Soft Computing (2020) 24:18113–18128
https://doi.org/10.1007/s00500-020-05065-6

METHODOLOGIES AND APPLICATION

# Design of optimal low-pass filter by a new Levy swallow swarm algorithm

Shubhendu Kumar Sarangi[1] · Rutuparna Panda[2] · Ajith Abraham[3]

Published online: 30 May 2020


**Abstract**

The swallow swarm optimization (SS) is a challenging method of optimization, which has a quicker convergence speed, not getting caught in the local extreme points. However, the SS suffers from a few shortcomings—(1) the movement speed of particles is not controlled suitably during the search due to the requirement of an inertia weight and (2) the less flexibility of variables does not permit to maintain a balance between the local and the global searches. To solve these problems, a new Levy swallow swarm optimization (SSLY) algorithm with the exploitation capability is proposed. This article also provides an optimal design methodology for the low-pass filter using the suggested SSLY technique. A *new objective function* is introduced to achieve the maximally flat frequency response, which is another important contribution to the field. The firefly algorithm (FA), the sine cosine algorithm (SCA) and the standard global optimizers—real coded genetic algorithm (GA), conventional particle swarm optimization (PSO), cuckoo search (CS) and SS, are considered for a comparison. The proposed SSLY outperforms the FA, SCA, GA, PSO, CS and SS algorithms. Results authenticate suitability of the proposed algorithm for solving the filter design problems in the FIR domain.

**Keywords** FIR filter design · Evolutionary techniques · Levy swallow swarm algorithm

## 1 Introduction

Nowadays, digital filters are found to be an integral part of the DSP system. Filters are classified into two classes: FIR and IIR which are depended on the form of filter equations and the structure of the implementation (Parks and Burrus 1987; Parks and McClellan 1972). The window and frequency sampling methods are the most popular methods of the FIR filter design. The amount of allowable ripples in the pass-band, stop-band, stop-band attenuation and transition width decides different design procedures. FIR filters offer better stability and linear phase under certain conditions. Realization of the FIR filters using a recursive formula is also possible. Implementation of a finite word length digital system is free from oscillations. It has greater flexibility to control the nature of its magnitude response (Karaboga and Cetinkaya 2011).

In the past, authors have proposed several methods for the design of digital filters. Since the FIR filter offers several advantages as compared to the IIR filter, researchers consider the design of non-recursive filters as a recent demanding optimization problem. Although traditional techniques like steepest descent, quasi-Newton and gradient-based algorithms are used in an optimal design, these methods are generally helpful for solving unimodal problems of optimization. The optimization methods based on the classical gradient techniques are not suitable for optimizing the FIR filter, as they are highly susceptible to the initial points, when the variable numbers along with the size of the solution space are more. They also recurrently approach to the local best possible solution, otherwise

Communicated by V. Loia.

✉ Rutuparna Panda
r_ppanda@yahoo.co.in

Shubhendu Kumar Sarangi
shubhendu1977@gmail.com

Ajith Abraham
ajith.abraham@ieee.org

1 Department of Electronics and Instrumentation Engineering, I.T.E.R, SOA Deemed to be University, Bhubaneswar, India

2 Department of Electronics and Telecommunication Engineering, Veer Surendra Sai University of Technology, Burla, India

3 MIR Labs, Washington, USA

diverge. Another disadvantage is that the cost function must be continuously differentiable. So, most useful evolutionary methods are to be implemented in digital filter design for enhanced control of parameters and superior approximation to an ideal filter.

The FIR filter design technique is based on the multi-modal minimization of the MSE (mean square error). Different evolutionary algorithms are used by various researchers in designing the FIR filters. The objective function used for the optimal filter design precisely controls the different parameters. This creates a nature of vastly non-uniformity, nonlinearity, non-differentiability and multi-modality. Such objective function optimization tasks cannot be done by using the classical methods and cannot touch the global minimum solution. Therefore, to avoid the drawbacks of optimization by using the classical methods, utilization of numerous optimization algorithms having heuristic and meta-heuristic nature is done by several researchers (Sarangi et al. 2014). The majority of these algorithms depend on evolutionary techniques.

Different EAs have been deployed in the design of the FIR filters competently by fulfilling the design requirements, which would otherwise unachievable. These include simulated annealing (SA), GA (Mastorakis et al. 2003; Ahmad and Andreas 2006), differential evolution (DE), bee colony optimization (Karaboga 2009), collective animal behavior (CAB), etc. The GA appears as a promising candidate while considering the global optimization techniques used for the design of a digital filter. It has the ability of getting nearer global finest solution in the filter design. RGA has a fine performance for locating hopeful areas of the search space, but they are incompetent in finding the global optimum while considering the speed of convergence. To avoid problems related to GA (Ahmad and Antoniou 2006), the orthogonal genetic algorithm has been proposed. Adaptive differential evolution, differential cultural algorithm, PSO, quantum PSO (Fang et al. 2006), adaptive inertia weight PSO, craziness PSO (Mandal et al. 2011, 2012), gravitation search algorithm (Rashedi et al. 2011), seeker optimization, cat swarm optimization (Saha et al. 2013), DE-PSO (Karaboga and Cetinkaya 2006) etc., are also used for the filter design. Mandal et al. (2012) presented the filter design using the craziness PSO. Aggarwal et al. (2016) discussed the design of the optimal FIR filters based on the evolutionary methods.

In Mukherjee et al. (2017), the authors suggested the use of the whale optimization algorithm for the design of type I and type II filters having orders 20 and 30, respectively. Similarly, the author Ji (2016) uses the ABC algorithm. In Zhang and Kwan (2017), multi-objective teaching–learning is deployed in designing the filters. The paper (Liang and Kwan 2017) discusses the design of filters using the multi-objective cuckoo search. The researcher in the paper (Chen et al. 2018) designs the low-pass filter using the multi-objective optimization algorithm. In Rana et al. (2016), the authors utilize the constrained genetic algorithm for designing the filters. The author in San-José-Revuelta (2018) suggested the use of a memetic algorithm for the filter design. Similarly, Dhabal et al. (2016) presented the design of filters using the cuckoo search. Kwan (2017) used the interactive self-learning algorithm for designing the low-pass filter. Recently, an adaptive cuckoo search technique has been used to obtain optimal coefficients for the design of the high-pass and the band-stop FIR filters are presented in (Sarangi et al. 2018). In Dwivedi et al. (2018), a comprehensive review of the design of the FIR filters using evolutionary methods is presented. The use of the shuffle frog-leaping algorithm for the design of the FIR filters is discussed in Jiménez-Galindo et al. (2019). A hybrid ESA-DE method for the design of the FIR filters is found in Deng et al. (2019). The authors Ravi et al. (2019) presented a review on the design of the FIR filters using the evolutionary methods. Dash et al. (2020) discussed the use of DE-PSO algorithm for the FIR filter design. However, these authors never used the mean *rms* error frequency while proposing the objective functions, limiting the scope of achieving the maximally flat frequency response.

The majority of the above-mentioned algorithms illustrate the troubles of setting up control parameters of an algorithm, pre-convergence, saturation and recomputing equal result at frequent intervals. For practical realization, the optimal design parameters of the digital filter are very much needed. However, the majority of the existing evolutionary optimization techniques provide us the suboptimal design parameters only. These are the reasons why we are motivated to innovate new swarm intelligence-based meta-heuristic algorithm having (1) exploration; (2) exploitation; and (3) adaptation capabilities to handle the filter design problems efficiently, which is a worthwhile subject of study. Recently, Neshat et al. (2012) proposed a swarm-based optimization technique coined as the swallow swarm optimization (SS). Swallow birds possess high intelligence and fly collectively with a very high speed. These birds produce different sounds in different situations for a strong interaction between them. They eat insects, which are collected by their wings on flying. Sometimes, they also forage for the prey off branches and on the ground. Interestingly, the SS possesses the high exploration capability. On the other hand, cuckoo birds (Yang 2009; Yang and Deb 2010) possess high exploitation ability. It is remarkable to mention here that the exploration and the exploitation are abilities to explore the total search space and assemble to an improved result, respectively. Integrating these two above features, most excellent solutions can be acquired with a few function evaluations. This paper made an attempt to supplement the exploitation feature of the CS in SS in order to enhance the exploitation ability. In addition, the Levy flight is integrated in the proposed algorithm, which assists us to reach the near (global) optimum speedily. The aim of this idea is to acquire the faster and a stable solution. So, the proposed algorithm SSLY lands up with a quicker convergence than the CS and the SS algorithms. So, optimal design parameters are found with an appreciably less number of function evaluations. In addition, the newly proposed objective function based on the mean *rms* error frequency is more effective than the existing objective functions used for the design of the FIR filters by the above authors.

The organization of the paper is as follows: Sect. 1 is the introduction. Section 2 describes the FIR filter design methodology. A new objective function is also introduced in this section. Section 3 explains the algorithm implementation steps. Section 4 presents the simulation results and discussions. Conclusions are made in Sect. 5.

## 2 FIR filter design methodology

The unit sample response of the FIR filter is

$$H(z) = \sum_{n=0}^{N} h(n) z^{-N} \quad n = 0, 1 \ldots N \tag{1}$$

where *h(n)* is the unit sample response.

The difference equation can be represented by

$$y(n) = h(0)x(n) + h(1)x(n-1) + \cdots + h(N)x(x-N) \tag{2}$$

The length of $h(n)$ is $N + 1$ with the similar number of coefficients and $N$ is recognized as the order of the filter. Here, the optimal values of $h(n)$ are obtained using the optimization techniques. In this paper, the GA, PSO, CS, SS, FA, SCA and SSLY algorithms are used for the optimal design of the low-pass filter. This work highlights the design of an even order FIR filter. Note that here we assume the positive even symmetry for $h(n)$. The dimension of the problem on hand is reduced by a factor of 2 by updating only half of the coefficients. The low-pass optimal FIR filters are designed in this paper using GA, PSO, CS, SS, FA, SCA and SSLY individually. Each optimization algorithm attains the minimum error between the desired frequency response and the actual frequency response by deciding the optimal values of $h(n)$ after a certain maximum number of iterations. The desired filter has a magnitude one on the pass band and a magnitude zero on the stop band. The individuals with lower fitness values represent the better the filter. The various filter parameters such as the ripples in the pass band as well as in the stop band and stop-band attenuation along with the transition width are acting as the desirable constraints for a practical filter design.

The well-known frequency response of the low-pass FIR filter (to be designed) is,

$$H(w_k) = \sum_{n=0}^{N} h(n) e^{-jw_k n} \tag{3}$$

where $w_k = (2\pi k/N)$

The objective function (first hand) proposed here for the filter design is basically the error function. It is interesting to mention here that the error in the pass band of the filter frequency response is defined as,

$$E_p^2(\omega) = \frac{1}{\omega_1} \int_0^{\omega_1} \left[1 - \hat{H}(\omega)\right]^2 W_A(\omega) \mathrm{d}\omega. \tag{4}$$

where $\omega_1$ signifies the buildup interval. Usually, the buildup interval is reflected as the pass-band frequency $\omega_p$. $W_A(\omega)$ represents the weighting function. In this work, we consider the unity value for $W_A(\omega)$. Note that $\hat{H}(\omega)$ represents the desired maximally flat frequency response.

Here, the measure for the pass-band error $\omega_e$ is given as,

$$\omega_e = \frac{\int_0^{\omega_1} \omega E_p(\omega) \mathrm{d}\omega}{\int_0^{\omega_1} E_p(\omega) \mathrm{d}\omega}, \quad \text{for } 0 < \omega_1 \le \omega_p \tag{5}$$

Note that $\omega_e$ is called the mean rms error frequency; $\omega_p$ provide us the centered error frequency within the pass-band frequencies $0 < \omega_1 \le \omega_p$.

To make the ratio of the pass-band to the stop-band ripple more adjustable, we propose an effective objective function in this paper.

The objective function $J$ is,

$$J = \max_{\omega \le \omega_p} (\omega_e - \delta_p) + \max_{\omega \ge \omega_s} (\omega_e - \delta_s) \tag{6}$$

Note that $\delta_p$, $\delta_s$ represent ripples in pass band and stop band, respectively. Here, $\omega_p$, $\omega_s$ denote pass-band and stop-band normalized edge frequencies, respectively.

The objective function $J$ described by Eq. (6) is the proposed error fitness function. This function is minimized by the proposed SSLY method.

## 3 Evolutionary computing techniques

In this section, six popular evolutionary computing techniques, i.e., GA, PSO, CS, FA, SCA and SS, are discussed. Further, the newly proposed Levy SS (SSLY) algorithm is also presented.

### 3.1 Genetic algorithm (GA)

Based on Darwin's theory, GA is categorized as an optimization technique depending on the philosophy of genetics. It allows a population to have the individuals under the specified selection rules to get an optimum of the fitness values (i.e., minimizes/maximizes the cost function). There are four main characteristics found in the GA, which shows the differences from the traditional optimization algorithms. The natural selection principle was explained by Charles Darwin. The genetic algorithm is the most popular algorithm; it has been used for the decades for the experimental purposes, getting the optimal solution in the filter design applications.

GA is the stochastic search method that generally used for finding an optimal solution in searching the evolutionary function of an optimization problem. GA is a little bit different from the classical optimization; and there are several aspects of searching methods rather than focusing on a single solution. It operates on a trial group of solutions where they can operate a population of individuals in each iteration. Each individual is indicated as the chromosome representing as one solution to the problem. GA can be applied to the solution for designing the problems provided by a number of trial solutions, which can be coded in the form of data structures such as a string (Ababneh and Bataineh 2008). These evaluations of trial solutions can have a relative basis of the possibility for getting the solution. A comparison is done on the average fitness of the whole population with the fitness function, which is assigned to each trial solution, to provide a comparative fitness value. Basically initialization, selection, mutation and crossover are the four operators used in the GA.

### 3.2 Particle swarm optimization (PSO)

The PSO (Kennedy and Eberhart 1995) is a meta-heuristic optimization algorithm, which depends on the population. In the PSO, particles are initialized, which are updated again and again to search for the optimum solution. The particle velocity changes dynamically (within the search space).

$$V_i^{(k+1)} = w * V_i^k + C_1 * \mathrm{rand}_1 * \left(\mathrm{pbest}_i^k - S_i^k\right) + C_2 * \mathrm{rand}_2 * \left(\mathrm{gbest}^k - S_i^k\right) \tag{7}$$

$$S_i^{k+1} = S_i^k + V_i^{k+1} \tag{8}$$

where $V_i^{k+1}$ is the updated velocity, $V_i^k$ is the velocity, $w$ is an inertia factor, $S_i^k$ is the position of a particle '$i$' in the $k$th iteration, $C_1$ and $C_2$ are acceleration constants. $\mathrm{pbest}_i^k$ is the previous best position of the $i$th particle in the $k$th iteration, $\mathrm{gbest}^k$ is the global best position of the entire population, $\mathrm{rand}_1, \mathrm{rand}_2$ are two random numbers generated in the range [0,1].

### 3.3 Cuckoo search (CS)

The cuckoo search is an algorithm of optimization proposed by Yang and Deb (2009, 2010. It is encouraged by reproduction of several cuckoo groups, i.e., lay eggs in other host bird's nest. Sometimes, the host bird may presume foreign eggs as its own and takes care. The hatching probability of the cuckoo's egg depends on a host bird's cleverness. But, if a host bird locates foreign eggs, it will either throw or simply abandon its nest (Panda et al. 2013). The cuckoo search algorithm is presented below.

$$\mathrm{X}_i^{t+1} = \mathrm{X}_i^t + \alpha \oplus \mathrm{Levy}(\lambda), \tag{9}$$

where $\alpha > 0$ is the step size, the symbol $\oplus$ represents entrywise multiplication. $\mathrm{Levy}(\lambda)$ is taken from Levy flight distribution (Yang and Deb 2009, 2010). More details on Eq. (9) are found in Yang and Deb (2009, 2010).

### 3.4 Firefly algorithm (FA)

The firefly algorithm provides a concrete platform in the computational intelligence (Yang 2009). The mathematical model for the movement of fireflies is done using in the optimization of several parameters in the world of engineering. The differences in the intensity of light along with the attractiveness between the fireflies are two of the important parameters controlling the algorithm. The brightness of fireflies is symbolized as $Gw(x)$ which is defined at a particular $x$ with the intensity of light as the reference as $Gw_0$

$$Gw = Gw_0 e^{-\gamma r^2} \tag{10}$$

The relationship for the attractiveness of firefly having the neighboring fireflies with $\beta_0$ is attractive parameter at $r = 0$ can be referred as

$$\beta = \beta_0 e^{-\gamma r^2} \tag{11}$$

In between any two fireflies, the relative distance can be expressed as

$$r_{ij} = \left\| x_i - x_j \right\| = \sqrt{\sum_{k=1}^{D} \left(x_{i,k} - x_{j,k}\right)^2} \tag{12}$$

The above equation refers to the distance between the two relative positions of the fireflies. Depending on this, the movement of one firefly to another is given by

$$Ps_i = Ps_i + \beta_0 e^{-\gamma r_{ij}^2}\left(Ps_j - Ps_i\right) + \alpha\left(\mathrm{rand} - \frac{1}{2}\right) \tag{13}$$

where the value of $\beta$ is positive for proper movement of fireflies. The parameter $\alpha$ is adoptable in nature whose values are taken arbitrarily in the process of program execution within the range 0 to 1 for a variety of applications. The value of $\beta_0$ is also 1 for most of the applications.

### 3.5 Sine cosine algorithm (SCA)

The SCA is recognized as a multi-agent-based optimization algorithm using the concept of trigonometric sine and cosine functions. This algorithm creates multiple preliminary random solutions and uses the concept of mathematics to move toward the best possible solution. The initial populations of random variables positioned in the solution space are evaluated each time by utilizing the objective function of the design problem. By the use of mathematical expressions specified below, the positions of each variable are updated to produce the final best value after the end of the simulation experiment (Mirjalili 2016)

$$X_i^{t+1} = X_i^t + r_1 \times \sin(r_2) \times \left| r_3 P_i^t - X_i^t \right| \quad \text{if } r_4 < 0.5 \tag{14}$$

$$X_i^{t+1} = X_i^t + r_1 \times \cos(r_2) \times \left| r_3 P_i^t - X_i^t \right| \quad \text{if } r_4 \geq 0.5 \tag{15}$$

The value of $r_1$ can be selected by using the equation

$$r_1 = a - t\frac{a}{T}$$

Note that $t$ represents the current iteration; $T$ denotes the max no. of iterations. Here, $a$ represents a constant decreasing linearly from $a$ to 0 (usually $a$ = 2) over the iterations.

### 3.6 Swallow swarm optimization (SS)

The SS is (swarm intelligence) developed by Neshat et al. (2012). It uses the behavior of the swallow swarm birds. The swallows are characterized by an immigration, high speed flying, skilled hunting, power for communication and interest in the social life. They are also leaders, which lead the small groups in the search of food. The leadership ability develops the concept of the local leaders and the head leaders. The algorithm potential lies with the concept of the exploration of a large search space. The SS and PSO have some similar features, but they also have several noteworthy differences. Progressive optimization is done of the randomly generated initial population of particles. Three categories of the particles take into consideration while executing the searching operation. The leader particles are divided into two categories: local leaders (LL), so as to deal with the connected internal sub-colonies, illustrate a local optimum point; and head leader (HL) is accountable for the leadership of the whole colony and it represents a global optimum point. Here, the explorer particles constitute a bigger part of the population. These pay attention toward the exploration of the design space which might have missed by the groups. The aimless particles move randomly for a better solution. In every iteration ($k$), they take part in the special roles. The equations governing the movement and the exploring procedure are as specified below.

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{16}$$

$$V_i^{k+1} = V_{\mathrm{HL}i}^{k+1} + V_{\mathrm{LL}i}^{k+1} \tag{17}$$

$$V_{\mathrm{HL}i}^{k+1} = V_{\mathrm{HL}i}^{k} + \alpha_{\mathrm{HL}}\mathrm{rand}()(X\mathrm{best}_i^k - X_i^k) + \beta_{\mathrm{HL}}\mathrm{rand}()(\mathrm{HL}_i^k - X_i^k) \tag{18}$$

$$V_{\mathrm{LL}i}^{k+1} = V_{\mathrm{LL}i}^{k} + \alpha_{\mathrm{LL}}\mathrm{rand}()(X\mathrm{best}_i^k - X_i^k) + \beta_{\mathrm{LL}}\mathrm{rand}()(\mathrm{LL}_i^k - X_i^k) \tag{19}$$

$$O_{i+1} = O_i + \left[\mathrm{rand}(\{-1,1\}) * \frac{\mathrm{rand}(\min_s, \max_s)}{1 + \mathrm{rand}()}\right] \tag{20}$$

$X_i^{k+1}$ is the new position of an exploring particle, $X_i^k$ is the position of an exploring particle, $V_i^{k+1}$ is the changed new velocity of an exploring particle, $V_{\mathrm{HL}i}^k$ is the velocity of the head leader, $V_{\mathrm{LL}i}^{k+1}$ is the velocity of the local leader, $X\mathrm{best}_i^k$ is the best position of an exploring particle, $\mathrm{HL}_i^k$ is the position of the head leader, $\mathrm{LL}_i^k$ is the position of the local leader, $\alpha_{\mathrm{HL}}$ denotes the acceleration coefficient of the head leader, $\beta_{\mathrm{HL}}$ represents the acceleration coefficient of the head leader,$\alpha_{\mathrm{LL}}$ is the acceleration coefficient of the local leader, $\beta_{\mathrm{LL}}$ represents the acceleration coefficient of the local leader and $O_{i+1}$ is the next position of the random particle. The various parameters $\alpha_{\mathrm{HL}}$, $\beta_{\mathrm{HL}}$, $\alpha_{\mathrm{LL}}$, $\beta_{\mathrm{LL}}$ can be calculated as discussed in the referred article (Neshat Mandal et al. 2012).

### 3.7 The proposed Levy swallow swarm algorithm (SSLY)

The proposed algorithm provides remarkable better solutions. Swallow swarm algorithm is popular for its effortlessness and its exploitation capability to search for the global or near-global solution. In addition, the SS algorithm provides an improved local search method with good initial estimates to resolve the filter design problems. The algorithm potential also lies within the concept of exploitation, to avoid the local minima, for getting a global or near-global solution. Here, we exploit more numbers of younger swallow birds to search better food randomly, so that they never stuck off with the local minima. We call them exploited birds. Sometimes, they also provide us useful global information. The product $\oplus$ indicates the entrywise multiplication. In this sense, an enhanced exploitation feature is incorporated in our proposed SSLY algorithm, which is described below. Equations (21) to (24) are same as the normal swallow swarm mathematical expressions, which produce the improved solutions. These improved solutions are further refined by the use of Eq. (25) with the concept of Levy distribution function. After the refinement, the value is the fitness of the global best and is compared with the fitness of the aimless particle for further improvement, if possible.

$$V_{\mathrm{HL}i}^{k+1} = V_{\mathrm{HL}i}^{k} + \alpha_{\mathrm{HL}}\mathrm{rand}()(X\mathrm{best}_i^k - X_i^k) + \beta_{\mathrm{HL}}\mathrm{rand}()(\mathrm{HL}_i^k - X_i^k) \tag{21}$$

$$V_{\mathrm{LL}i}^{k+1} = V_{\mathrm{LL}i}^{k} + \alpha_{\mathrm{LL}}\mathrm{rand}()(X\mathrm{best}_i^k - X_i^k) + \beta_{\mathrm{LL}}\mathrm{rand}()(\mathrm{LL}_i^k - X_i^k) \tag{22}$$

$$V_i^{k+1} = V_{\mathrm{HL}i}^{k+1} + V_{\mathrm{LL}i}^{k+1} \tag{23}$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \tag{24}$$

$$\mathrm{X}_i^{k+1} = \mathrm{X}_i^{k+1} + \alpha \oplus \mathrm{Levy}(\lambda) \tag{25}$$

$$\mathrm{Levy}(\lambda) = \frac{u}{[v]^{\frac{1}{(\lambda-1)}}}$$

$$O_{i+1} = O_i + \left[\mathrm{rand}(\{-1,1\}) * \frac{\mathrm{rand}(\min_s, \max_s)}{1+\mathrm{rand}()}\right] \tag{26}$$

where $\alpha$ is the step size newly introduced in this algorithm, $\mathrm{Levy}(\lambda)$ is taken from the Levy distribution (Mantegna 1994). In this connection, Mantegna's algorithm may be used. It is computed as (Mantegna 1994). The step size $\alpha$ is chosen for a better exploitation of the local searched space. In all these equations, the coefficients are calculated using the mathematical formulations provided here. This method works as it uses the concept of the Levy distribution to further refine the search space. Equation (24) is the update equation for the normal SSO. The addition of the Levy term as in Eq. (25) helps us in updating the position faster by exploiting the optimum solution from the solution space. The characteristics of the Levy flight make the step size adaptive resulting in a quicker selection of the optimum solution (Fig. 1).

Steps of the Levy swallow swarm optimization algorithm

*Step 1* Initialize randomly the positions along with velocities of the entire particles.

*Step 2* Calculate the objective fitness function values for all the particles.

*Step 3* Assign HL and LLs from the initial population.

*Step 4* Update location and velocity of every exploring particle.

*Step 5* Find the Local leader in different groups by considering the fitness values.

*Step 6* Form all local leaders and find a head leader.

*Step 7* Apply the Levy concept to further update the position and velocity of the head leader; and select the best after comparison with the past head leader value.

*Step 8* Update the velocity of the exploring particle using the updated parameter of the head leader in the next iteration.

*Step 9* Compare the refined head leader value with the random aimless particle for further improvement; if possible.

*Step 10* Verify the termination condition.

*Step 11* If the criterion is not fulfilled, then repeat the step number 2 to 10.

*Step 12* After the criterion is fulfilled, get the optimum values.

This newly introduced algorithm is used for computing the optimal filter coefficients. It is reiterated that the objective function introduced in Eq. (6) is used to achieve the best results.

## 4 Results and discussions

This section presents non-recursive FIR filter design using GA, PSO, CS, FA, SCA, SS and SSLY. Extensive simulation using the MATLAB (Hussain et al. 2011) is

**Table 1** Control parameters for GA, PSO, SS, CS, FA, SCA and SSLY

| Algorithm parameters and standard values | |
|---|---|
| GA | PSO |
| Population size 100 | Population size 100 |
| Crossover rate 0.8 | Cognitive parameter ($C_1$)-2.05 |
| Crossover two-point | Cognitive parameter ($C_2$)-2.05 |
| Mutation rate 0.001 | Inertia weight ($w$)-0.9 |
| Selection probability 1/3 | |
| CS | FA |
| Population size 100 | Population size 100 |
| Discovery rate (Pa) 0.25 | $\alpha - 0.2$, $\beta - 1$, $\Upsilon - 1$ |
| SCA | SS |
| Population size 100 | Population size 100 |
| $r1$, $r2$, $r3$, $r4$ | $\alpha_{\mathrm{HL}}$, $\beta_{\mathrm{HL}}$, $\alpha_{\mathrm{LL}}$,$\beta_{\mathrm{LL}}$- |
| Referred in Mirjalili (2016) | Neshat et al. (2012) |
| SSLY | |
| Population size 100 | |
| $\alpha_{\mathrm{HL}}$, $\beta_{\mathrm{HL}}$, $\alpha_{\mathrm{LL}}$,$\beta_{\mathrm{LL}}$ | |
| Neshat et al. (2012) | |
| Mantegna (1994) | |

**Table 2** Optimized coefficients for 20 order FIR low-pass filter

| hh($n$) | GA | PSO | CS | SS | SSLY | FA | SCA |
|---|---|---|---|---|---|---|---|
| hh(1) = hh(21) | 00.02371 | 00.02054 | 00.02031 | 00.01920 | 00.01822 | 00.02021 | 00.01979 |
| hh(2) = hh(20) | − 00.04570 | − 00.04772 | − 00.04760 | − 00.03931 | − 00.02930 | − 00.04758 | − 00.04648 |
| hh (3) = hh(19) | − 00.03592 | − 00.03460 | − 00.03401 | − 00.03371 | − 00.03250 | − 00.03398 | − 00.03370 |
| hh(4) = hh(18) | 00.00951 | 00.00901 | 00.00850 | 00.00820 | 00.00721 | 00.00852 | 00.00902 |
| hh(5) = hh(17) | 00.07221 | 00.06860 | 00.06801 | 00.05501 | 00.04501 | 00.06794 | 00.05997 |
| hh(6) = hh(16) | 00.02430 | 00.02358 | 00.02358 | 00.02251 | 00.02150 | 00.02356 | 00.01953 |
| hh(7) = hh(15) | − 00.08732 | − 00.07714 | − 00.07710 | − 00.07230 | − 00.07131 | − 00.07714 | − 00.08012 |
| hh(8) = hh(14) | − 00.07731 | − 00.07911 | − 00.07909 | − 00.06751 | − 00.06650 | − 00.07619 | − 00.08116 |
| hh(9) = hh(13) | 00.06362 | 00.06244 | 00.06144 | 00.05840 | 00.05801 | 00.06082 | 00.06187 |
| hh(10) = hh(12) | 00.28941 | 00.28901 | 00.27900 | 00.29070 | 00.29001 | 00.27805 | 00.28900 |
| hh(11) | 00.42001 | 00.42001 | 00.42001 | 00.42001 | 00.42001 | 00.42001 | 00.42001 |

**Table 3** Optimized coefficients for 30 order FIR low-pass filter

| hh($n$) | GA | PSO | CS | SS | SSLY | FA | SCA |
|---|---|---|---|---|---|---|---|
| hh(1) = hh(31) | 00.02692 | 00.02178 | 00.02270 | 00.02268 | 00.01586 | 00.02598 | 00.02076 |
| hh(2) = hh(30) | − 00.00499 | − 00.00557 | − 00.00599 | − 00.00598 | − 00.00580 | − 00.00489 | − 00.00566 |
| hh (3) = hh(29) | − 00.02257 | − 00.02252 | − 00.02350 | − 00.02342 | − 00.02502 | − 00.02253 | − 00.02247 |
| hh(4) = hh(28) | − 00.00827 | − 00.00842 | − 00.00856 | − 00.00855 | − 00.00872 | − 00.00825 | − 00.00838 |
| hh(5) = hh(27) | 00.01874 | 00.01562 | 00.01673 | 00.01672 | 00.01411 | 00.01871 | 00.01545 |
| hh(6) = hh(26) | 00.02084 | 00.02088 | 00.02099 | 00.02098 | 00.02195 | 00.02085 | 00.02091 |
| hh(7) = hh(25) | − 00.00918 | − 00.00943 | − 00.00924 | − 00.00923 | − 00.00939 | − 00.00916 | − 00.00939 |
| hh(8) = hh(24) | − 00.03069 | − 00.03077 | − 00.03089 | − 00.03088 | − 00.03197 | − 00.03064 | − 00.03071 |
| hh(9) = hh(23) | − 00.00154 | − 00.00145 | − 00.00144 | − 00.00144 | − 00.00133 | − 00.00146 | − 00.00143 |
| hh(10) = hh(22) | 00.04736 | 00.04524 | 00.04720 | 00.04714 | 00.04703 | 00.04726 | 00.04562 |
| hh(11) = hh(21) | 00.03164 | 00.03023 | 00.03154 | 00.03153 | 00.03139 | 00.03162 | 00.03048 |
| hh(12) = hh(20) | − 00.05899 | − 00.05778 | − 00.05889 | − 00.05888 | − 00.05890 | − 00.05887 | − 00.05674 |
| hh(13) = hh(19) | − 00.08514 | − 00.08248 | − 00.08460 | − 00.08458 | − 00.08405 | − 00.08514 | − 00.08241 |
| hh(14) = hh(18) | 00.06995 | 00.06652 | 00.06993 | 00.06992 | 00.06991 | 00.06896 | 00.06586 |
| hh(15) = hh(17) | 00.31240 | 00.30150 | 00.30100 | 00.30130 | 00.29680 | 00.30214 | 00.30138 |
| hh(16) | 00.40510 | 00.40510 | 00.40510 | 00.40510 | 00.40510 | 00.40510 | 00.40510 |

performed to design the FIR LP filters (order 20 & 30). Hence, filter coefficients lengths are taken as 21 & 31, respectively. Equation (6) is used here as the objective function. Optimized coefficients are obtained by deploying seven optimization techniques. For getting the best results, we consider 50 runs. All codes are implemented in the MATLAB on a core i3 processor, 2.20 GHz with 4 GB RAM.

The design of FIR filter is a highly computational task, and it involves a lot of parameters to be optimized, to get the best solution. A perfect design process involves several parameters to handle (as operational constraints). The design constraints include the ripples in the pass band together with the stop band, the pass-band attenuation, the stop-band attenuation and the transition time. The computational cost of hardware, including memory as well as the computation time, is the most necessary ingredients of the design process. When the order of the filter increases, there is an increase in the computational complexity. Hence, the computation time also increases. To meet the design challenges, the hardware that includes a faster processor and more memory to handle the faster processing with a lesser amount of time is of prime requirement. For real-time applications, the memory and the time of computation play a vital role. So, there is a need of trade-off between these two factors. When the order increases from

**Table 4** Statistical analysis of pbr for FIR low-pass filters (order 20)

| Method | Normalized pass-band ripple | | | |
|---|---|---|---|---|
| | Max. | Mean | Var. | SD |
| GA | 1.172 | 1.072 | 0.0092 | 0.0959 |
| PSO | 1.158 | 1.061 | 0.0076 | 0.0871 |
| CS | 1.149 | 1.052 | 0.0053 | 0.0728 |
| FA | 1.147 | 1.049 | 0.0052 | 0.0721 |
| SCA | 1.150 | 1.046 | 0.0048 | 0.0692 |
| SS | 1.094 | 1.041 | 0.0038 | 0.0616 |
| SSLY | 1.053 | 1.019 | 0.0031 | 0.0556 |

**Table 5** Statistical analysis of pass-band ripple for FIR low-pass filters (order 30)

| Method | Normalized pass-band ripple | | | |
|---|---|---|---|---|
| | Max. | Mean | Var. | SD |
| GA | 1.038 | 1.017 | 0.0087 | 0.0932 |
| PSO | 1.021 | 1.001 | 0.0071 | 0.0842 |
| CS | 1.022 | 0.998 | 0.0051 | 0.0714 |
| FA | 1.029 | 0.999 | 0.0052 | 0.0721 |
| SCA | 1.019 | 0.996 | 0.0046 | 0.0678 |
| SS | 1.021 | 0.994 | 0.0034 | 0.0583 |
| SSLY | 1.017 | 0.988 | 0.0028 | 0.0529 |

**Table 6** Statistical analysis of sbr for FIR low-pass filter (order 20)

| Method | Normalized stop-band ripple | | Transition width |
|---|---|---|---|
| | Maximum | Mean | |
| GA | 0.1487 | 0.1173 | 0.0732 |
| PSO | 0.1368 | 0.1107 | 0.0764 |
| CS | 0.1217 | 0.1064 | 0.0795 |
| FA | 0.1328 | 0.1066 | 0.0762 |
| SCA | 0.1173 | 0.0983 | 0.0764 |
| SS | 0.1079 | 0.0782 | 0.0855 |
| SSLY | 0.0975 | 0.0618 | 0.0945 |

**Table 7** Statistical analysis of sbr for FIR low-pass filters (order 30)

| Method | Normalized stop-band ripple | | Transition width |
|---|---|---|---|
| | Max. | Mean | |
| GA | 0.1095 | 0.0531 | 0.0605 |
| PSO | 0.0919 | 0.0452 | 0.0668 |
| CS | 0.0882 | 0.0411 | 0.0672 |
| FA | 0.0998 | 0.0408 | 0.0674 |
| SCA | 0.0741 | 0.0402 | 0.0680 |
| SS | 0.0778 | 0.0360 | 0.0682 |
| SSLY | 0.0711 | 0.0319 | 0.0687 |

**Table 8** Statistical analysis of stop-band attenuation (sba) for FIR low-pass filter (order 20)

| Method | sba (dB) | | |
|---|---|---|---|
| | Mean | Var. | SD |
| GA | − 18.6212 | − 49.8970 | − 24.9485 |
| PSO | − 19.1214 | − 51.7005 | − 25.8502 |
| CS | − 19.3736 | − 54.8945 | − 27.4472 |
| FA | − 19.3428 | − 54.9606 | − 27.4303 |
| SCA | − 20.1338 | − 55.7654 | − 27.8827 |
| SS | − 22.1402 | − 56.4781 | − 28.2390 |
| SSLY | − 24.1814 | − 58.4613 | − 29.2306 |

**Table 9** Statistical analysis of sba for FIR low-pass filter (order 30)

| Method | sba (dB) | | |
|---|---|---|---|
| | Mean | Var. | SD |
| GA | − 25.4890 | − 50.1727 | − 25.0863 |
| PSO | − 26.8964 | − -52.3957 | − 26.1978 |
| CS | − 27.7302 | − 55.3910 | − 27.6955 |
| FA | − 27.7854 | − 55.3764 | − 27.6882 |
| SCA | − 27.9132 | − 56.2456 | − 28.1228 |
| SS | − 28.8698 | − 57.0774 | − 28.5387 |
| SSLY | − 29.9326 | − 59.1721 | − 29.5860 |

20 to 30, there is an increase in the computational time and the memory requirement. If the computation time is to be reduced, then the hardware requirement increases. This, in turn, increases the computational cost for the processing. So, the trade-off is required in between the computational time and the computational cost while choosing an FIR filter of certain orders. In this paper, the designs of FIR filter of order 20 and 30 are discussed. The idea can also be extended to the design of higher order FIR filter with an increased computation time and memory.

The pass-band ripples and the stop-band ripples play a vital role in the design of the optimized filters with required constraint handling capability. For a perfect digital optimal filter, the pass-band ripple as well as the stop-band ripples should be as minimum as possible or should meet the desired constraints. The stop-band attenuation of the

**Table 10** Qualitative analysis (order 20)

| Method | Min. sba (dB) | Max. average pba (dB) | Exen. time (s) per 100 cycles variance |
|---|---|---|---|
| GA | − 16.60 | 1.0745 | 9.288446 |
| PSO | − 17.69 | 0.9742 | 3.003681 |
| CS | − 17.28 | 0.8462 | 2.561602 |
| FA | − 17.51 | 0.9448 | 3.009822 |
| SCA | − 18.62 | 0.8747 | 3.018784 |
| SS | − 19.34 | 0.5807 | 4.010023 |
| SSLY | − 20.27 | 0.3488 | 4.070722 |

**Table 11** Qualitative analysis (order 30)

| Method | Min.sba (dB) | Max. avg.pba (dB) | Exen. time (s) per 100 cycles variance |
|---|---|---|---|
| GA | − 19.17 | 0.2684 | 9.832849 |
| PSO | − 20.73 | 0.1847 | 3.183201 |
| CS | − 20.96 | 0.1640 | 2.753601 |
| FA | − 20.02 | 0.1812 | 3.125668 |
| SCA | − 22.26 | 0.1628 | 3.258860 |
| SS | − 22.18 | 0.1248 | 4.341829 |
| SSLY | − 22.97 | 0.1102 | 4.420722 |

**Table 12** *t* test result (order 20 LP)

| Algorithm | pbr | | sbr | |
|---|---|---|---|---|
| | *t* | $St_{error}$ | *t* | $St_{error}$ |
| GA | 1.5119 | 0.0350 | 2.6490 | 0.0209 |
| PSO | 1.2853 | 0.0326 | 2.5118 | 0.0194 |
| CS | 1.1392 | 0.0289 | 2.5729 | 0.0173 |
| FA | 1.0419 | 0.0287 | 2.5771 | 0.0172 |
| SCA | 0.9618 | 0.0280 | 2.1740 | 0.0167 |
| SS | 0.8384 | 0.0262 | 0.9989 | 0.0164 |

designed filter should be as minimum as possible. The pass-band as well as the stop-band ripples along with the pass-band and the stop-band edge frequencies are acting as the constraints on the designed objective function. The design parameters are:

Pass-band ripple (pbr) = 0.08.

Stop-band ripple (sbr) = 0.02.

Pass-band (normalized) edge frequency (xp) = 0.40.

Stop-band (normalized) edge frequency (xs) = 0.44.

Transition width = 0.04.

Table 1 signifies the best selected control parameters used for various optimization algorithms. We have experimented with different values for the selection of control parameters. The results are verified over a range of ± 20 percent of the reported value. After extensive simulation work, the best values for control parameters are obtained, which is generally done by the optimization research community. These values are reported in the Table 1. It is significant to declare here that some of the control parameters are directly chosen from the paper as noted in Table 1.

From the simulation study, it is observed that the GA offers wide diversity of new solutions, which are obtained through mutations and crossovers. It seems that the search space is more enhanced (better exploration). However, a higher mutation may drift away the solutions with a premature convergence. That is the reason why we have chosen a very low mutation rate of 0.001. The selection process in GA also exhibits a vital role in the convergence of the algorithm. Through this process, the best solutions are selected and the rest is discarded. Here, the selection probability is 1/3. Further, a two-point crossover technique is deployed with a crossover rate of 0.8.

Table 2 shows the finest optimized coefficients for designing FIR LP filters of order 20, and Table 3 shows similar coefficients for 30 order LP FIR filter. Note that hh($n$) indicates the impulse response, where $n$ is an integer. These coefficients are calculated (optimized) using GA, PSO, CS, FA, SCA, SS and SSLY algorithms. In fact, the design parameters like pass-band and stop-band ripple; pass-band and stop-band normalized edge frequencies, stop-band attenuation and width for transition are determined by these coefficients. Therefore, it is inevitable to obtain optimal filter coefficients to get the best design parameters. Attained design parameters, by using our algorithm (SSLY), are displayed in this section.

Tables 4, 5, 6, 7, 8, 9, 10 and 11 summarize outcome of various performance parameters calculated by means of GA, PSO, CS, FA, SCA, SS and SSLY for LP filters of

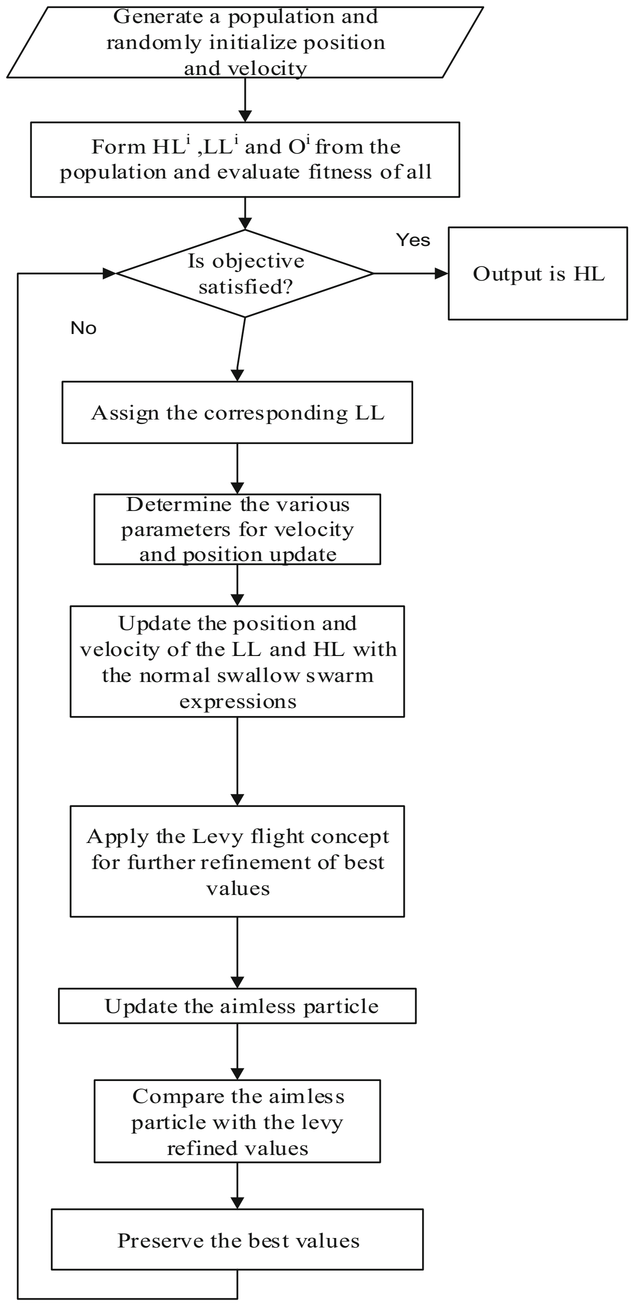



**Fig. 1** Flowchart of the Levy swallow swarm algorithm

orders 20 and 30, respectively. Tables 4 and 5 display values of the normalized pass-band ripple for 20 and 30 order low-pass FIR filters, respectively. The pass-band ripple obtained by the SSLY is the lowest in terms of the mean, max., min., var. and stdv. compared to other algorithms. This shows that the SSLY is very efficient in reduction of the pass-band ripple with the least deviation, which is a desirable property for an effective digital filter. On comparing Tables 4 and 5, it can be easily verified that the ripples are less in 30 order filter designed using the SSLY algorithm.

The statistical analysis of the stop-band ripple is listed in Tables 6 and 7 for the low-pass filter of orders 20 and 30, respectively. From Table 6, it can be easily verified that the maximum normalized stop-band ripple value of the SSLY algorithm is 0.0975, i.e., minimum as compared to other optimization algorithms. Similar conclusions can be drawn from Table 7 for the low-pass filter of order 30.

The stop-band attenuation (sba) plays an essential role in the filter design. More is the stop-band attenuation, better is the performance of a low-pass filter. Using the filter coefficients displayed in Tables 2 and 3, the stop-band attenuations for different orders are computed and presented in Tables 8 and 9. It is observed that the suggested SSLY algorithm offers an average (avg.) stop-band attenuation of − 24.1814 dB for the order 20 and − 29.9326 dB for the order 30 which is best among all other algorithms. The variance and the standard deviation provided by the SSLY are also minimum, i.e., − 58.4613 dB and − 29.2306 dB for order 20 with similar range of values for order 30, respectively. The best part of the SSLY algorithm is that the variance and the standard deviation are the lowest among all algorithms which emphasize the accuracy and precision of the result. The minimum values of the variance and the standard deviation indicate the consistent and precise results.

The execution time for the different algorithms for low-pass FIR filters of order 20 and order 30 is listed in Tables 10 and 11, respectively. GA takes the maximum time, whereas CS takes the least time. CS requires a very few control parameters resulting in a very high speed. Our proposed SSLY algorithm takes 4.42 s, which is close to the second contestant SS algorithm. However, our algorithm performs better than the CS and SS while considering the optimal design parameters. But when the comparison is done in the domain of the transition width, the SSLY is taking a little higher transition width among all the algorithms.

Figures 2 and 3 show normalized frequency response and the magnitude (dB) response of the FIR low-pass filters using GA, PSO, CS, FA, SCA, SS, SSLY algorithms for the order 20 and 30, respectively. Figures 4 and 5 show the enlarged pass band and enlarged stop band of the (20-order) filter. From these figures, it can be easily verified that the normalized pass-band ripple as well as the stop-band ripple is minimized, when filter design is done using the proposed SSLY algorithm. A clearer picture regarding the effectiveness of the SSLY is obtained by Figs. 4 and 5 which indicates the maximum average attenuation.

Figures 6, 7, 8 and 9 describe the normalized frequency response, enlarged pass-band and expanded stop-band, magnitude response in dB for the low-pass filter of order 30 using all seven algorithms, respectively. From these figures, it can be easily verified that the SSLY algorithm has

Author's personal copy



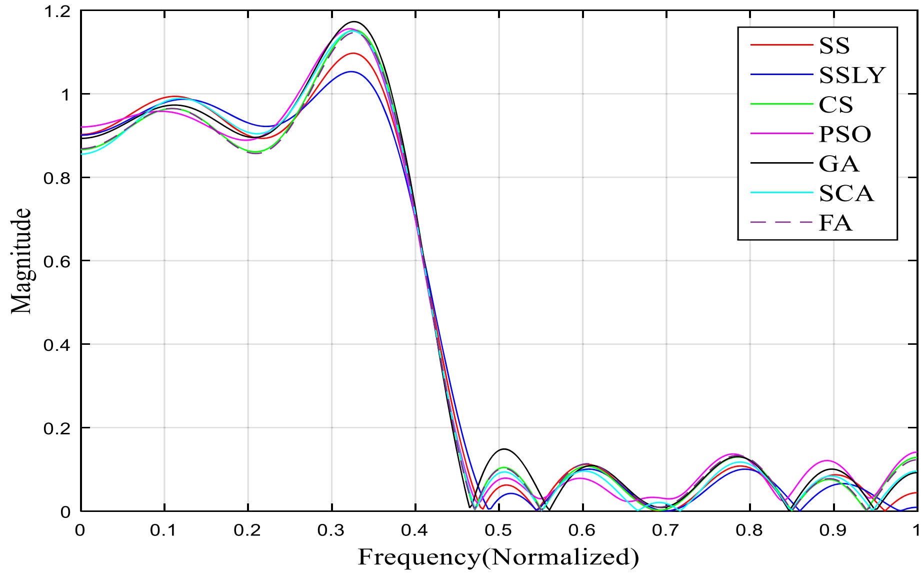



**Fig. 2** Magn. response (order 20 LP)

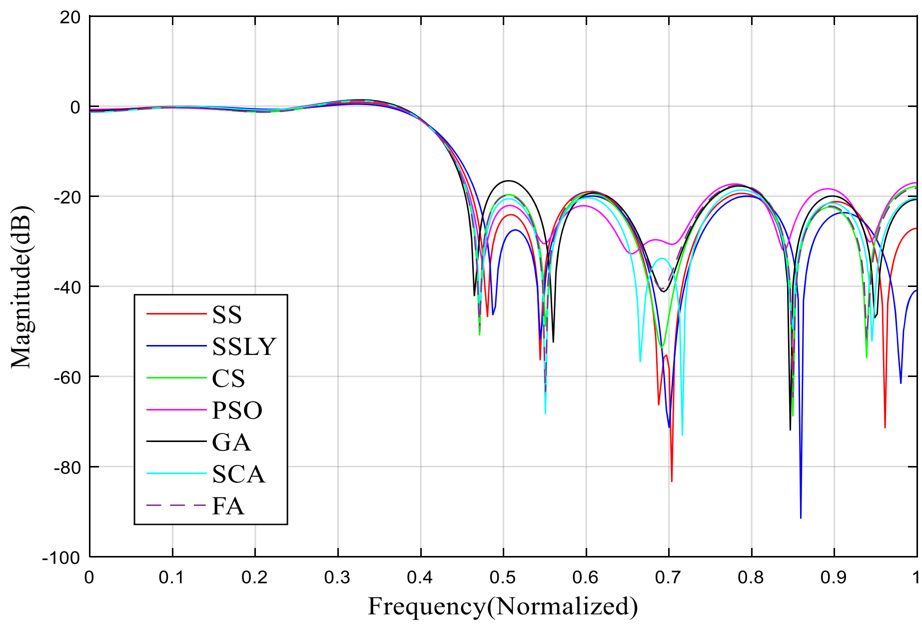



**Fig. 3** Magn. response (dB) (order 20 LP)

the lowest pass-band ripple, the lowest stop-band ripple and the maximum stop-band attenuation.

Figure 10 indicates the convergence profiles of the LP filter of order 20 for different algorithms. Simulation results are also obtained for the LP filters of orders of 30, but not shown here to conserve the space. The profile shows that the SSLY, i.e., the proposed swallow swarm optimization algorithm-based filter design, converges to a great deal of lower error compared to GA, PSO, CS, FA, SCA and SS algorithm.

In summary, the proposed SSLY algorithm technique for the design of the low-pass filter of 20th order results in − 20.27 dB minimum attenuation in the stop band. It also provides a maximum ripple (normalized) of 1.053 in the pass band; maximum ripple (normalized) in the stop band of 0.0975 and a transition width of 0.0945. Same proposed SSLY-based approach to the design of the low-pass filter of 30th order results in a minimum attenuation of − 22.97 dB in the stop band. It also provides a maximum ripple (normalized) in the pass band of 1.017, normalized maximum stop-band ripple of 0.0711 and a width of transition of 0. 0687. The proposed SSLY algorithm shows an improvement of attenuation in the stop band as compared to the SS (previous) algorithm. So, in the stop-band section, filters designed by the SSLY algorithm result in a finer response; and it also shows a better result as compared to all other six algorithms.

Fig. 4 Enlarged pass-band response (order 20 LP)

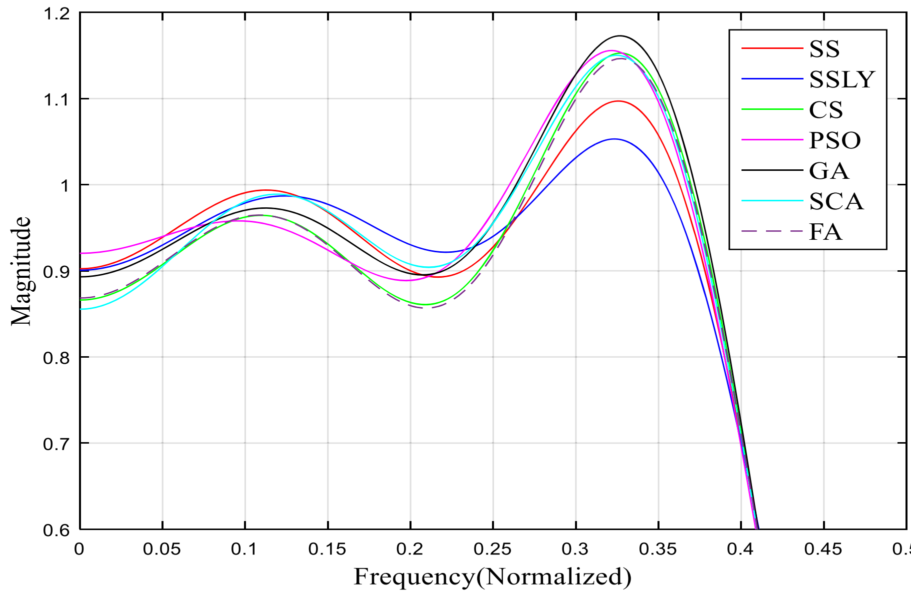
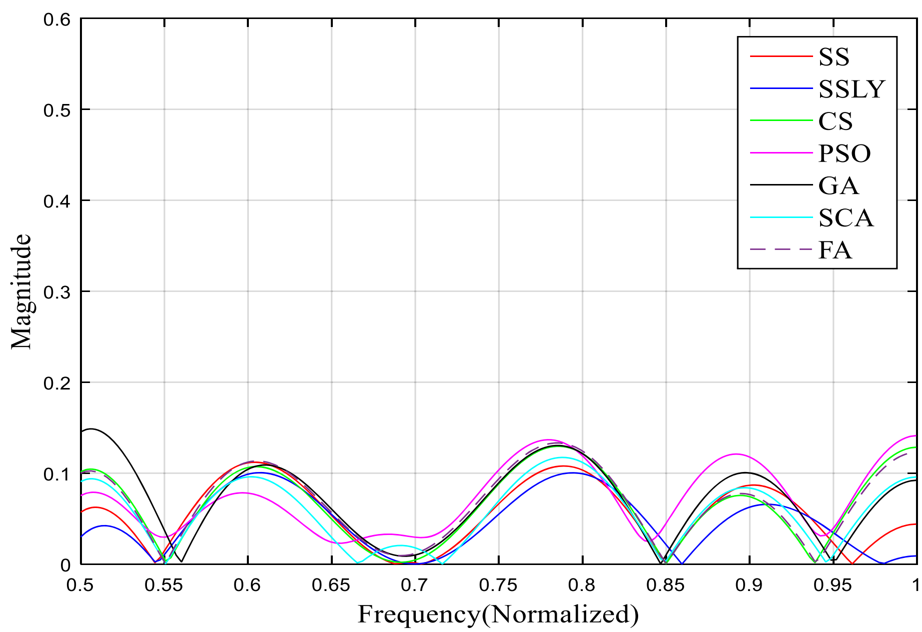



Fig. 5 Expanded stop-band response (order 20LP)




From Tables 4, 5, 6, 7, 8, 9, 10 and 11, it can be finally verified that the filter design by the swallow Swarm optimization algorithm improved by Levy flight is best among others available for this purpose. The figures demonstrate the superiority of the modified Levy swallow swarm algorithm as compared to some conventional evolutionary optimization techniques. From the simulation results, output figures and discussions, it can be verified that with the nearly equal level of transition width, the filter design approach based on a modified SSLY algorithm provides the highest sba (dB) and the lowest sbr with a lowest in the pbr compared to those of GA, conventional PSO, cuckoo search algorithm (CS), firefly algorithm (FA), sine cosine algorithm (SCA) and the conventional swallow swarm optimization algorithm (SS). With a view to the above facts and discussions, it could lastly be incidental that the performance of the modified Levy swallow swarm optimization algorithm with Levy distribution search is the finest among all algorithms.

### 4.1 *t* Test

To further strengthen our claim, a statistical analysis is provided here. Nowadays, the *t* test is used to determine whether the means of 2 sets are statistically dissimilar. This test helps us to validate our algorithm. One can compute

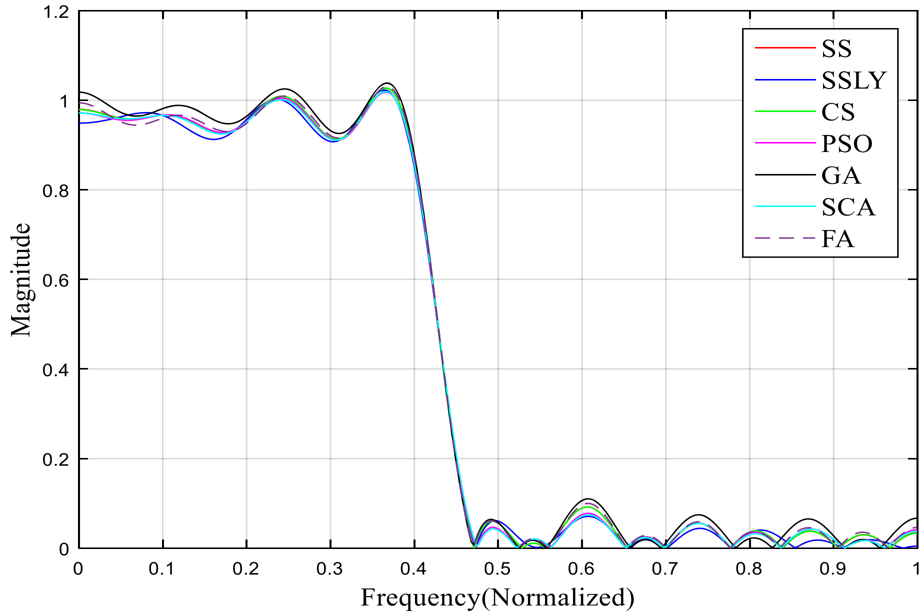



**Fig. 6** Magn. response (order 30 LP)

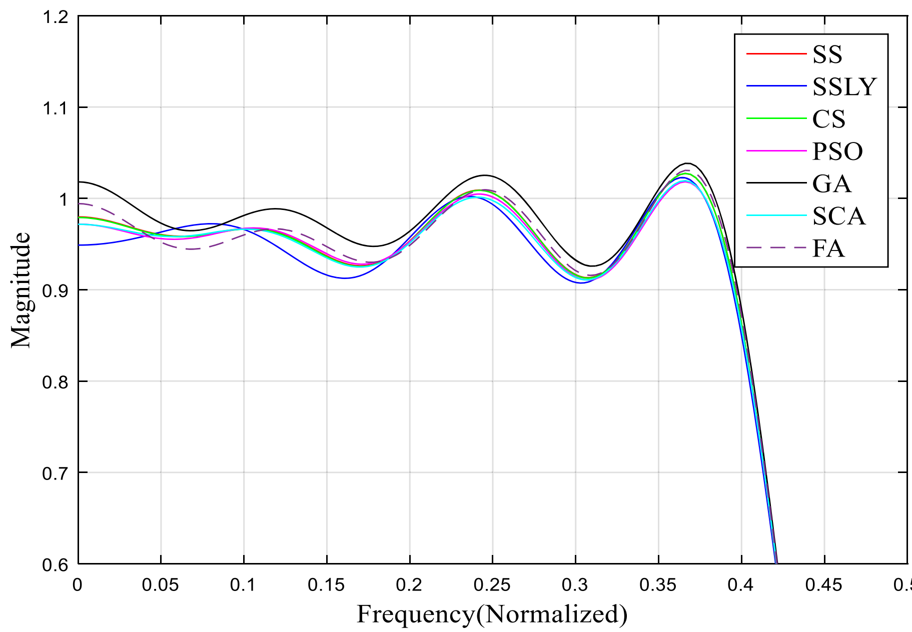



**Fig. 7** Enlarged pass-band response (order 30 LP)

the values of the $t$ test with the help of the following equations.

$$t = \frac{m_o - m_a}{\sqrt{\left(\sigma_o^2/n_o\right) + \left(\sigma_a^2/n_a\right)}} \tag{22}$$

Standard error of difference ($\mathrm{St}_{\mathrm{error}}$) is represented as

$$\mathrm{St}_{\mathrm{error}} = S_p\left(\frac{1}{n_o} + \frac{1}{n_a}\right)^{0.5} \tag{23}$$

The pooled standard deviation $S_p$ is written as

$$S_p = \left(\frac{(n_o - 1)\sigma_o^2 + (n_a - 1)\sigma_a^2}{n_o + n_a - 2}\right)^{0.5} \tag{24}$$

where $m_a$ is the mean value of the SSLY. Here, $m_o$ is the mean value of the rest of the methods considered for a comparison (GA, PSO, CS, FA, SCA and SS). In this section, $\sigma_a$ and $n_a$ denote the stdv. of SSLY and the no. of samples calculated in SSLY, respectively. Further, $\sigma_o$ denotes the stdv. of the rest of the methods (GA, PSO, CS,FA, SCA,SS) while $n_o$ is used to refer to the number of samples calculated in the other methods. These parameters play important role in the statistical analysis.

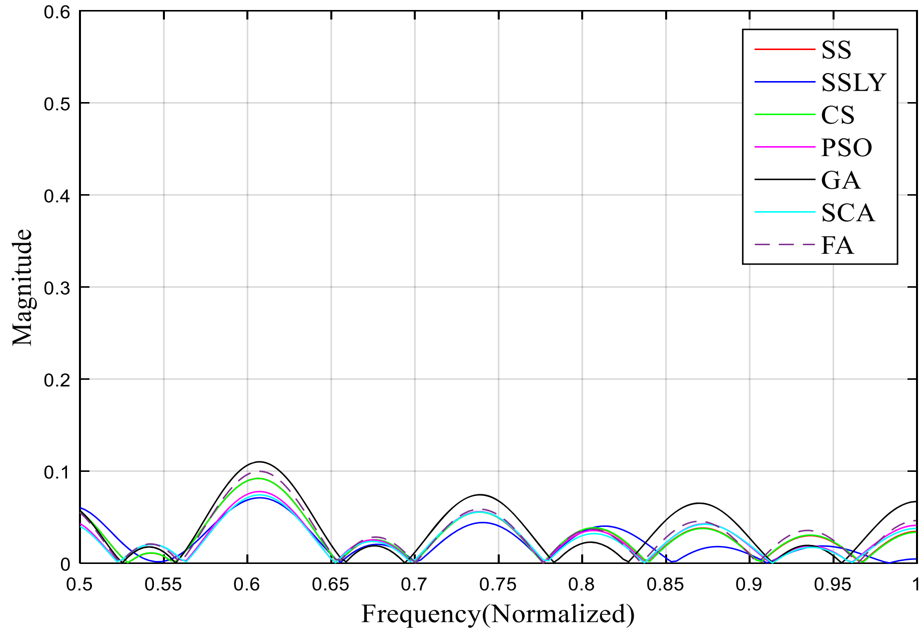



**Fig. 8** Expanded stop-band response (order 30 LP)

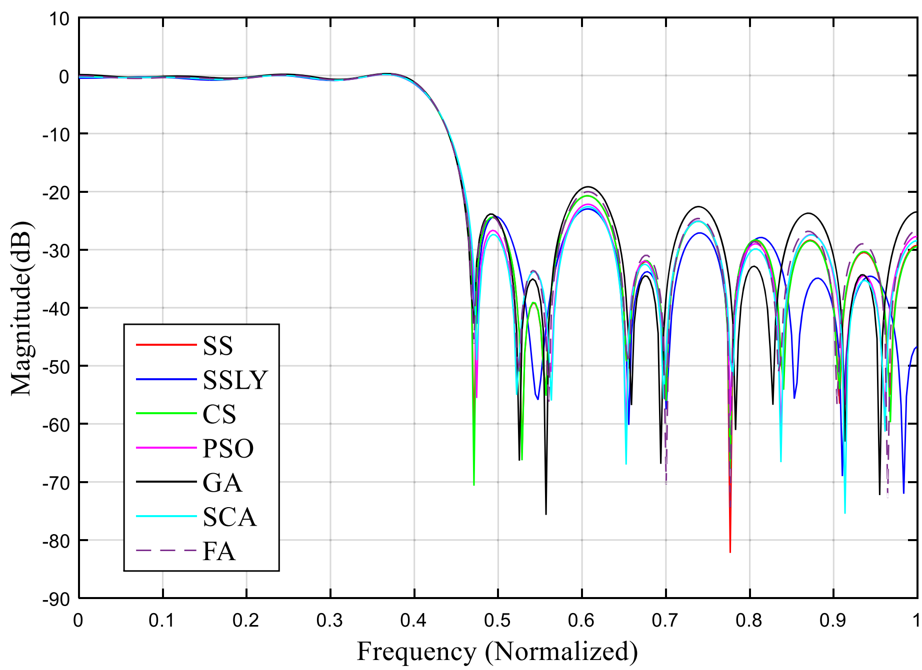



**Fig. 9** Magn. response (dB) (order 30 LP)

A deeper analysis is carried out using *t* test. GA, PSO, CS, FA, SCA or SS are compared with SSLY. From Tables 12 and 13, it is seen that the t values are positive. This shows the dominance of the newly suggested SSLY method over other methods (SS, SCA, FA, CS, PSO or GA). Table 12 clearly shows the dominance of the SSLY over SS, SCA, FA, CS, PSO and GA for designing the LP filter of order 20. It may be reiterated that t values pbr & sbr for SS, SCA, FA, CS, PSO and GA are +ve. It may be noted that $St_{error}$ and the t value of the SS method is lesser than SCA, FA, CS, PSO and GA. This implies that SS is better than SCA, FA, CS, PSO and GA. Table 13 reveals that SSLY has shown a superior act while designing FIR low-pass filter of order 30 compared to SS, SCA, FA, CS, PSO and GA methods. From the above statistical analysis, it is also observed that the t value & $St_{error}$ of SS is min. out of the rest five, which signify its dominance over SCA, FA, CS, PSO and GA methods. In this sense, SS is the second contestant among all 7 methods. To be precise, our proposed SSLY method is the first contestant. It performs well and seems to be well suited for the filter design with a higher accuracy.

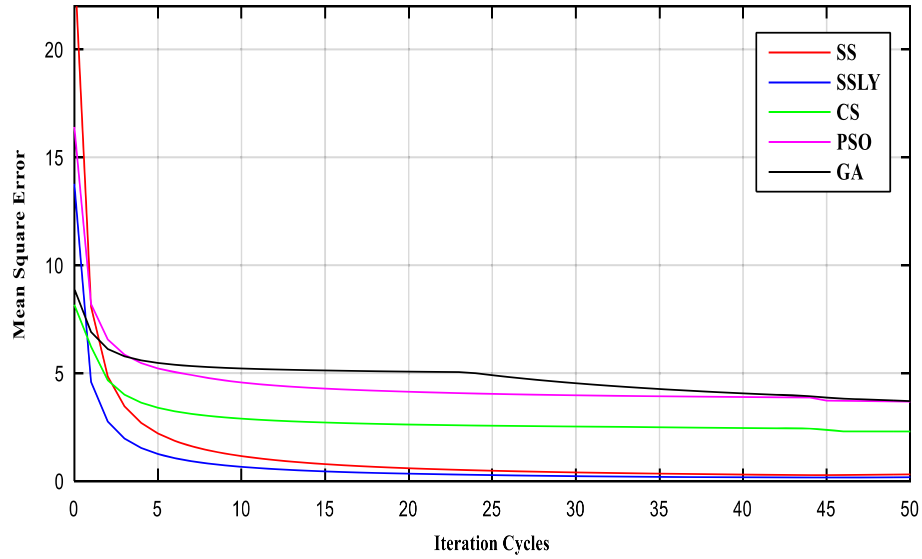



**Fig. 10** MSE versus Iteration

**Table 13** *t* test result (order 30 LP)

| Algorithm | pbr | | sbr | |
|---|---|---|---|---|
| | $t$ | $St_{error}$ | $t$ | $St_{error}$ |
| GA | 0.8551 | 0.0338 | 1.0349 | 0.0204 |
| PSO | 0.4131 | 0.0314 | 0.7110 | 0.0186 |
| CS | 0.3558 | 0.0281 | 0.5497 | 0.0167 |
| FA | 0.3889 | 0.0282 | 0.5318 | 0.0167 |
| SCA | 0.2940 | 0.0272 | 0.5147 | 0.0162 |
| SS | 0.2409 | 0.0248 | 0.2593 | 0.0157 |

## 5 Conclusions

This research develops a new proficient algorithm in swarm intelligence by introducing Levy flight in the swallow swarm optimization. Since the population is distributed in sub-colonies, the particles have the scope to learn from the finest globally experienced particle as well as commencing the most excellent particle of every sub-colony. The suggested SSLY method integrated the concept of searching better with further refinement as well as searching a large space for a better design of optimal filters. An appropriate balance is created amid the global and the local exploration. In this work, the explorer particles are better utilized by the exploited particles; to additionally regulate local searching ability. Our suggested method is tested in the design of an optimal filter by comparing it with the six standard global optimization algorithms. Numerical results verified the effectiveness of the suggested optimization algorithm that outperformed others in the design of an optimal filter with higher stop-band attenuation. The proposed modified SS algorithm may be useful for optimization. Further, the newly proposed objective function may be useful for the design of the maximally flat filters.

## Compliance with ethical standards

**Conflict of interest** The authors declare that they have no conflict of interest.

## References

Ababneh JI, Bataineh MH (2008) Linear phase FIR filter design using particle swarm optimization and genetic algorithms. Digit Signal Process 18(4):657–668

Aggarwal A, Rawat TK, Upadhyay DK (2016) Design of optimal digital FIR filter using evolutionary and swarm optimization techniques. AEU-Int J Electron Commun 70(4):373–385

Ahmad SU, Andreas A (2006) Cascade-form-multiplierless FIR filters design using orthogonal genetic algorithm. In: IEEE international symposium on signal processing and information technology, pp 932–7

Ahmad SU, Antoniou A (2006) A genetic algorithm approach for fractional delay FIR filters. IEEE Int Symp Circuits Syst ISCAS 2006:2517–2520

Chen T, Wang Q, Liu HL (2018) FIR digital filter design based on evolutionary multi-objective algorithm. In: 14th international conference on computational intelligence and security (CIS). IEEE, pp 349–352

Dash J, Dam B, Swain R (2020) Design and implementation of sharp edge FIR filters using hybrid differential evolution particle swarm optimization. AEU-Int J Electron Commun 114(153019):1–16

Deng L, Sun H, Zhang L (2019) A new algorithm (ESA-DE) for designing FIR digital filters. In: International conference on wireless and satellite systems, pp 640–652

Dhabal S, Chakraborty N, Mukherjee A, Biswas J (2016) Design of higher order low pass filter using cuckoo search algorithm. In:

International conference on communication and signal processing, pp 1036–1042
Dwivedi AK, Ghosh S, Londhe ND (2018) Review and analysis of evolutionary optimization-based techniques for FIR filter design. Circuits, Syst Signal Process 37(10):4409–4430
Fang W, Sun J, Xu W, Liu J (2006) FIR digital filters design based on quantum-behaved Particle Swarm Optimization. In: First international conference on innovative computing, information and control, ICICIC '06, vol 1, pp 615–619
Hussain ZM, Sadik AZ, O'Shea P (2011) Digital signal processing: an introduction with MATLAB applications. Springer, Berlin
Ji D (2016) The application of artificial bee colony (ABC) algorithm in FIR filter design. In: 12th international conference on natural computation, fuzzy systems and knowledge discovery (ICNC-FSKD), pp 663–667
Jiménez-Galindo D, Casaseca-de-la-Higuera P, San-José-Revuelta LM (2019) A novel design method for digital FIR/IIR filters based on the shuffle frog-leaping algorithm. In: 2019 27th European signal processing conference (EUSIPCO), pp 1–5
Karaboga N (2009) A new design method based on artificial bee colony algorithm for digital IIR filters. J Frankl Inst 346(4):328–348
Karaboga N, Cetinkaya B (2006) Design of digital FIR filters using differential evolution algorithm. Circuits Syst Signal Process 25(5):649–660
Karaboga N, Cetinkaya MH (2011) An overly and efficient algorithm for adaptive filtering: artificial bee colony algorithm. Turk J Electr Eng Comp Sci 19(1):175–190
Kennedy J, Eberhart R (1995) Particle swarm optimization. Proc IEEE Int Conf Neural Netw 4:1942–1948
Kwan HK (2017) Asymmetric filter design using evolutionary optimization. In: IEEE 30th Canadian conference on electrical and conference engineering, pp 1–4
Liang J, Kwan HK (2017) FIR filter design using multiobjective cuckoo search algorithm. In: IEEE 30th Canadian conference on electrical and computer engineering (CCECE). IEEE, pp 1–4
Mandal S, Ghoshal SP, Kar R, Mandal D (2011) Optimal linear phase FIR band pass filter design using craziness based Particle Swarm Optimization Algorithm. J Shanghai Iaotong Univ (Sci) 16(6):696–703
Mandal S, Ghosal SP, Kar R, Mandal D (2012) Design of optimal linear phase FIR high pass filter using craziness based particle swarm optimization. J King South Univ 24:83–92
Mantegna RN (1994) Fast, accurate algorithm for numerical simulation of Levy stable stochastic processes. Phys Rev 49(5):4683–4977
Mastorakis NE, Gonos IF, Swamy MNS (2003) Design of two-dimensional recursive filters using genetic algorithm. IEEE Trans Circuits Syst I Fundam Theory Appl 50:634–639
Mirjalili S (2016) SCA: a sine cosine algorithm for solving optimization problems. Knowl-Based Syst 96:120–133
Mukherjee A, Chakraborty N, Das BK (2017) Whale optimization algorithm: An implementation to low pass FIR filter. In: International conference on innovations in power and advanced computing technologies [i-PACT2017], pp 1–5
Neshat M, Sepidnam G, Sargolzaei M (2012) Swallow swarm optimization algorithm: a new method to optimization. Neural Comput Appl 2012(23):429–454
Panda R, Agrawal S, Bhuyan S (2013) Edge magnitude based multilevel thresholding using Cuckoo search technique. Expert Syst Appl 40:7617–7628
Parks TW, Burrus CS (1987) Digital filter design. Wiley, New York
Parks TW, McClellan JH (1972) Chebyshev approximation for non-recursive digital filters with linear phase. IEEE Trans Circuit Theory CT-19:189–194
Rana KPS, Kumar V, Nair SS (2016) Efficient FIR filter designs using constrained genetic algorithms based optimization. In: IEEE 2nd international conference on communication, control and intelligent systems (CCIS), pp 1–5
Rashedi E, Hossien N, Saryazdi S (2011) Filter modelling using gravitational search algorithm. Eng Appl Artif Intell 24(1):117–122
Ravi RV, Subramaniam K, Roshini TV, Muthusamy SPB, Venkatesan GP (2019) Optimization algorithms, an effective tool for the design of digital filters: a review. J Ambient Intell Human Comput. https://doi.org/10.1007/s12652-019-01431-x
Saha SK, Ghosal SP, Kar R, Mandal D (2013) Cat swarm optimization algorithm for optimal linear phase FIR design. ISA Trans 52:1–14
San-José-Revuelta M (2018) Design of optimal frequency-selective FIR filters using a memetic algorithm Luis. In: 26th European signal processing conference (EUSIPCO), pp 1172–1176
Sarangi SK, Panda R, Dash M (2014) Design of 1-D and 2-D recursive filters using crossover bacterial foraging and Cuckoo Search techniques. Eng Appl Artif Intell 34:109–121
Sarangi SK, Panda R, Das PK, Abraham A (2018) Design of optimal high pass and band stop FIR filters using adaptive Cuckoo search algorithm. Eng Appl Artif Intell 70:67–80
Yang XS (2009) Firefly algorithms for multimodal optimization. In: Proceedings of the 5th international conference on stochastic algorithms foundations and applications, LNCS Springer, vol 5792, pp 169–178
Yang XS, Deb S (2009) Cuckoo search via levy flights. In: World congress on nature & biologically inspired computing, pp 210–214
Yang XS, Deb S (2010) Engineering optimization by cuckoo search. Int J Math Model Numer Optim 1(4):330–343
Zhang M, Kwan HK (2017) FIR filter design using multiobjective teaching-learning-based optimization. In: IEEE 30th Canadian conference on electrical and computer engineering (CCECE), pp 1–3