Efficient Design of FIR filter using Gbest-guided Cuckoo Search Algorithm

Supriya Dhabal, Adrika Mukherjee, Niloy Chakraborty, Jayeta Biswas

Abstract: This paper presents an efficient design of Finite Impulse Response (FIR) filter using Gbest-guided Cuckoo Search (GCS) algorithm. To reduce the parameter dependency in traditional Cuckoo Search Algorithm (CSA) as well as better searching of optimal coefficient in the filter designing problems, some modification in the approach of CSA is done in the proposed GCS, which results in a noteworthy faster convergence with an optimal solution. Here, low-pass and band-pass filters are designed, for both Type1 and Type 2 FIR, using GCS, CSA, and Artificial Bee Colony (ABC) algorithm. From the graphical and statistical observations of GCS, CSA, and ABC, it is clear that GCS surpass its competitors in terms of convergence rate, execution time, and filter response. Furthermore, the stop-band attenuation (A_s) and pass-band ripple (R_p) obtained for the filters developed using the proposed GCS is much better than that designed using CSA, ABC or, Parks and McClellan method. An average study suggests that GCS shows 9.23% increase in A_s and 26.05% decrease in R_p for lower order LPF(for both Type 1 and Type 2, order less than 40); 4.27% increase in A_s and 18.85% decrease in R_p for higher order LPF(for both Type 1 and Type 2, order greater than 40 and less than 100); 32.47% increase in A_s and 28.18% decrease in R_p for lower order BPF(for both Type 1 and Type 2, order less than 40); 6.92% increase in A_s and 10.44% decrease in R_p for higher order BPF(for both Type 1 and Type 2, order greater than 40 and less than 100), compared to PM.

***Index Terms*— FIR Filter, GCS, CSA, Lévy Flight, ABC.**

1. Introduction

The design of optimum digital filter involves the finding of appropriate filter coefficients such that the filter response obtained holds a levelled pass-band and maximum stop-band attenuation possible [1]. Digital filters are superior to analog ones in the context of repeatability, flexibility, ability to store signal, portability, versatility, accuracy, and stability. Currently, digital filters play a very significant role in numerous areas, such as in audio and speech processing, military, image processing, biomedical engineering etc. Digital filters are mainly classified as Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) Filter. FIR filters are non-recursive and when an impulse signal is given as input then the corresponding output will fade away within a certain amount of time in contrast to IIR filters. FIR filters beat IIR filters since they show greater stability and possess linear phase.

The most frequently used methods, to design digital FIR filter are window method [2], frequency sampling, uniform approximation etc. The most effortless way is the traditional windowing method where the absolute impulse response is multiplied with fitting window function depending on the design requirements. Another customarily used method to acquire an optimum filter is a method adopted on least squares (LS), where error function is approximated in L2-norm form. To overcome the flaws in the filter output at discontinuities generated by LS technique, minimax method is put to use [1]. In this method, the Chebyshev error is generated, and an equiripple filter is obtained depending on either L1-norm or L∞-norm [1]. However the above-mentioned techniques have some flaws and filter designing being a challenging engineering problem can now be readily solved by various heuristic algorithms.

The process of designing digital filters using these algorithms have an objective of minimizing the error function. The error function or cost function is the difference between the desired response of the filter and the derived response. With the development of meta-heuristic algorithms, the requirement of cost functions which are differentiable and continuous, applicable for conventional gradient-based schemes, are not in use anymore. In addition to this generation of global optima throughout the search space has become simpler. Algorithms like, Gravitational Search Algorithm [3-5], Artificial Bee Colony Algorithm [6-8], Cuckoo Search Algorithm [9-16], Genetic Algorithm [17], Simulated Annealing [18], Particle Swarm Optimization [19-23], are significant computational techniques developed a while back. Most of these algorithms may not provide accurate results but provides approximated and quite satisfactory solution which is closer to the accurate result. Here, GCS is used for designing FIR Type 1 and Type 2 low-pass (LP) as well as band-pass (BP) filter, where not only a justified balance is maintained between pass-band ripple and stop-band attenuation in respect to other heuristic algorithms like CSA, PM and, ABC, but also a satisfactory computational efficiency is achieved.

2. Problem Formulation For Low-Pass and Band-Pass FIR Filter

The system transfer function $H(z)$ used in designing FIR Filter is given by [22]:

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

which has (M-1) trivial poles and (M-1) zeroes located anywhere in z plane.

2.1 Error Function Representation

In reality, by changing degrees of favorable outcomes and minimizing the error between approximated and the ideal response, the desired response is obtained. Weighted difference of the ideal and approximated frequency responses in stop-band, as well as pass-band, defines the error function as given in [22]:

$$E(\omega) = W(\omega)\left[\, H_d\left(e^{j\omega}\right) - H_i\left(e^{j\omega}\right)\,\right] \tag{2}$$

The error function $E(\omega)$, given by Parks-McClellan (PM) is represented in Eq. (2), where $W(\omega), H_d\left(e^{j\omega}\right)$, $H_a\left(e^{j\omega}\right)$ are the weight vector, ideal and, approximated frequency response in order. Weight function $W(\omega)$ modulates the minimization of error.

In Parks-McClellan error function, peak pass-band to stop-band ripple (δ_p/δ_s) ripple cannot take different values, so in order to rout the flaws in this function, a modified error function is put to use in [22]:

$$U = \max\left(\left|E(\omega)\right| - \delta_p\right) + \max\left(\left|E(\omega)\right| - \delta_s\right), \omega \le \omega_p \;\; \omega \ge \omega_s \tag{3}$$

where $\delta_p, \delta_s, \omega_p, \omega_s$ denotes the pass-band ripple, pass-band frequency, stop-band ripple and, stop-band frequency respectively. Now, the low-pass filter's ideal response can be denoted by [22]:

$$\begin{aligned} H_d\left(e^{j\omega}\right) &= 1, 0 \le \omega \le \omega_p \\ &= 0, \omega \ge \omega_s \end{aligned} \tag{4}$$

Similarly, the ideal response of band-pass filter is

$$\begin{aligned} H_d\left(e^{j\omega}\right) &= 0 \qquad 0 \le \omega \le \omega_{s1} \\ &= 1 \qquad \omega_{p1} \le \omega \le \omega_{p2} \\ &= 0 \quad \omega \ge \omega_{s2} \end{aligned} \tag{5}$$

Where ω_{s1}, ω_{s2} represents first and the second stop-band frequency of band-pass FIR filter. Similarly ω_{p1} and ω_{p2} are first and the second pass-band frequency of band-pass FIR filter.

2.2 *Type 1 and Type 2 Linear Phase FIR Filter*

The frequency response function of FIR filter is given by [24]:

$$H(e^{j\omega}) = \sum_{n=0}^{(M-1)} h(n) e^{-j\omega n}, -\pi < \omega \le \pi \tag{6}$$

Now linear phase constraint is imposed by [24]:

$$\angle H(e^{j\omega}) = -\alpha\omega, -\pi < \omega \le \pi \tag{7}$$

Here α is a constant phase delay .Now for Type 1 and Type 2 filters, $h(n)$ must be symmetric, i.e. [24]:

$$h(n) = h(M-1-n), 0 \le n \le (M-1) \text{ with } \alpha = \frac{M-1}{2} \tag{8}$$

Here $h(n)$ is symmetric about α, which is the index of symmetry [24]. The value of M, in Eq. (6), can take even or odd integer and hence two types of filters comes into the picture.

Type 1 FIR filter has symmetrical impulse response with M odd. In this case α is an integer (from Eq. (8)), and $h(n) = h(M-1-n)$, where $0 \le n \le M-1$. Then the response of the Type 1 filter can be written as [24]:

$$H(e^{j\omega}) = \left[\sum_{n=0}^{(M-1)/2} a(n)\cos\omega n\right] e^{-j\omega(M-1)/2} \tag{9}$$

Type 2 FIR filter has symmetrical impulse response where M is even. In this case α is not an integer (from Eq. (8)), and $h(n) = h(M-1-n)$, where $0 \le n \le M-1$. Then the response of the Type 2 filter can be written as [24]:

$$H(e^{j\omega}) = \left[\sum_{n=1}^{M/2} b(n)\cos\{\omega(n-0.5)\}\right] e^{-j\omega(M-1)/2} \tag{10}$$

2.3 Cost Function Formulation for Low-Pass and Band-Pass FIR Filter

In this paper, instead of mean square based error function, the cost function for low-pass is modified as follows [22]:

$$\phi = \alpha * E_P + (1-\alpha) * E_S,\ 0 < \alpha \le 1 \tag{11}$$

where E_P and E_S are pass-band and stop-band error for low-pass FIR filter which can be formulated as [22]:

$$E_S = \frac{1}{\pi}\int_{\omega_s}^{\pi} H(\omega)^2 d\omega = b^T C b \tag{12}$$

$$E_P = \frac{1}{\pi}\int_{0}^{\omega_p} (1-H(\omega))^2 d\omega = \left(\frac{\omega_P}{\pi}\right) - 2b^T P + b^T Q b \tag{13}$$

where $H(\omega) = b^T C(\omega)$, $b = [b_1, b_2, ..., b_{N/2}]^T$. P and, Q can be defined as:

$$P = \frac{1}{\pi}\int_{0}^{\omega_p} \cos(A\omega)d\omega \tag{14}$$

$$Q = \frac{1}{\pi}\int_{0}^{\omega_p} \cos(A\omega)\cos(B\omega)d\omega \tag{15}$$

where $A = \frac{N-1}{2} - m$, $m = 0,1,...(M-1)$ and $B = \frac{N-1}{2} - n$ Where $n = 0,1,...(M-1)$.

Now, in case of pass-band error, the value of C can be measured from the formula as following:

$$C(m,n) = \frac{1}{\pi}\int_{\omega_s}^{\pi} \cos(A\omega)\cos(B\omega)d\omega \tag{16}$$

In case of band-pass FIR filter the cost function is modified as follows

$$\phi = \alpha * E_p + (1-\alpha) * (E_{S1} + E_{S2}), 0 < \alpha \le 1 \tag{17}$$

where E_P , E_{S1} and E_{S2} can be formulated as [22]:

$$E_{S1} = \frac{1}{\pi}\int_{0}^{\omega_{s1}} (H(\omega))^2 d\omega = b_1{}^T C_1 b_1 \tag{18}$$

$$E_P = \frac{1}{\pi}\int_{\omega_{p1}}^{\omega_{p2}} (1-H(\omega))^2 d\omega = \left(\frac{\omega_{P2} - \omega_{p1}}{\pi}\right) - 2b_1{}^T P_1 + b_1{}^T Q_1 b_1 \tag{19}$$

$$E_{S2} = \frac{1}{\pi}\int_{\omega_{s2}}^{\pi} (H(\omega))^2 d\omega = b_1{}^T C_1 b_1 \tag{20}$$

where E_{S1} and, E_{S2} are first and second stop-band error for band-pass FIR filter. b_1, C_1, P_1 and, Q_1 are same as b, C, P and Q ; as described in Eq.(14) , Eq.(15), and Eq.(16). $H(\omega)$ is the magnitude response of the filter. Therefore, the design task is to minimize the weighted combination of E_p and E_s such that both are minimized simultaneously.

3. Design Methodologies

3.1 Cuckoo Search Algorithm

Cuckoo Search Algorithm (CSA) is a mathematical approach formulated to solve many issues on optimization, suggested by Yang and Deb [9]. This agent-based algorithm is motivated by brood parasitism of cuckoo birds and the theory of lévy flight.

Cuckoos follow unique parasitism; they reproduce eggs and lay it in other bird’s nests which is cited here as the host bird. Usually, offspring of cuckoos emerges prior to that of host birds’ and thus chicks of cuckoo have more likelihood to have food as they can propel the eggs of host bird away from the cuckoos' breeding ground. Accidentally, if any unfamiliar chick is discovered by the host bird, it either aggressively drives away the chick or leaves the nest, putting up a new nest elsewhere.

Lévy flight [12-13], where steps follow isotropic random direction in the space having multiple dimensions, is an exceptional condition of a random walk with a probability distribution that is heavy-tailed. The probability of discovering their food depends on step length, which stays within a specific bound ($1< \lambda \leq 3$). Lévy distribution function gives the step length of Lévy flight, also called as random walk is followed by this equation:

$$\text{lévy}(\lambda) = \left| \frac{\Gamma(1+\lambda)\times \sin(\pi\lambda/2)}{\Gamma((1+\lambda)/2)\times\lambda\times 2^{((\lambda-1)/2)}} \right|^{1/\lambda} \tag{21}$$

CSA is based on three assumptions [9] motivated by the conduct of cuckoo birds:

1) Every cuckoo dumps its eggs one by one in the host birds' nest randomly, where each egg is represented as a possible solution.
2) The qualitative egg having higher fitness value shall be imparted to the subsequent generation.
3) Host nests are fixed in number and the probability (Pa) that the host bird will detect a cuckoo egg remains within the boundary [0, 1].

In this situation the host bird either drives the cuckoo's egg away from the nest or they prefer to leave the nest.

Cuckoos pursue the postulate of Lévy flight when they lay eggs in random fashion where a new nest is generated followed by [10].

$$x_i^{t+1} = x_i^t + \alpha_1 \oplus \text{Lévy}(\lambda) \tag{22}$$

Here α_1 stands for step size (α_1>0). '$\oplus$'represents entry wise multiplication. We can obtain an improved search with the help of lévy flight where steps are derived from the lévy distribution function of Eq. (1). The isotropic and random steps random maintain a definite probability distribution depicted by [9]:

$$\text{Lévy} \sim u = t^{-\lambda},\ (1<\lambda<3) \tag{23}$$

Algorithmic steps of CSA:

Step 1: Initialize a population of n host nest size, denoted by x_i , where $(i = 1,2,\ldots, n)$.

Step 2: Find the fitness value of function F_i =f (x_i) for all x_i .

Step 3: Until the stopping criteria is satisfied, do steps 4-7.

Step 4: Randomly initiate cuckoo egg x_j from the host nests using the formulation of Lévy flight and find its fitness (F_j).

Step 5: If $F_i > F_j$ then replace x_i with x_j as well as replace F_i with F_j.

Step 6: Abandon P_a fraction of the inauspicious nests and randomly generate that amount of new nests.

Step 7: Calculate contemporary best; keep the best solutions and order the solution sequentially.

Step 8: Process the result and finally visualize.

3.2 Artificial Bee Colony Algorithm

Artificial Bee Colony Algorithm (ABC) has been a recently introduced optimization algorithm, motivated by the intellectual nature of swarms and suggested by Karaboga [6]. In ABC there are three varieties of agents categorized as employed bees, onlooker bees and the scout bees. The employed bees engage themselves in locating new food sources in the proximity of explored nectar sources and forward the knowledge they gather about the food source to the onlooker bees based on the nectar amount. The onlooker bees make the decision to select a latest food source depending on the information collected from employed bees through waggle dance. Scout bees randomly investigate for another food source in the proximity of the colony. By this procedure, employed bees and the onlookers perform the task of exploitation and the exploration part is accomplished by the scouts. Here every food source being viewed as n-dimensional real-valued vector, equals to our possible solution of the problem. Quantity of nectar corresponding to the resource is defined as the quality or "fitness value" of that food source.

The random position of initial food sources is calculated following the scope of the circumferences of parameters. The i^{th} Food source is represented as $X_i(1,2,....SN)$ which specifies a vector having D dimensions, $j = 1,2,...., D$ and $rand(0,1) \in [0,1]$ provides a random number. A unique food resource is discovered by the scouts is given by [7]:

$$X_{ij} = X_{min,j} + rand(0,1) \times (X_{max,j} - X_{min,j}) \tag{24}$$

Here upper bound is indicated by $X_{max,j}$ whereas lower bound is denoted as $X_{min,j}$. In order to renovate beneficial solutions, employed bee develops a new candidate food position in the locality of its previously explored position. The search for this assignment is achieved by the subsequent expression [8]:

$$V_{ij} = X_{ij} + R_{ij} \times (X_{ij} - X_{kj}) \tag{25}$$

where $k = 1,2,....,SN$ and $j = 1,2,...., D$ are randomly selected and k should be different from j, R_{ij} is takes a random value in between $[-1,1]$ and this controls the production of food source in the neighbourhood. V_{ij} is the new solution and X_{ij} is the value of the previous solution. Comparison is made between the fitness values of both V_{ij} and X_{ij}, if the latter is less than or equal to the former one then V_{ij}

substitutes X_{ij}; else X_{ij} will be remembered. The probability of selection of onlooker bees depends on the assessment of fitness value of the output at the time of estimation of nectar amount. The probability established on fitness is described as [6]:

$$P_i = \frac{fit_j}{\sum_{n=1}^{SN} fit_n} \tag{26}$$

where fit_j denotes the fitness value of j^{th} solution.

The framework of ABC to obtain prototype filter coefficients is described as follows:

Algorithmic steps of ABC:

Step 1: Initialize the population of solutions $X_i\,(1,2,....SN)$ and evaluate the population.

Step 2: Produce new solutions V_{ij} for the employed bees using Eq. (25) and evaluate them.

Step 3: Calculate the probability values using Eq. (26) and employed bees arrive back to the hive to share the information of food sources with onlooker bees.

Step 4: Based on the nectar quantity of food sources, update onlooker bees and memorize the food source which is best.

Step 5: If an employed bee cannot improve its solution quality within limit trails then it becomes a scout bee and is substituted by new food source using Eq. (24).

Step 6: When the maximum limit is reached, stop algorithm. Otherwise, revisit to Step 2.

3.3 Proposed Gbest-guided Cuckoo Search Algorithm

In the standard CSA, random walks based search equations cannot guarantee a fast convergence. Therefore a proposed algorithm is used in order to have faster convergence and auto-tuning where three modifications have been incorporated. The modifications are as follows:

3.3.1 Modification in replacement strategy:

In the original CSA, the convergence speed becomes less as the old nests are replaced randomly. In the proposed algorithm old nests have been replaced based on the global best solution to have better control step size instead of random searching like CSA. The modified equation is presented below:

$$nest_{new} = nest_{old} + rand * (nest_{best} - nest_{old}) \oplus K \text{ if } K > P_a \tag{27}$$

where $nest_{old}$, $nest_{best}$ represents the permutation matrix obtained from old and best nest respectively and $nest_{new}$ denotes the new nest, which is generated in current iteration. In this manner, the generated new nests are dependent upon the best nests obtained till now and hence the proposed algorithm is named as Gbest-guided Cuckoo Search Algorithm.

3.3.2 Modification in λ :

In CSA the Lévy distribution function has been directly adopted where a fixed value of $\lambda = 1.5$ in Lévy's distribution is assumed. In GCS, a new formulation of finding λ is used so as to have better exploration the searching of search-space. Here λ has been varied as the following equation:

$$\lambda = (\lambda_{max} - \lambda_{min}) \times \frac{(iter_{max} - iter)}{iter_{max}} + \lambda_{min} \tag{28}$$

where $\lambda_{max} \rightarrow \lambda_{min} = 1.5 \rightarrow 1$ and $iter_{max}$ and $iter$ represents maximum and current iteration respectively.

3.3.3 Modification in p_a *:*

Yang and Deb suggested that the CSA outperforms GA and PSO in terms of the number of parameters to be tuned, only the probability of abandoned nests p_a. According to their statement CSA converges insensitively to p_a and the setting of p_a = 0.25 is sufficient for all problems But in case of complex and multimodal design problems the experimental data affirms that the convergence rate can be improved through the suitable adjustments of the parameter p_a. Therefore, p_a is varied in order to make it self-tuned based on the following equation:

$$P_a = rand / D \tag{29}$$

where D indicates the dimension of the problem and *rand* represents a random number $\in [0,1]$ .The flowchart of the proposed GCS is shown in Fig.1.

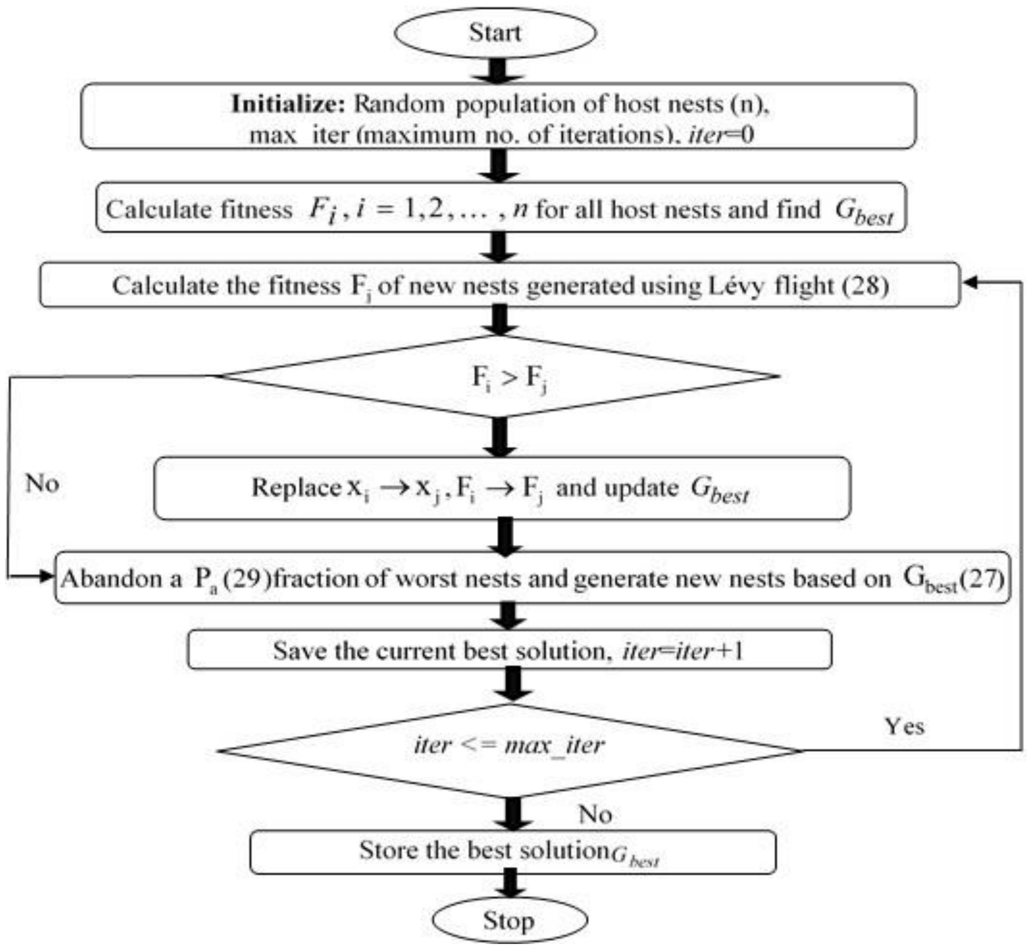


Fig.1. *Flowchart of GCS algorithm*

*Algorithmic steps of GCS**:***

Step 1: Initialize a population of n host nest size, denoted by x_i, where $(i = 1,2,\ldots,n)$.

Step 2: Find the fitness value of function F_i=f (x_i) for all x_i.

Step 3: Until the stopping criteria is satisfied, do steps 4-7.

Step 4: Randomly initiate cuckoo egg x_j from the host nests using the formulation of Lévy flight with proposed value of λ and find its fitness function (F_j).

Step 5: If $F_i > F_j$ then replace x_i with x_j as well as replace F_i with F_j.

Step 6: Abandon P_a fraction of the inauspicious nests using Eq. (29) and randomly generate that amount of new nests using Eq. (27) to overcome the nests lost.

Step 7: Keep the best solutions in order to rank each solution and find current best.

Step 8: Process the obtained results and visualize.

4. Simulation Result and Discussion

FIR filter cost function is minimized using GCS, CSA, PM, ABC and finally, the obtained simulation results are compared from the aspect of stop-band attenuation, ripple, and computational efficiency. After the MATLAB implementation, each algorithm is executed at least 50 times in a latest Intel core i5, 2.20 GHz processor PC.

4.1 Parameter Settings

Tuning of control parameters is a challenging task since the performance of an algorithm can be greatly influenced by them. Though there is no such specific method to tune the parameters, researchers suggested extensive studies on the parameters and then tune them within a specific range. GCS algorithm is the most advantageous, for the fact that it is parameter independent, hence, implementation of GCS becomes simpler. In this paper, to compare meta-heuristic algorithms, all algorithms are simulated to obtain the best possible results in A_s and R_p, by substantial study and immense random tuning of parameters. Parameter settings of all the algorithms that are used in this paper is listed in Table 1 and Table 2.

Table 1 Parameter Settings of ABC, CSA, and GCS for LPF

Order (Type 1/Type 2)	Method	Population Size	Iteration	P_a	α	Limit
18/19	ABC	80	3000	---	0.14	50
	CSA	100	2000	0.001	0.13	---
	GCS	120	3000	---	0.1	---
38/39	ABC	100	3000	---	0.32	50
	CSA	100	2000	0.001	0.27	---
	GCS	100	2000	---	0.2	---
58/59	ABC	150	3000	0.001	0.65	50
	CSA	100	3000	---	0.65	---
	GCS	100	3000	---	0.47	---
78/79	ABC	220	5000	0.0001	0.72	100
	CSA	100	2000	---	0.62	---
	GCS	100	5000	---	0.58	---
98/99	ABC	1000	15000	0.0001	0.7	150
	CSA	120	3000	---	0.8	---
	GCS	150	3000	---	0.75	---

Order (Type 1/Type 2)	Method	Population Size	Iteration	P_a	α	Limit
18/19	ABC	50	1000	---	0.22	50
	CSA	25	1000	0.1	0.22	---
	GCS	25	1000	---	0.22	---
38/39	ABC	50	1000	---	0.33	50
	CSA	50	1000	0.01	0.3	---
	GCS	50	1000	---	0.3	---
58/59	ABC	200	2000	---	0.35	50
	CSA	200	1000	0.001	0.37	---
	GCS	200	1000	---	0.4	---
78/79	ABC	300	4000	---	0.45	100
	CSA	200	2000	0.0001	0.4	---
	GCS	200	2000	---	0.4	---
98/99	ABC	1000	15000	---	0.55	150
	CSA	200	5000	0.0008	0.5	---
	GCS	300	5000	---	0.5	---

Table 2 Parameter Settings of ABC, CSA, and GCS for BPF

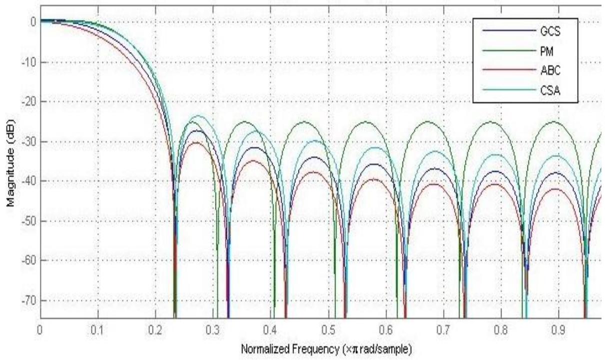
4.2 Impulse response of the Filter

Based on the recommended parameter settings mentioned in Table 1 and Table 2, GCS is simulated for band-pass and low-pass FIR filter (Type 1 and Type 2) and then the obtained magnitude response is compared with that of CSA, ABC. To demonstrate the potency of these algorithms, the standard Parks McClellan’s method is engaged in the comparison also. Due to the fairness of analysis and juxtaposition, the desired specifications of the filter were kept unchanged.

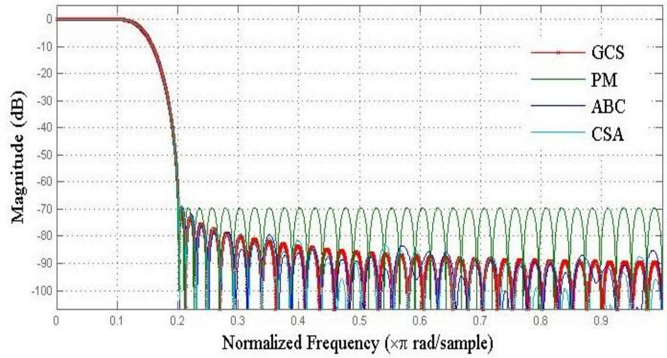
4.2.1 Low-pass FIR filter:

The design specification for Type1 and Type 2, low-pass filter is: pass-band frequency $\omega_p = 0.1\times pi$ and stop-band frequency $\omega_s = 0.2\times pi$. The design objective of both Type of low-pass filters has been discussed in Eq. (11-16). The obtained response of 79th order Type 2, low-pass FIR filter has been shown in Fig.2 as an example. It’s very clear from the figure that among all methods (i.e. GCS, PM, ABC, and

CSA), GCS shows the ripple of least magnitude response. Similarly in case of Type 1 filters, the same statement is justified.



(a)



(b)

Fig.2. *Output of Type 1and Type 2 low-pass FIR filter*

(a) 18th order Type 1, low-pass FIR filter; (b) 79th order Type 2, low-pass FIR filter

Detailed results, showing Ripple (R_p), Stop-band attenuation (A_s), Computational time required to get the optimal solution for different orders ranging from 10 to 100, are reflected in Table 3. Data from Table 3 suggests that GCS provides least R_p and maximum A_s in almost all cases; furthermore, the execution time of GCS is less.

Table 3 Simulation results for Type 1 and Type 2 LPF

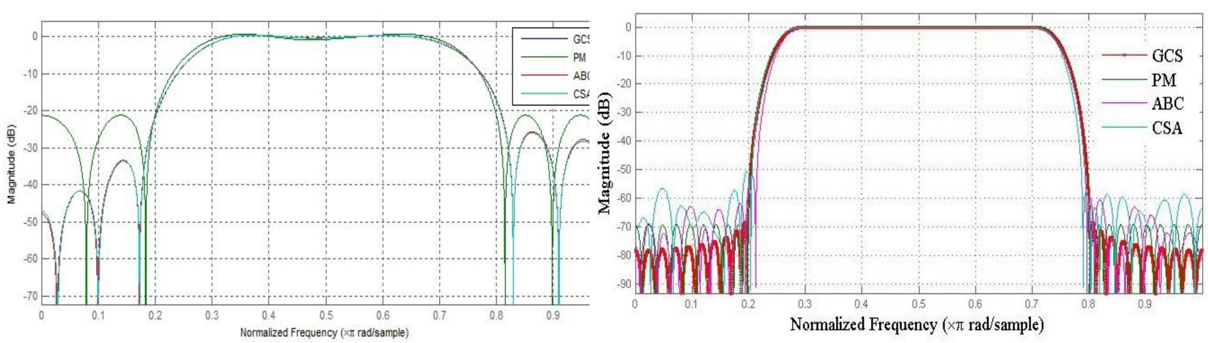
Type 1 LPF					Type 2 LPF				
N (Order)	Method	A_s (dB)	R_p (dB)	Time(s)	N (Order)	Method	A_s (dB)	R_p (dB)	Time(s)
18	PM	22.6279	1.3524	<1	19	PM	22.9883	1.1122	<1
	ABC	22.0812	3.7832	9.9015		ABC	17.1627	1.2020	12.3446
	CSA	23.4992	3.8896	14.6029		CSA	24.1423	**0.8025**	7.5858
	GCS	**24.9802**	**0.9334**	**5.567**		GCS	**25.3609**	0.8430	**4.8547**
38	PM	36.7002	0.2115	<1	39	PM	36.8820	0.2064	<1
	ABC	37.9578	0.2765	12.3884		ABC	38.9414	0.28	13.3019
	CSA	38.1934	0.2841	17.6197		CSA	39.7616	0.1693	**7.9674**
	GCS	**38.4958**	**0.1478**	**7.41**		GCS	**40.3977**	**0.1673**	9.4642
58	PM	49.7258	0.0386	<1	59	PM	50.8277	0.0336	<1
	ABC	**51.9373**	0.0381	18.8036		ABC	51.9954	0.0278	21.5153
	CSA	50.0474	**0.0342**	29.5655		CSA	**52.7999**	0.0333	31.2383
	GCS	51.7479	0.0355	**11.8001**		GCS	52.5928	**0.0316**	**19.3018**
78	PM	64.0842	0.0058	<1	79	PM	64.1538	0.0058	<1
	ABC	**93.1159**	0.0052	54.9771		ABC	**82.5854**	0.0058	56.6412

	CSA	92.3886	0.0055	**20.2133**		CSA	69.2722	**0.0052**	76.8015
	GCS	67.6349	**0.0049**	24.1975		GCS	68.8032	**0.0052**	**12.4352**
98	PM	75.1599	0.0013	<1	99	PM	76.0458	0.0011	<1
	ABC	78.2015	0.0011	678.815		ABC	78.0891	4.90E-41	583.4213
	CSA	**79.1589**	0.0011	34.3266		CSA	77.7867	4.64E-04	112.8933
	GCS	77.289	**7.23E-04**	**16.4009**		GCS	**79.9305**	**7.82E-04**	**18.9548**

From Table 3, it is clearly seen that although in some orders ABC provides better A_s , but it does not provide better R_p at the same time. In case of 19th order, Type 1 LPF , GCS provides 24.9802 dB A_s and 0.9334 dB R_p in 5.567s; Similarly it is evident that for Type 2 ,60th ordered LPF , CSA provides 52.799 dB, 0.0333dB and, 31.2383s A_s, R_p and Time respectively; but GCS shows 52.5928 dB, 0.0316 dB , and 19.3018s A_s, R_p and Time respectively. Absolute Parameter independence, as well as potential searching strategy, are the major reason behind this better result.

4.2.2 Band-pass FIR filter:

The design specification for band-pass filter is: first and second stop-band frequency $\omega_{s1} = 0.2\times \text{pi}$, $\omega_{s2} = 0.8\times \text{pi}$; first and second stop-band frequency are: $\omega_{p1} = 0.3\times \text{pi}$ $\omega_{p2} = 0.7\times \text{pi}$. The estimated design of both Type of band-pass filters has been discussed in Eq. (17-20) .Based on these equations, the algorithms are simulated and the graphical comparison of the magnitude response for Type 1, 78th order band-pass FIR filter and Type 2, 19th order band-pass filter has been shown in Fig.3.



(a) (b)

Fig.3. *Output of Type 1 and Type 2 low-pass FIR filter*

(a) 19th order Type 2, band-pass FIR filter; (b) 78th order Type 1, band-pass FIR filter

It is very much evident from the above figure that GCS provides better transition than PM, ABC, and CSA. Again the ripple obtained is less than that of other methods. For Type 1 BPF the observation is same, which proves the effectiveness of GCS. The detailed information of obtained R_p, A_s and execution time has been listed in Table 4. In case of FIR Band-pass filter, GCS conquers over ABC, CSA and PM by showing better A_s and R_p in less time. For example, in case of order 19 of Type 1 BPF, GCS provides 32.9009 dB, 1.3991 dB A_s and R_p respectively in 0.6892s. But ABC provides 30.9010 dB, 1.5991 dB A_s and R_p in 2.5548s and CSA provides 32.9009 dB, 1.58481 dB A_s and R_p in 3.8431s. Similarly, it can be seen that for Type 2 Band-pass filter, GCS shows 79.5849 dB, 0.0056 dB A_s and R_p respectively in 7.4898s time, which is more desirable than the output acquired both from ABC and CSA. So the outline is, ABC does not provide a satisfactory decrease in the ripple for higher order filter, even though CSA provides comparable results with PM, the computational time required for CSA is more than that of GCS. So, GCS triumphs over ABC and CSA while taking into consideration all three aspects, i.e. stop-band attenuation, ripple as well as time.

Table 4 Simulation results for Type 1 and Type 2 BPF

Type 1 BPF					Type 2 BPF				
N (Order)	Method	A_s (dB)	R_P(dB)	Time(s)	N (Order)	Method	A_s (dB)	R_P(dB)	Time(s)
18	PM	19.8713	1.9706	<1	19	PM	21.8928	1.5205	<1
	ABC	31.9010	1.5991	2.5548		ABC	26.4243	1.1307	2.5575
	CSA	32.9009	1.5848	3.8431		CSA	26.496	0.8647	4.3027
	GCS	**32.9009**	**1.3991**	**0.6892**		GCS	**26.985**	**0.821**	**0.5263**
38	PM	38.9794	0.1978	<1	39	PM	38.481	0.2103	<1
	ABC	44.0116	0.1516	2.7546		ABC	46.6588	0.2006	2.5986
	CSA	44.0319	0.1589	4.0012		CSA	49.0344	0.2103	4.1078
	GCS	**44.0322**	**0.1435**	**1.676**		GCS	**49.312**	**0.1887**	**0.4831**
58	PM	51.3191	0.0474	<1	59	PM	53.2443	0.0379	<1
	ABC	53.8355	0.0451	20.6491		ABC	55.5501	0.0292	15.9255
	CSA	54.7316	0.0413	16.4739		CSA	55.3894	0.0327	9.5344
	GCS	**54.9714**	**0.0423**	**6.2784**		GCS	**56.0025**	**0.0314**	**3.2341**
78	PM	69.3937	0.0059	<1	79	PM	68.9088	0.0062	<1
	ABC	69.1717	0.0156	66.3259		ABC	69.2717	0.0105	57.0476
	CSA	69.4061	0.0129	34.4433		CSA	78.4772	0.0069	18.4513
	GCS	**71.9444**	**0.0057**	**14.0418**		GCS	**79.5849**	**0.0056**	**7.4898**
98	PM	81.3022	0.0015	<1	99	PM	83.1026	0.0012	<1
	ABC	81.3184	0.0034	700.0178		ABC	83.0088	0.0088	327.5281

	CSA	86.7188	0.0015	97.2907		CSA	85.0213	0.0013	49.0759
	GCS	**86.795**	**0.0013**	**51.8108**		GCS	**85.8723**	**0.0011**	**20.0936**

4.3 Statistical Analysis of GCS

Statistical Analysis has been done in Fig.4, which depicts the percentage improvement of GCS with respect to ABC and CSA.

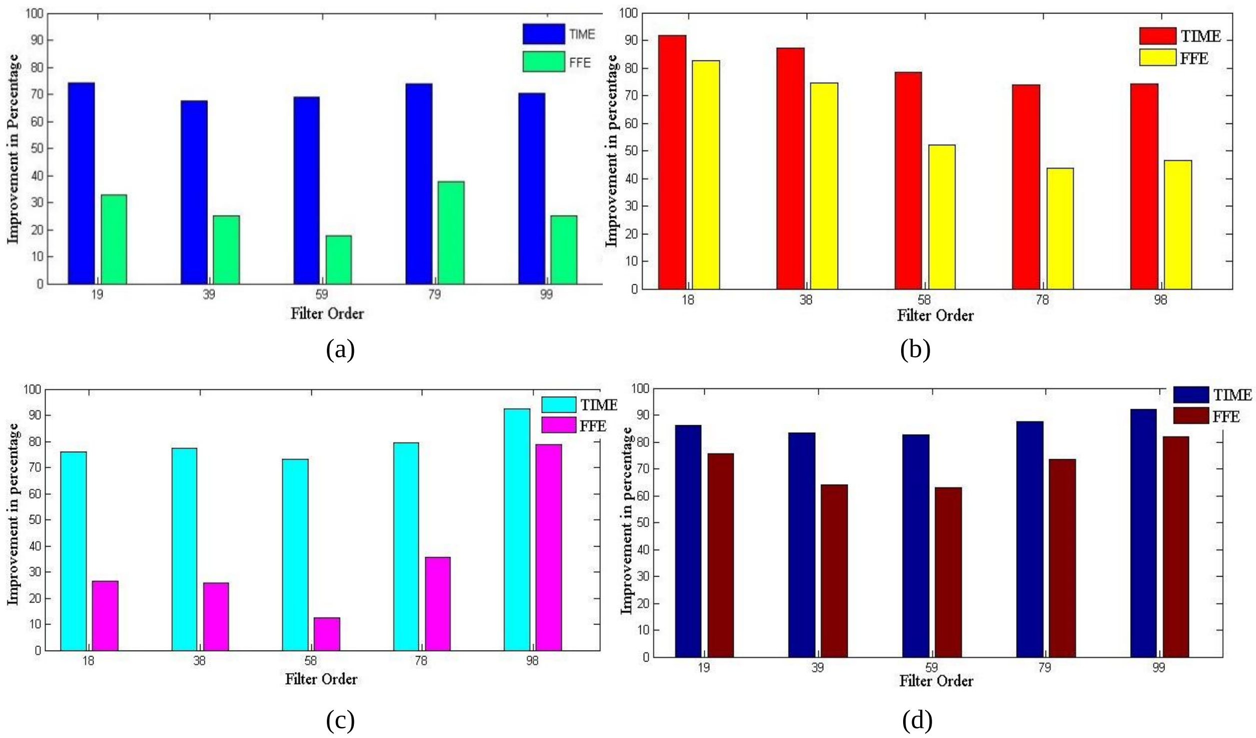


Fig.4. *Percentage improvement of GCS with Time and FFE*

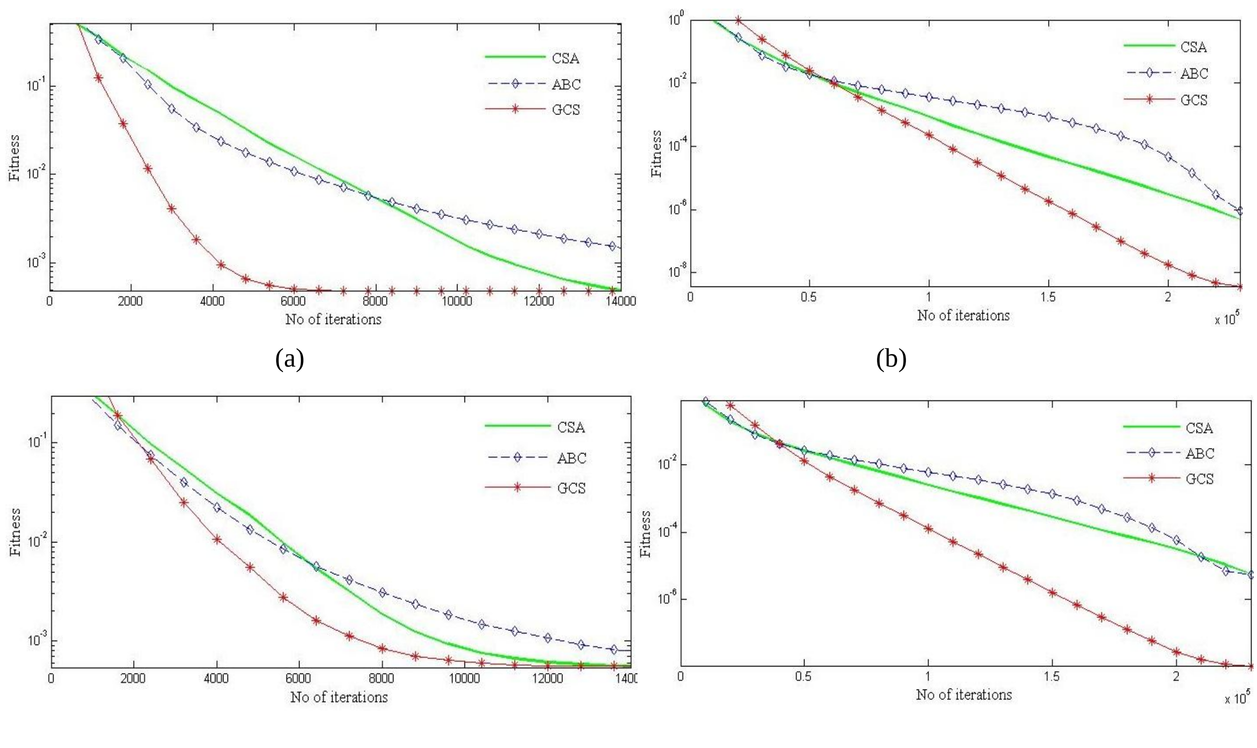
(a) Type 1 LPF with respect to CSA, (b) Type 2 LPF with respect to ABC, (c) Type 1 BPF with respect to ABC, and (d) Type 2 BPF with respect to CSA

In order to compare the algorithms in terms of time required for convergence (TIME) i.e. time required to reach minimum fitness value and the number of cycles or iterations required by an algorithm to converge (FFE), the code has been simulated 50 times for different orders of the filter i.e.18, 38,58,78,98 for Type 1 LPF and BPF; 19, 39,59,79,99 for Type 2 LPF and BPF. GCS shows 75.88%, 77.21% 73.08% 79.50% 92.63% improvement in time and 26.34%, 25.93% ,12.35% .35.69%, 78.81% improvement in FFE compared to ABC for 18,38,58,78,98 order BPF respectively. Similarly, GCS shows 74.11%, 67.65%, 69.07%, 73.91%, 70.28% improvement in time and 32.81%, 25.10%, 17.67%, 37.64%, 25.17% improvement in FFE compared to CSA for 19,39,59,79,99 order BPF respectively.

For Low-pass filters GCS depicts 91.11%, 87.26%, 78.47%, 73.68%, 74.34% upgrade in time; 82.66%, 74.53%, 51.99%, 43.73%, 46.66% betterment in FFE compared to CSA for 18,38,58,78,98 order LPF and 86.23% 83.49%, 82.52%, 87.53%, 91.97% recovery in time; 75.56%, 64.06%, 63.14%, 73.40%, 81.90% betterment in FFE compared to ABC for 19,39,59,79,99 order LPF. Thus GCS takes less time for execution and FFE than its contemporary optimization algorithms CSA and ABC.

4.4 Convergence profile analysis

An algorithm converges when it reaches its minimum magnitude error fitness value and higher the rate of convergence of an algorithm more is its effectiveness. The parameters that affect the convergence of an algorithm are: order complexity, number of iteration and the stopping criterion defined for a specific algorithm. The fruitfulness of an algorithm can be determined from its convergence behavior. We can find the minimum error fitness value and the maximum number of iterations required to reach this value from the graphical examination of the convergence behavior. The potency of an algorithm increases with the decrease in the number of iterations required for an algorithm to reach its minimum fitness value.



(a) (b) (c) (d)

Fig.5. *Convergence profile analysis of GCS , CSA and ABC*
(a) Type 1-18th order FIR Low-pass filter, (b) Type 2-79th order FIR Low-pass filter, (c) Type 2-19th order FIR Band-pass filter, (d) Type 1-78th order FIR Band-pass filter.

The convergence profiles for GCS, CSA, and ABC have been shown in Fig.5. It is evident that the convergence of GCS is faster compared to CSA and ABC with minimum fitness error value in all the cases. In the case of Type 1-18th order and Type 2-79th order low-pass filter; GCS converges after 6500 and 2.3x10^5 iterations respectively while ABC and CSA cannot reach an early convergence in both cases. In the case of Type 2-19th order, band-pass filter, GCS converges after 1.2x10^3 iterations whereas CSA takes 1.3x10^4 iteration and ABC takes even more than 1.4x10^4 iterations to converge that also with a much higher error fitness value. Similarly, for 78th order band-pass filter, GCS converges after 2.3x10^5 iterations, whereas CSA and ABC remain far beyond convergence at 2.3x10^5 iteration. Again after going through all the plots above we can infer that GCS reaches to minimum error fitness value compared to CSA and ABC which yields higher error fitness value.

5. Conclusion

In this paper, an improved approach of cuckoo search algorithm or GCS has been used to obtain the optimal set of coefficients for FIR filter. All the results and graphical analysis prove that GCS has greater potential compared to other optimization algorithms like CSA and ABC. It not only provides faster convergence but also gives a very good balance between A_s and R_p, for both higher and lower order BPF and LPF. The improvement in the stop-band attenuation and pass-band ripple when GCS is used is reflected in Table 3 and Table 4. Again, higher convergence rate and significantly smaller error fitness value is achieved for GCS compared to other optimization algorithms for all the orders of the filter. Therefore, the time required to get the optimal solution is also very less. Parameter independency of GCS ensures that no tuning of parameters is required to design a filter and to achieve the optimal set of filter coefficients, at the same time GCS is simpler to use than ABC and CSA. GCS is proved to be durable, and a universal algorithm where the convergence becomes faster under the influence of Gbest value. Even while designing higher order filters, i.e. when the complexity of the problem increases, GCS provides praiseworthy results and skillfully vanquish the flaws or limitations which are present in other optimization algorithms.

6. References

[1] Aggarwal, A., Rawat,T.K., Upadhyay, D.K.: ‘Design of optimal digital FIR filters using evolutionary and swarm optimization techniques’. Int J Electron Commun(AEÜ),2015. OK

[2] Kavsaoglu ,Resit, ‘A windows-based digital filter design’, Math. Comput. Appl.,2003, **8**,(3) ,pp. 287–294. OK

[3] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, “GSA: A Gravitational Search Algorithm”, In Elsevier, Information Sciences, vol. 179, no. 13, 2009, pp. 2232–2248.

[4] E. Rashedi, H. Nezamabadi-pour, S. Saryazdi, “Filter modeling using gravitational search algorithm”, In Elsevier, Engineering Applications of Artificial Intelligence, vol. 24, no. 1, 2011, pp. 117-122.

[5] T. Chakraborti, K. Das Sharma, A. Chatterjee, "A novel local extrema based gravitational search algorithm and its application in face recognition using one training image per class", In Elsevier, Engineering Applications of Artificial Intelligence, vol. 34, 2014, pp. 13-22.

[6] D. Karaboga, B. Basturk, “Artificial Bee Colony (ABC) Optimization Algorithm for Solving Constrained Optimization Problems”, P. Melin et al. (Eds.): IFSA 2007, LNAI 4529, pp. 789–798, 2007. Springer-Verlag Berlin Heidelberg 2007.

[7] W. Gao, S. Liu, L. Huang, “A global best artificial bee colony algorithm for global optimization”, Journal of Computational and Applied Mathematics,vol. 236, no.11, 2012, pp. 2741–2753.

[8] D. Karaboga, B. Akay, “A comparative study of Artificial Bee Colony algorithm”, Applied Mathematics and Computation,vol. 214, no. 1, 2009, pp. 108–132.

[9] X.S. Yang, S. Deb, “Cuckoo Search via Lévy Flights”, Proceedings of IEEE World Conference on Nature and Biologically Inspired Computing, 2009, pp. 210–214..

[10] X.S. Yang, S. Deb, “Engineering Optimization by Cuckoo Search”, Int. J. Mathematical Modelling and Numerical Optimization Proceedings, vol. 1, no. 4 ,2010, pp. 330–343.

[11] M.Kumar, T.K.Rawat, “Optimal fractional delay-IIR filter design using cuckoo search algorithm” , ISA Trans 59(2015) 39-54.

[12] P. Barthelemy, J. Bertolotty, D.S. Wiersma, “A Lévy flight for light”, Proceedings of nature Publishing Group, vol.453, no. 4, ,2008, pp. 495–498.

[13] C.T. Brown, L.S. Liebovitch, R. Glendon, “Lévy flights in Dobe Ju/hoansi foraging patterns”, Human Ecol. vol. 35, no. 1, ,2007,pp 129-138.

[14] S. Walten, O. Hassan, K. Morgan, M. R. Brown, “Modified Cuckoo Search: A new Gradient Free Optimization Algorithm”, Chaos, Solitons and Fractals, vol. 44, no. 9, pp. 710-718.

[15] A. Kaveh, T. Bakshpoori, “Optimum Design of Space Trusses using Cuckoo Search algorithm with Lévy Flights”, IJST Transactions of Civil Engineering, vol. 37, no. C1, pp. 1-15.

[16] A. R. Yildiz, “Cuckoo Search Algorithm for the Selection of Optimal Machining Parameters in Milling Operations”, The International Journal of Advanced Manufacturing Technology, vol. 64, no. 1-4, pp. 55-61.

[17] N.E. Mastorakis, I.F. Gonos, M.N.S. Swamy, “Design of two dimensional recursive filters using genetic algorithms”, IEEE Trans. Circ. Syst. I – Fundam. Theory Appl.vol. 50, no. 5, 2003, pp. 634–639.

[18] N. Benvenuto, M. Marchesi, A. Uncini, “Applications of simulated annealing for the design of special digital filters”, IEEE Transactions on Signal Processing, vol. 40, no. 2, 1992, pp. 323-332.

[19] J.M.P. Langlois, “Design of linear phase FIR filters using particle swarm optimization”, Proceedings of 22nd Biennial Symposium on Communications, Kingston, Ontario, Canada, 2004.

[20] J. Sun, W.B. Xu, B. Feng, “A global search strategy of quantum-behaved particle swarm optimization”, Cybernetics and Intelligent Systems Proceedings. 2004, pp. 111–116.

[21] A. Banks, J. Vincent, C. Anyakoha, “A review of particle swarm optimization. Part II: hybridization, combinatorial, multicriteria and constrained optimization, and indicative applications”,Natural Computing, vol. 7, no. 1, 2008, pp. 109-124.

[22] S. Dhabal , S. Sengupta, "Efficient design of high pass FIR filter using quantum-behaved particle swarm optimization with weighted mean best position," Computer, Communication, Control and Information Technology (C3IT),Third International Conference on, Hooghly, 2015,pp.1-6.

[23] A. Sarangi, R.K. Mahapatra, S.P. Panigrahi, “DEPSO and PSO-QI in digital filter design.”, Expert Syst. Appl., vol. 38, no. 9, 2011, pp. 10966–10973.

[24] V.K.Ingle, J.G.Proakis, “Digital signal Processing using MATLAB” , Chapter 7. “FIR FILTER DESIGN” pp. 228-234.