	0.0530	0.14961
Jaya	0.2392	0.15719
SaDE	0.0851	0.12599
FLDCSaDE	0.0730	0.12549

Table 12: Symbols and Abbreviations.

S. No.	Symbol	Definition
1	EEG	Electroencephalography
2	MRI	Magnetic resonance imaging
3	ECG	Electrocardiogram
4	RTL	Register transfer language
5	CO	Crossover rate
6	IIR	Infinite impulse response
7	MATLAB	Matrix laboratory
8	PSO	Particle swarm optimization
9	GA	Genetic algorithm
10	FIS	Fuzzy interference system
11	ω_c	Cut-off frequency
12	N	Order of the filter
13	W	Weights
14	D	Dimension
15	N_S	Population size
16	LP	Linear programming optimization

7. Conclusion and Future Scope

Through the medium of this paper, performance for the designed finite impulse response (FIR), high pass (HP), and band stop filter (BS) has been scrutinized. The analysis is compared for various metaheuristic evolutionary optimization algorithms, namely, self-adaptive differential evolution (SaDE), fuzzy logic-based diversity-controlled self-adaptive differential evolution algorithm (FLDCSaDE), covariance matrix adaptation evolution strategy (CMAES), and Jaya algorithm, by taking the differential evolution (DE) algorithm as a reference. The FIR filter design aims at finding optimal coefficients for the given filter such that it minimizes its ripple (both stopband and passband) and hence obtains a smoother response with a minimal relative absolute error, thus attaining a response closer to the ideal response. The observed optimal solutions obtained for the FLDCSaDE algorithm were found to be the best when compared to other algorithms and satisfied the requirements of an efficient FIR filter in terms of performance and reduced ripple. At the same time, the algorithms are found to attain a high attenuation in the stopband, in-turn having a narrower transition band. The superiority of the FLDCSaDE algorithm compared to other algorithms can be attributed to the fact that it offers better diversity because it updates the crossover rate for the next generation. This helps in the retrieval of useful information from the given population and in obtaining a better solution.

The application of the given FLDCSaDE algorithm provides better tuning and performance. The performance in terms of amplitude and ripples for both high pass and band stop filters has been appreciable. The ripples in the order of 0.02 and 0.07 provide a smooth transition. In a practical sense, circuits with lower attenuation in the passband which extends from 100 MHz to infinity will cause the signals to experience an attenuation loss of approximately −10 dB which limits the appreciable bandwidth that can be used in the circuit. The fact that utilizing different filter topologies can cause reduction in ripples may have an added disadvantage of undesired resonances and introduce amplifying elements. The use of FLDCSaDE reduces the ripples to an order of 1, thus scaling the desired output and reducing the group delay. Although the fact that the fuzzy approach can be applied to any randomly generated population, there might be cases where the accuracy can be compromised, especially in the case of multimodal problems. In addition to it, one might need to manually feed the crossover rate. Even in these problems, the accuracy obtained is almost comparable to that obtained by the other algorithms. Moreover, the value of the crossover rate defined will be optimized post each iteration, so the crossover rate fed initially does not result in inaccuracy which makes it efficient. The research can be extended to solve multidimensional and multi-objective problems and can be used to design a corresponding infinite impulse response (IIR) filter. The usage of FIR filter design can be used to improve modulation parameters such as spectral efficiency, clock synchronization, frequency masking, and latency, thus improving bandwidth diversification [63]. The use of digital filters to improve energy efficiency in industrial automation can be considered for future investigation. The use of fuzzy-based optimization can further provide frequency sampling. This approach can be applied to the decision support system which can be utilized to control the differential curve, thereby making it suitable for many nonlinear applications. The usage of adaptive fuzzy systems can further add to the reasoning on the fuzzy patterns, thus enabling task processing and broadening the spectrum of functional intelligence. This provides a practical and an effective solution for various applications such as audio systems for sending and controlling the variable frequency components, biological instruments, and communication systems.

Appendix

A. Symbols and Abbreviations

Symbols and Abbreviations are presented in Table 12.

B. Hardware and Software Specifications

MATLAB 2020a and above Intel Core i3 2 GHz processor with a minimum of 4 GB of RAM must be compatible for the implementation of the given FIR filter design.

Data Availability

The data that support the findings of the study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] S. Kuzhaloli, G. JoselinRetna Kumar, M. S. Godwin Premi, N. Anusha, and M. Rajmohan, "FIR filter design for advanced audio/video processing applications," *AIP Conference Proceedings*, vol. 2523, no. 1, Article ID 20057, 2023.

[2] J. Li, X. Bai, S. Han, and Y. Yu, "The design of FIR filter based on improved DA and implementation to high-speed ground penetrating radar system," in *Proceedings of the 2020 International Wireless Communications and Mobile Computing (IWCMC)*, pp. 1140–1144, Limassol, Cyprus, June 2020.

[3] W. Shi-Wei, "A new denoising method in seismic data processing based on wienerfilter," *Electron Signal Process LNEE*, vol. 97, pp. 191–194, 2011.

[4] S. Janwadkar and R. Dhavse, "Power and area efficient FIR filter architecture in digital encephalography systems," *E-Prime- Advances in Electrical Engineering, Electronics and Energy*, vol. 4, Article ID 100148, 2023.

[5] S.& S. Dogra and N. Sharma, "Comparison of different techniques to design of filter," *International Journal of Computer Application*, vol. 97, no. 1, pp. 25–29, 2014.

[6] A. K. Dwivedi, S. Ghosh, and N. D. Londhe, "Review and analysis of evolutionary optimization-based techniques for FIR filter design," *Circuits, Systems, and Signal Processing*, vol. 37, no. 10, pp. 4409–4430, 2018.

[7] R. V. Ravi, K. Subramaniam, T. V. Roshini, S. P. B. Muthusamy, and G. K. D. PrasannaVenkatesan,

"Optimization algorithms, an effective tool for the design of digital filters; a review," *Journal of Ambient Intelligence and Humanized Computing*, vol. 1–17, 2019.

[8] A.&R. Aggarwal, T. K. Rawat, and D. K. Upadhyay, "Design of optimal digital FIR filters using evolutionary and swarm optimization techniques," *AEU- International Journal of Electronics and Communications*, vol. 70, no. 4, pp. 373–385, 2016.

[9] A. Aggarwal, T. K. Rawat, M. Kumar, and D. Upadhyay, "Optimal design of FIR high pass filter based on L1 error approximation using real coded genetic algorithm," *Engineering Science and Technology, an International Journal*, vol. 18, no. 4, pp. 594–602, 2015.

[10] A. Jiang, H. K. Kwan, Y. Tang, and Y. Zhu, "Sparse FIR filter design via partial 1-norm optimization," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 8, pp. 1482–1486, Aug 2020.

[11] H. Yu, Y. Gao, and J. Wang, "A multi-objective particle swarm optimization algorithm based on competition mechanism and Gaussian variation," *Complexity*, vol. 2020, Article ID 5980504, pp. 1–23, 2020.

[12] S. M. Mirjalili and S. Z. Mirjalili, "Single-objective optimization framework for designing photonic crystal filters," *Neural Computing & Applications*, vol. 28, no. 6, pp. 1463–1469, 2017.

[13] K. Boudjelaba, F. Ros, and D. Chikouche, "Potential of particle swarm optimization and genetic algorithms for FIR filter design," *Circuits, Systems, and Signal Processing*, vol. 33, no. 10, pp. 3195–3222, 2014.

[14] C. X. Wu, X. Xu, X. Zhang, and L. Zhao, "Sparse FIR filter design based on simulated annealing algorithm," *Advances in Electrical and Computer Engineering*, vol. 15, no. 1, pp. 17–22, 2015.

[15] K. Boudjelaba, F. Ros, and D. Chikouche, "An efficient hybrid genetic algorithm to design finite impulse responsefilters," *Expert Systems with Applications*, vol. 41, no. 13, pp. 5917–5937, 2014.

[16] I. Sharma, B. Kuldeep, A. Kumar, and V. K. Singh, "Performance of swarm based optimization techniques for designing digital FIR filter: a comparative study," *Engineering Science and Technology, an International Journal*, vol. 19, no. 3, pp. 1564–1572, 2016.

[17] A. G. Gad, "Particle swarm optimization algorithm and its applications: a systematic review," *Archives of Computational Methods in Engineering*, vol. 29, no. 5, pp. 2531–2561, 2022.

[18] K. Kassoul, N. Zufferey, N. Cheikhrouhou, and S. BrahimBelhaouari, "Exponential particle swarm optimization for global optimization," *IEEE Access*, vol. 10, pp. 78320–78344, 2022.

[19] P. Das, S. K. Naskar, S. Samanta, and S. N. Patra, "An approach to optimize FIR filter coefficients using GA, PSO and BAT algorithm and their comparative analysis," in *Proceedings of the 2016 International Conference on Computer, Electrical & Communication Engineering (ICCECE)*, Kolkata, India, December 2016.

[20] P. Stubberud, "Digital IIR filter design using a differential evolution algorithm with polar coordinates," in *Proceedings of the 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, pp. 1029–1035, Las Vegas, NV, USA, January 2022.

[21] F. Shaikh, I. H. Kalwar, T. D. Memon, and S. Sheikh, "Design and analysis of linear phase FIR filter in FPGA using PSO algorithm," in *Proceedings of the 2017 6th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1–4, Bar, Montenegro, June 2017.

[22] R. Kar, S. Mondal, S. P. Ghoshal, and D. Mandal, "Craziness based Particle Swarm Optimization algorithm for FIR bandstop filter design," *Swarm and Evolutionary Computation*, vol. 7, pp. 58–64, 2012.

[23] S. Mandal, S. P. Ghoshal, R. Kar, and D. Mandal, "Design of optimal linear phase FIR high pass filter using craziness-based particle swarm optimization technique," *Journal of King Saud University- Computer and Information Sciences*, vol. 24, no. 1, pp. 83–92, 2012.

[24] S. K. Saha, S. P. Ghoshal, R. Kar, and D. Mandal, "Cat Swarm Optimization algorithm for optimal linearphase FIR filterdesign," *ISA Transactions*, vol. 52, no. 6, pp. 781–794, 2013.

[25] S. K. Sarangi, R. Panda, P. K. Das, and A. Abraham, "Design of optimal high pass and band stop FIR filters using adaptive Cuckoo search algorithm," *Engineering Applications of Artificial Intelligence*, vol. 70, pp. 67–80, 2018.

[26] S. Yuenyong and A. Nishihara, "A hybrid gradient-based and differential evolution algorithm for infinite impulse response adaptive filtering," *International Journal of Adaptive Control and Signal Processing*, vol. 28, no. 10, pp. 1054–1064, 2013.

[27] B. Karmakar, A. Kumar, R. Mallipeddi, and D. G. Lee, "CMA-ES with exponential based multiplicative covariance matrix adaptation for global optimization," *Swarm and Evolutionary Computation*, vol. 79, no. 2023, Article ID 101296, 2023.

[28] M. Kumar and S. K. Mishra, "Jaya-FLANN based adaptive filter for mixed noise suppression from ultrasound images," *Biomedical Research*, vol. 29, no. 5, 2017.

[29] L.-B. Deng, C.-L. Li, and G.-J. Sun, "An adaptive dimension level adjustment framework fordifferentialevolution," *Knowledge-Based Systems*, vol. 206, Article ID 106388, 2020.

[30] R. Sancibrian, A. Sedano, E. G. Sarabia, and J. M. Blanco, "Hybridizing differential evolution and local search optimization for dimensional synthesis of linkages," *Mechanism and Machine Theory*, vol. 140, pp. 389–412, 2019.

[31] W. Zhong, "Linear phase FIR digital filter design using differential evolution algorithms," Electronic Theses and Dissertations, University of windsor, Windsor, UK, 2017, https://scholar.uwindsor.ca/etd/5959.

[32] J. Dash, B. Dam, and R. Swain, "Design and implementation of sharp edge FIR filters using hybrid differential evolution particle swarm optimization," *AEU- International Journal of Electronics and Communications*, vol. 114, Article ID 153019, 2020.

[33] K. S. Reddy and S. K. Sahoo, "An approach for FIR filter coefficient optimization usingdifferential evolution algorithm," *AEU- International Journal of Electronics and Communications*, vol. 69, no. 1, pp. 101–108, 2015.

[34] X.-G. Zhou, G.-J. Zhang, X.-H. Hao, L. Yu, and D.-W. Xu, "Differential evolution with multi-stage strategies for global optimization," in *Proceedings of the 2016 IEEE Congress on Evolutionary Computation (CEC)*, pp. 2550–2557, Vancouver, BC, UK, July 2016.

[35] S. L. Wang, F. Morsidi, T. F. Ng, H. Budiman, and S. C. Neoh, "Insights into the effects of control parameters and mutation strategy on self-adaptive ensemble-based differential evolution," *Information Sciences*, vol. 514, pp. 203–233, 2020.

[36] J. Li, Y. Gao, K. Wang, and Y. Sun, "A dual opposition-based learning for differential evolution with protective mechanism for engineering optimization problems," *Applied Soft Computing*, vol. 113, Article ID 107942, 2021.

[37] H. Peng, Z. Guo, C. Deng, and Z. Wu, "Enhancing differential evolution with random neighbors-based strategy," *Journal of Computational Science*, vol. 26, pp. 501–511, 2018.

[38] W. Deng, S. Shang, X. Cai, H. Zhao, Y. Song, and J. Xu, "An improved differential evolution algorithm and its application in optimization problem," *Soft Computing*, vol. 25, no. 7, pp. 5277–5298, 2021.

[39] M. Duan, H. Yang, S. Wang, and Y. Liu, "Self-adaptive dual-strategy differential evolution algorithm," *PLoS One*, vol. 14, no. 10, Article ID 222706, 2019 Oct 3.

[40] H. A. Hashim, S. El-Ferik, and M. A. Abido, "A fuzzy logic feedback filter design tuned with PSO for L1 adaptive controller," *Expert Systems with Applications*, vol. 42, no. 23, pp. 9077–9085, 2015.

[41] N. Hansen and A. Ostermeier, "Adapting arbitrary normal distributionsin evolution strategies: the covariance matrix adaptation," in *Proceedingsof the Third IEEE Conference on Evolutionary Computation*, Nagoya, Japan, May 2020.

[42] M. Konieczka, A. Poturała, J. Arabas, and S. Kozdrowski, "A modification of the PBIL algorithm inspired by the CMA-ES algorithm in discrete knapsack problem," *Applied Sciences*, vol. 11, no. 19, p. 9136, 2021.

[43] M. Georgioudakis and V. Plevris, "On the performance of differential evolution variants in constrained structural optimization," *Procedia Manufacturing*, vol. 44, pp. 371–378, 2020.

[44] R. A. Zitar, M. A. Al-Betar, M. A. Awadallah, I. A. Doush, and K. Assaleh, "An intensive and comprehensive overview of JAYA algorithm, its versions and applications," *Archives of Computational Methods in Engineering*, vol. 29, no. 2, pp. 763–792, 2022.

[45] R. Venkata Rao, "Jaya: a simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 7, pp. 19–34, 2016.

[46] M. Aslan, M. Gunduz, and M. S. Kiran, "Jayax:jaya algorithm with xor operator for binary optimization," *Applied Soft Computing*, vol. 82, Article ID 105576, 2019.

[47] P. J. Angeline, *Adaptive and Self-Adaptive Evolutionary Computation, 'inComputational Intelligence: A Dynamic System Perspective*, M. Palaniswami, Y. Attikiouzel, R. J. Marks, D. Fogel, and T. Fukuda, Eds., pp. 152–161, IEEE Press, New York, NY, USA, 1995.

[48] V. R. J. A. Rao and R. Jaya, "A simple and new optimization algorithm for solving constrained and unconstrainedoptimization problems," *International Journal of Industrial Engineering Computations*, vol. 7, no. 1, pp. 19–34, 2016.

[49] R. V. Rao and V. Patel, "An elitist teaching-learning-based optimization algorithm for solving complex constrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 3, no. 4, pp. 535–560, 2012.

[50] R. V. Rao and K. C. More, "Design optimization and analysis of selected thermal devices using self-adaptive jayaalgorithm," *Energy Conversion and Management*, vol. 140, pp. 24–35, 2017.

[51] H. Herrera and M. Lozano, "Fuzzy evolutionary algorithms and genetic fuzzy systems: a positive collaboration between evolutionary algorithms and fuzzy systems," in *Computational Intelligence—Collaboration, Fusion and Emergence*, C. L. Mumfordy and L. Jain, Eds., pp. 83–130, Springer, New York, NY, USA, 2009.

[52] D. Ortiz-Boyer, C. Hervás-Martínez, and N. Garcia-Pedrajas, "CIXL2: a crossover operator for evolutionary algorithms based on population features," *Journal of Artificial Intelligence Research*, vol. 24, pp. 1–48, 2005.

[53] S. Miruna Joe Amali and S. Baskar, "Fuzzy logic-based diversity-controlled self-adaptive-differentialevolution," *Engineering Optimization*, vol. 45, no. 8, Article ID 899915, 2013.

[54] Z. Li, K. Lin, M. Nouioua, S. Jiang, and Y. Gu, "DCDG-EA: dynamic convergence–diversity guided evolutionary algorithm for many-objective optimization," *Expert Systems with Applications*, vol. 118, pp. 35–51, 2019.

[55] M. A. Hannan, J. A. Ali, M. S. HossainLipu et al., "Role of optimization algorithms based fuzzy controller in achieving induction motor performance enhancement," *Nature Communications*, vol. 11, no. 1, p. 3792, 2020.

[56] S. Traferro, F. Capparelli, F. Piazza, and A. Uncini, "Efficient allocation of power-of-two terms in FIR digital filter design using tabu search," in *Proceedings of the 1999 IEEE International Symposium on Circuits and Systems (ISCAS)*, vol. 3, pp. 411–414, Orlando, FL, USA, May 1999.

[57] H. Monsef, M. Naghashzadegan, A. Jamali, and R. Farmani, "Comparison of evolutionary multi objective optimization algorithms in optimum design of water distribution network," *Ain Shams Engineering Journal*, vol. 10, no. 1, pp. 103–111, 2019.

[58] T. K. Rawat, *Digital Signal Processing*, Oxford University Press, Oxford, UK, 1 edition, 2014.

[59] L. Amador-Angulo and O. Castillo, "Statistical analysis of type-1 and interval type-2 fuzzy logic in dynamic parameter adaptation of the BCO," in *Proceedings of the 2015 Conference of the International Fuzzy Systems Association and the European Society for Fuzzy Logic and Technology (IFSA-EUSFLAT-15)*, Gijón, Spain, June 2015.

[60] E. Cuevas, J. Gálvez, and O. Avalos, "Fuzzy logic based optimization algorithm," in *Recent Metaheuristics Algorithms for Parameter Identification. Studies in Computational Intelligence*, vol. 854, Berlin, Germany, Springer, 2020.

[61] S. K. Saha, R. Dutta, R. Choudhury, R. Kar, D. Mandal, and S. P. Ghoshal, "Efficient and accurate optimal linear phase FIR filter design using opposition-based harmony search algorithm," *The ScientificWorldJOURNAL*, vol. 2013, Article ID 320489, 15 pages, 2013.

[62] L. Jiang, H. Zhang, S. Cheng, H. Lv, and P. Li, "An overview of FIR filter design in future multicarrier communication systems," *Electronics*, vol. 9, no. 4, p. 599, 2020.

[63] T. Sinha and J. Bhaumik, "Design of computationally efficient sharp FIR filter utilizing modified multistage FRM technique for wireless communication systems," *Journal of Electronic Science Technology*, vol. 17, pp. 185–192, 2019.