A SOFT COMPUTING BASED APPROACH FOR SIGNAL PROCESSING

Thesis Submitted by

Poulami Das

Doctor of Philosophy (Engineering)

Department of Computer Science & Engineering
Faculty Council of Engineering & Technology
Jadavpur University
Kolkata, India
2018

JADAVPUR UNIVERSITY

KOLKATA-700032, INDIA

INDEX NO. 166/15/E

1. Title of the Thesis:

A SOFT COMPUTING BASED APPROACH FOR SIGNAL PROCESSING

2. Name, Designation & Institution of the Supervisor/s:

Dr. Sudip Kumar Naskar

Assistant Professor

Department of Computer Science and Engineering, Jadavpur University,

Kolkata – 700 032, India

&

Dr. Sankar Narayan Patra

Associate Professor

Department of Instrumentation and Electronics Engineering, Jadavpur University,

Kolkata - 700 032, India

3. List of Publication:

(A) Journal Publications (3):

i. **Poulami Das**, Sudip Kumar Naskar and Sankar Narayan Patra. 2019. Fast Converging Cuckoo Search Algorithm to design symmetric FIR filters, *International Journal of Computers and Applications*, DOI: 10.1080/1206212X.2019.1588512, Taylor and Francis.

ii. **Poulami Das**, Sudip Kumar Naskar and Sankar Narayan Patra. 2018. Global Best Steered Quantum Inspired Cuckoo Search Algorithm. *Applied Soft Computing*, Volume 71, October 2018 pp. 1-19. (IF: 3.907), Elsevier. (Cited by 1)

iii. **Poulami Das**, Subhas Chandra Panja, Sudip Kumar Naskar and Sankar Narayan Patra**.** 2016. An Approach for obtaining least noisy signal using Kaiser Window &

Genetic Algorithm. *International Journal of Computer Applications,* (0975 – 8887) Volume 150, No.6, pp. 16- 21, September 2016.

(B) Conference / Workshop / Symposium Publications (3):

i. **Poulami Das**, Sudip Kumar Naskar and Sankar Narayan Patra. 2017. Adaptive Global Best Steered Cuckoo Search Algorithm for FIR Filter Design. *International Conference on Research in Computational Intelligence and Communication Networks (*ICRCICN *2017)*, 3-5 November, 2017, IEEE.

ii. **Poulami Das**, Sourav Samanta, Sudip Kumar Naskar and Sankar Narayan Patra. 2016. An approach to optimize FIR filter coefficients using GA, PSO & BAT Algorithm and their Comparative Analysis. *International Conference on Computer, Electrical & Communication Engineering (ICCECE 2016)*, 16-17 December, 2016, IEEE.

iii. **Poulami Das**, Sudip Kumar Naskar and Sankar Narayan Patra. 2016. An Approach to Enhance the Performance of Kaiser Window Based Filter. *International Conference on Research in Computational Intelligence and Communication Networks (*ICRCICN *2016)*, 23-25 September, 2016, IEEE.

4. List of Patents: Nil

5. List of Presentations in National / International / Conferences/Workshops/Symposiums:

i. **Poulami Das**, Sudip Kumar Naskar and Sankar Narayan Patra. 2017. Adaptive Global Best Steered Cuckoo Search Algorithm for FIR Filter Design. *International Conference on Research in Computational Intelligence and Communication Networks (*ICRCICN *2017),* 3-5 November, 2017, IEEE. (Oral)

ii. **Poulami Das**, Sourav Samanta, Sudip Kumar Naskar and Sankar Narayan Patra. 2016. An approach to optimize FIR filter coefficients using GA, PSO & BAT Algorithm and their Comparative Analysis. *International Conference on Computer, Electrical & Communication Engineering (ICCECE 2016)*, 16-17 December, 2016, IEEE. (Oral)

iii. **Poulami Das**, Sudip Kumar Naskar and Sankar Narayan Patra. 2016. An Approach to Enhance the Performance of Kaiser Window Based Filter. *International Conference* *on Research in Computational Intelligence and Communication Networks (*ICRCICN *2016)*, 23-25 September, 2016, IEEE. (Oral)

iv. **Poulami Das**, Subhas Chandra Panja, Sudip Kumar Naskar and Sankar Narayan Patra. 2015. A new approach based on Genetic Algorithm for de-noising a signal. *National Symposium on Recent Trends in Instrumentation Science and Technology,* Jadavpur University, 19-21 March, 2015. Published in Journal IISER. (Oral)

Certificate from the Supervisors

This is to certify that the thesis entitled "A SOFT COMPUTING BASED APPROACH FOR SIGNAL PROCESSING" submitted by Miss. Poulami Das, who got her name registered on November 9th, 2015, for the award of Ph.D (Engg.) degree of Jadavpur University is absolutely based upon her own work under the supervision of Dr. Sudip Kumar Naskar and Dr. Sankar Narayan Patra and that neither her thesis nor any part of the thesis has been submitted for any degree or any other academic award anywhere before.

Signature of the Supervisors

Dr. Sankar Narayan Patra
Associate Professor
Department of Instrumentation and Electronics Engineering,
Jadavpur University,
Kolkata - 700 032, India

Dr. Sudip Kumar Naskar
Assistant Professor
Department of Computer Science and Engineering,
Jadavpur University,
Kolkata – 700 032, India

Declaration

I declare that the work described in this thesis is entirely my own. No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or institute. Any help or source information, which has been availed in the thesis, has been duly acknowledged.

Signature:

____

Poulami Das

Dedicated to

My Parents

Acknowledgements

Research has been defined as the systematized effort to gain new knowledge. This journey to new insights becomes easier when one receives proper direction and encouragement. During my journey of PhD work, there were many ups and downs. I would like to express my sincere gratitude to those people who helped me to overcome all the hurdles throughout my PhD period.

First and foremost, I would like to gratefully acknowledge my PhD supervisors, Dr. Sudip Kumar Naskar and Dr. Sankar Narayan Patra for their continuous support and guidance. Without their support it was not possible to complete this thesis. Respected Dr. Sankar Narayan Patra Sir provided me all the necessary facilities and infrastructure to carry out my research work. I remember his motivating words during the struggling period of research. Not only that, his friendly attitude and amazing environment in his Astronomical Instruments Design (AID) Laboratory made the time happy and enjoyable. It is not an exaggeration to say that my curiosity and eagerness in research was evoked by respected Dr. Sudip Kumar Naskar Sir. I would like to thank him for inspiring me to pursue excellence. The discussions with him stimulated to explore new ideas. His helps to write the papers helped me most to shape the research papers as well as my PhD thesis. The importance of supervision is well-known to anyone who conducts research. In this context, I would wholeheartedly thank my supervisor and co-supervisor for their unconditional support. I will also like to thank Prof. Subhash Chandra Panja, who has been an absolute kind heart and never refused my request for any kind of helps. His research ideas and motivations helped me to shape my research. This list would be incomplete without Dr. Nilanjan Dey for helping me to visualize the word PhD so clearly. I must thank him because as a mentor of my Master's project he was kind enough to let me know the details of the research. It completely changed my outlook towards Research.

Next, I want to express gratitude to my parents for standing by me always and keeping faith on me. It can't be expressed in words what my parents did for me and still doing for me. I must thank my parents for helping me in every step of my life and pampering me so much.

Next, I like to thank my all-purpose guide and my husband Avishek for his continuous and unconditional support during my research work. I must say that Avishek is the person for whom

I could dare to pursue PhD. I used to hear most inspiring words from him during the period of failures in research. Without his support it won't be possible for me to complete the thesis.

Another person I want to thank is 'Didan' (my grand-mother) for encouraging me to follow her profession of teaching. After my parents 'Didan' is the next who will be most happy with my success.

I like to thank Jeevan Rekha Diagnostic Pvt. Ltd., Uttar Dinajpur, West Bengal, India for providing necessary biomedical data for carrying out this research work.

Signature:

____

Poulami Das

Abstract

In the present mechanized world, digitalization has encroached almost every sphere of technology like biomedical information processing, defense applications, astrophysical data analysis, just to name a few. Biomedical information comprises of one dimensional or two dimensional signals which are received as output of different biomedical instruments. Defence technologies involve use of radar signals received as outputs of sensor devices. Watermarking is another application of digitalization in defense technologies that provides security to the secret information. Visible watermarking is also very much useful in the department of law to prove intellectual property rights. Astrophysical instruments acquire information generally in time series that is basically one dimensional series of data points indexed in order of time. The current globalized era is marked by a rapid increase in the use of wireless media to exchange information over globally distributed locations. This advancement and growth of technologically mediated information help provide medical care remotely by exchanging biomedical signals amongst various hospitals and diagnostic centers across the globe. It also helps in the area of defense by sharing the radar signals instantly via phone or internet. However, while transmitting, these signals may get affected by some unwanted components termed as noise which are adverse but inevitable. Removal of such unwanted components from the signals has remained challenging for the researchers since the earliest days of signal processing. For removing noise from signals, the use of digital filters has been proved to be more effective than the analog filters due to the flexibility of hardware efficiency provided by the digital filters. Among the two types of digital filters, Finite Impulse Response (FIR) filters are used more extensively compared to the Infinite Impulse Response (IIR) filters because of FIR filters' outstanding characteristic of stability and the capability of obtaining linear phase response. FIR filters take a discrete time signal as input and performs addition and multiplication operations to obtain the desired filtered discrete time output signal. Among various techniques proposed by several researchers to design FIR filters, the use of window functions is the most popular approach. For implementation of FIR filters, several soft computing approaches have also been proved to be effective. Limitations of conventional approaches such as window methods, frequency sampling methods used for filter design motivated us to use soft computing techniques to design FIR filters.

In this thesis, an innovative combined approach has been proposed to de-noise biomedical signals using the most effective window function Kaiser Window and Genetic Algorithm, well known evolutionary approach. Kaiser Window with varying passband and stopband ripples is initially used for filtration of noisy heart sound signals. Genetic Algorithm is then used to obtain the least noisy signal. Optimization of parameters used to design a digital filter using Kaiser Window function is also performed in this thesis using an adaptive Ant Weight Lifting Algorithm.

Another approach for FIR filter design is to use optimized set of coefficients. In this thesis, we also make a comparative study of a few traditional algorithms in optimizing filter coefficients. A novel algorithm, namely *Global Best Steered Cuckoo Search Algorithm,* is also proposed for the same purpose. This new algorithm is proved to be much more efficient in filter design compared to the traditional algorithms in terms of passband ripple and stopband attenuation of the filters.

Nowadays, many battery operated devices such as mobile phones, hearing aids, FM radios, etc. also use FIR filters. Due to the power starving nature of these devices, implementation of FIR filters with as low power as possible is of utmost necessity. Therefore, aiming to address the need for perpetually demanding high speed and low power devices, innovative techniques for implementing hardware efficient FIR filter are also proposed in this thesis. These techniques involve the use of fixed length coefficients and two innovative algorithms are proposed to obtain optimized coefficients. A new quantum algorithm, namely *Global Best Steered Quantum Inspired Cuckoo Search Algorithm*, is proposed in this thesis for obtaining optimized filter coefficients capable of acquiring desired filter responses. This algorithm is also proved to be effective in reducing the number of SPT terms. With the help of common sub-expression elimination technique, required number of adders is further minimized for hardware efficient filter implementation. We also proposed another new algorithm, namely *Fast Converging Flower Pollination Algorithm*, for the same purpose. These new algorithms have been proved to be much more effective compared to their traditional versions which have also been proved to be efficient in the domain of filter coefficient optimization compared to the conventional approaches like window methods and frequency sampling methods.

Title of the Thesis

A SOFT COMPUTING BASED APPROACH FOR SIGNAL PROCESSING

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1. History and Evolution of Intelligent Signal Processing

"Artificial Intelligence is whatever hasn't been done yet"

- Larry Tesler

"Signal Processing is more than its beloved name"

- Rabab Ward

Signal processing has significant applications in many spheres of technologies like biomedical instrumentation, astrophysical instrumentation, sensor development, defense applications, weather forecasting, broadcasting in television and many more. Basically signal processing involves operation on signals with the aim of retrieving expedient information from the signals. For processing signals, processors or systems are required similar to a human seeing something and using his/her visual pathways to retrieve and process information in the brain about the observed scenario. In case of human processing of visual information, the processor is biological in nature. Signal processors may be a sensor, an electronic system, a mechanical system or even a computer program. All these processors are capable of extracting information from different one-dimensional signals like video and audio signals, biomedical signals (heart sound, Electrocardiogram, Electroencephalogram, Electrooculogram, Intravascular ultrasound, etc.), astrophysical signals (solar radio flux, solar radiation, etc.), time series (daily temperature, annual rainfall of a place, etc.), seismic signals (movement of rocks due to earthquake, volcanic eruption, etc.) as well as multi-dimensional signals like different images. Processing of signals involves the following operations.

a. Simple Time-Domain Operations
b. Filtering
c. Generation of Complex-Valued Signals
d. Amplitude Modulation
e. Multiplexing and De-multiplexing
f. Quadrature Amplitude Modulation

All these operations can be performed cleverly using intelligent algorithms that are implemented using the concept of Artificial Intelligence (AI). Key notion behind the AI is making machines smart like human beings. Rapid advancements in AI in the last few decades also caused tremendous improvements in signal processing.

Filtering is the most important as well as a very challenging task in signal processing. For filtering signals, or in other words to remove noise from signals, several intelligent techniques have been proved to be effective. In 1969, Gold and Jordan (1969) and in the next year Rabiner et al. (1970) used frequency sampling method for filter design which involves computation of filter coefficients by sampling the ideal filter in the frequency domain. In this method a specific frequency can be approximated by fixing most of the Discrete Fourier Transform coefficients and rejecting the unspecified coefficients that lie in the range of transition bands. These unspecified coefficients are optimized with an aim of minimizing a weighted approximation error over the desired frequency range. Optimization was carried out using an intelligent direct search approach. In case of optimal approaches, filter coefficients are obtained by minimizing the maximum error between the desired and the actual frequency response using different optimization techniques. In the last two decades, invention of wide number of efficient soft computing techniques led to outstanding improvements in filter implementation approaches.

1.2. General Concepts of Analog and Digital Signal Processing

Signals can be classified into two types: analog and digital. Analog signals have an infinite number of values in a range, whereas digital signals can have only a limited number of values. Analog signal processing is based on the capability of analog systems to solve differential equations describing physical systems. It uses analog circuit elements like resistors, capacitors, transistors, diodes, etc. Analog signal processing was governing the last century until digital computers and microprocessors were started being used widely. Digital signal processing involves numerical calculations. Two basic advantages of the digital approach over the analog approach for signal processing are flexibility and repeatability. Flexibility refer to the fact that the same hardware can be used for different kind of digital signal processing operations, while in core analog signal processing a system has to be designed for each operation. Repeatability in digital signal processing means that the same signal processing operation can be repeated several times obtaining the same results, whereas in analog systems, parameter variation may occur due to change in temperature or voltage.

1.3. Digital Filters for Signal Noise Removal

In the present digital age of electronic appliances, dealing with signals has become a part and parcel of everyday modern life. During transmission via any media signals get affected by unwanted components; which is adverse but inevitable. Elimination of such unwanted components, termed as *noise*, from transmitted signals proved to be an important as well as puzzling task for the researchers from the initial days of Signal Processing. Among a significant number of techniques proposed for removal of noise from signals, the use of digital filters has become most effectual in multiple ways. Digital filters have much less design complexity and cost compared to the analog ones. Design of digital filters basically involves obtaining a perfect set of coefficients using programmable optimization algorithms. Another reason for the digital filters being more expedient for signal de-noising is the presence of high latency. Digital filters add noise caused by quantization but the noise is least effective, whereas analog filters add highly effective components based thermal noise. Moreover, analog filters have accuracy limitations due to component tolerances and undesired nonlinearities which make the digital filters more useful.

1.4. Digital FIR Filter Design

Digital filters (Antoniou, 1993; Smith, 2011; Sharma, 2009; Mitra, 2013) have extensive use compared to analog filters due to much lesser design overhead. Lower hardware cost and amazing behavior of altering characteristics with changes in the discrete values stored in the registers have made the digital filters more effectual than the analog ones. Digital filters can be classified into two types - Finite Impulse Response (FIR) Filter (Sharma, 2009; Mitra, 2013; Proakis & Manolakis, 2015), and Infinite Impulse Response (IIR) Filter (Sharma, 2009; Mitra, 2013). In comparison to the IIR filters, FIR filters are more effective in digital audio and video signal processing due to the outstanding characteristics of stability and the capability of obtaining linear phase response. Output of FIR filters depend on the current and past input samples which can be realized non-recursively. Quantization in finite word-length does not generate much erroneous response which is another advantage of FIR filters over the IIR filters.

Filter implementation involves determination of the coefficients that nearly approximate the desired filter characteristics. These desired characteristics are specified in the frequency domain in terms of the desired magnitude and phase responses of the filters. For the ideal FIR filters, responses can be defined as follows.

$$H_d\left(e^{j\omega}\right) = 1 \, for \; 0 < \omega \leq \omega_{cutoff}$$
$$= 0 \, for \; \omega > \omega_{cutoff} \quad (1.1)$$

This means in case of ideal filters, frequency response must be equal to 1 in the pass band, and equal to 0 in the stop band.

Based on the filter order and symmetricity of the filter coefficients, filters are categorized into four types.

i. Type I- Even order and symmetric coefficients,

ii. Type II- Odd order and symmetric coefficients,

iii. Type III- Even order and asymmetric coefficients,

iv. Type IV- Odd order and asymmetric coefficients.

For the symmetric FIR filters, number of filter coefficients (d) is computed as follows.

$d = \frac{M+1}{2}$; when M is odd, $d = \frac{M}{2}$; when M is even (1.2)

1.4.1. Conventional Approaches for Digital FIR Filter Design

The foremost step that the design of FIR filters involves is computation of filter coefficients. Among a considerable number of techniques proposed for finding the filter coefficients, Window method (Sharma, 2009), frequency sampling and optimal algorithm are the most efficiently used. The most commonly used fixed window functions are Rectangular, Hanning, Hamming, Blackman and Bartlett (Salivahanan, Vallavaraj, & Gnanapriya, 2007), (Sharma, 2009). In these window functions, values of passband ripple δ_p and stopband ripple δ_s are specific as well as same. Henceforth, result will show either too small pass band ripple or too large stop band attenuation. In variable window such as Kaiser (Sharma, 2009), the values of δ_p and δ_s are chosen by using the ripple control parameter specified by the Designer. Application of the window function also has two effects on the amplitude response of the filter. First, the amplitudes of Gibbs' oscillations (Chakravarti & Mehra, 2014) in the passbands and stopbands are directly related to the ripple ratio of the window. Second, transition bands are introduced between passbands and stopbands whose width is directly related to the main-lobe width of the window. In case of frequency sampling approach instead of having very few independent variables, this method lacked behind due to the presence of a large number of constraints and as well as fixed control over the band edge frequencies or the passband ripples. In case of optimal approaches, filter coefficients are obtained by minimizing the maximum error between the desired and actual frequency response using different optimization algorithms. An optimal approach for FIR filter design using Chebyshev sense was first developed by Herrmann (Herrmann, 1970). In this method frequency response of the optimal low pass filter was assumed to be equiripple in both the pass band and stop band. The number of ripples in each band was made fixed, a set of non-linear equations describing the filter were also developed using an iterative descent method. Only disadvantage of this technique is restriction in filter length of about 40. This limitation of Herrmann's approach was rectified by Hofstetter et al. (Hofstetter, Oppenheim, & Siegel, 1971), by developing an algorithm, called as "leminiscent of the Remez exchange algorithm". But both of these methods result in extra ripple or maximum ripple filters, which are restricted subsets of optimal minimax filters. These methods also have the shortcoming of not accommodating the pass band and stop band cut-off frequencies of the filters. Another efficient optimal approach for designing non-recursive digital filter was designed in 1972 using Chebyshev approximation (Parks & McClellan, 1972). This technique comprised multi pass band - stop band filters, differentiators and Hilbert transformers, in addition to the conventional low pass, high pass, band pass and band stop filters, but this algorithm does not permit independent selection of passband and stopband ripples where as it selects a ratio of passband and stopband ripples.

1.4.2. Soft Computing in Digital FIR Filter Design

Soft Computing is an emerging approach in computation that tries to mimic the astonishing ability of the human mind to reason and learn in an environment of uncertainty. The main goal of soft computing is to develop intelligent machines to provide solutions to the hard hitting real world problems. The term "Soft computing" was introduced by Lotif Zadeh in 1981. It is an assortment of evolutionary computing, neural network and fuzzy logic. Soft computing techniques resemble biological processes more closely than traditional techniques, which are largely based on formal logical systems, such as sentential logic and predicate logic, or rely heavily on computer-aided numerical analysis. Soft computing techniques are intended to complement each other. Components of soft computing are shown in chart below.

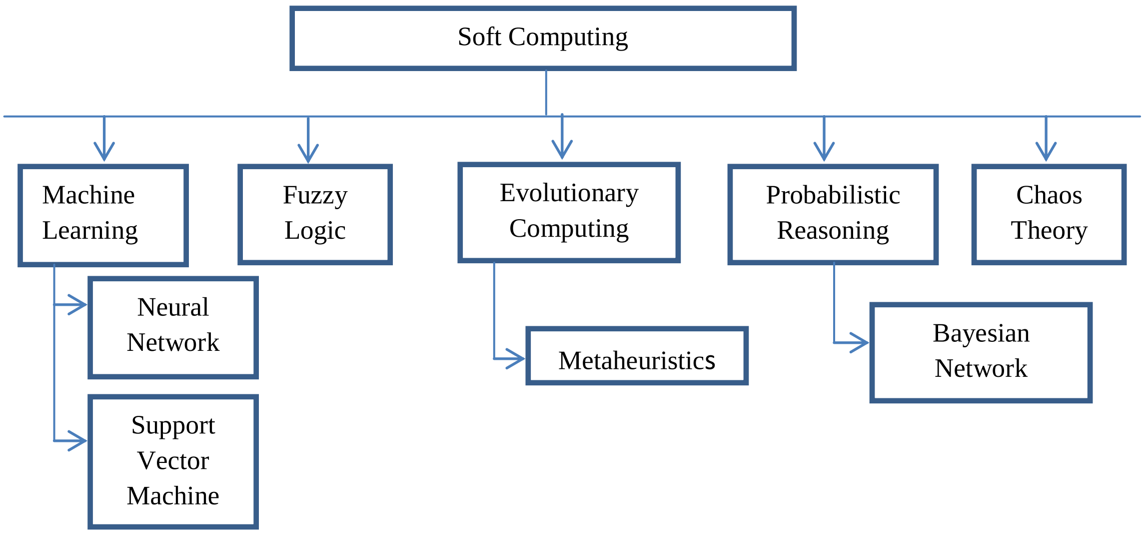

Figure 1.1: Taxonomy of Soft Computing

For implementation of FIR filters, several soft computing approaches have been proved to be effective. Limitations of conventional approaches used for digital FIR filter design motivated the researchers to use soft computing techniques to circumvent these limitations.

Neural Network was used for sharp linear phase FIR filter design synthesized by using basic and multistage frequency response masking techniques. The method was proposed using a batch back-propagation neural network algorithm with a varying learning rate mode. Construction of filters using combining Genetic Algorithm and Neural Network was proved to be effective. Recently design and analysis of low pass FIR filters was also performed using different types of learning algorithms in artificial neural network for comparative study. Fuzzy logic is another soft computing approach that has been used proficiently for filter design in isolation as well as with combination of several metaheuristic algorithms. Different types of metaheuristic algorithms including Genetic Algorithm, Particle Swarm Optimization, Simulated Annealing, Ant Colony Optimization, Artificial Bee Colony Optimization, Cuckoo Search Algorithm and their advanced versions have been widely used for FIR filter design, specifically for filter coefficient optimization, in the present decade.

1.5. Motivations

Most of the FIR filters designing techniques are based on Window method, Optimal Sampling Method and Frequency Sampling Method. Optimal approaches involve the use of Mixed Integer Linear Programming, thereby leading to higher computation time, whereas suboptimal approaches are faster, but they do not assure optimal results. Heuristic Algorithms (Kokash, 2018;, Kenny, Nathal, & Saldana, 2014; Eiselt & Sandblom, 2000) have an outstanding characteristic of searching within its neighborhood to obtain optimal solution. Hence, using heuristic optimization algorithms for obtaining perfect set of filter coefficients is much worthy. Obtaining the parameters required for implementing filters using Kaiser Window function and obtaining least noisy signal from a set filtered signal are two promising optimization problems that can be solved using heuristic algorithms.

1.5.1. Why Metaheuristics?

Heuristic search algorithms do not guarantee satisfactory results to the optimization problems with inadequate information. This limitation of heuristic search motivated to use a progressive search technique known as Metaheuristic (Yang X. S., 2010; Yang X. S., Metaheuristic Optimization, 2011) for solving hard hitting optimization problems (Desale et al., 2015). Most of the metaheuristic algorithms are motivated by some astonishing natural phenomena. These algorithms are essentially known as Nature Inspired Metaheuristics.

1.5.2. Why Quantum Inspired Metaheuristics?

Quantum Computing (QC) (Nielsen & Chuang, 2000; Yanofsky, 2007; Wolf, 2011) involves operations like coherence, de-coherence and superposition on the essential basis states which are characterized by quantum mechanical properties (Ventura & Martinez, 1997). In quantum mechanical systems, the linear combination of each possible solution outputs another solution declared through the property of superposition. QC is capable of bringing parallelism that reduces the complications of the algorithms. This advantageous characteristic of parallel processing is used adeptly in finding favorable solutions for optimization problems. Quantum behaved nature inspired metaheuristic algorithms (Dey et al., 2017; Samanta et al., 2017) have also been found to be efficient in solving a considerable number of hard hitting optimizations problems including filter coefficient optimization in lesser execution time.

1.6. Goals

The research presented in this thesis is aimed towards the following objectives.

- To identify parameters of a noisy signal and designing a soft computing based technique for filtration of noisy signal.

- To propose an algorithmic strategy for optimizing the parameters needed for implementation of digital FIR filters.
- To propose an innovative algorithm capable of optimizing set of coefficients to design hardware efficient FIR filters.
- To propose adaptive algorithm for obtaining optimized filter coefficients in lesser execution time.

The motivation behind this work is to implement an optimized filter that is capable of generating signal with minimum loss of data.

1.7. Contributions

A technique of using Genetic Algorithm is proposed for obtaining least noisy signal from a set of filtered signals. Kaiser Window with varying passband and stopband ripples has been used in this thesis work for filtration of noisy heart sound signals. Hence a set of filtered signals have been obtained. Genetic Algorithm is then used to find out the least noisy signal by using the set of filtered signals as the initial population. Considering the initial population is the starting point of the Genetic Algorithm where the initial population comprises of a set of possible solutions to the specified problem. An adaptive parameter of signals is introduced for evaluation of fitness values of the solutions.

A new strategy based on the weight lifting strategy of ants has been proposed for optimizing parameters to design digital filter using Kaiser Window function. A new innovative objective function has been introduced for optimization that performs based on the signal de-noising capability of the filters implemented by the optimized sets of parameters. A case study was carried out on heart sound signals with a filter designed using Kaiser Window with the optimized parameters.

A comparative study was carried out on the efficacy of three conventional optimization algorithms to optimize FIR filter coefficients. The efficacy of the proposed method was compared with the traditional approach of filter design with Parks McClellan algorithm as reference. Mean square error based cost function was used as the fitness function in the optimization algorithms. It is seen was observed that the BAT algorithm statistically outperforms Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) in terms of stopband attenuation characteristics and ripple performance of the designed filter.

We have also proposed design of even order low pass FIR filters and odd order bandpass FIR filters using coefficients optimized by an adaptive Global Best steered Cuckoo Search Algorithm (gbest CSA). For optimization, we use a mean square error based cost function as the fitness function. We evaluated the efficacy of the proposed technique by comparing the filter responses with responses of the filters designed using standard Cuckoo Search Algorithm (CSA) and traditional technique of filter design with Parks McClellan algorithm. Efficacy of the proposed algorithm compared to the conventional CSA is proved using seven standard benchmark functions.

A novel algorithm namely *Global Best Steered Quantum Inspired Cuckoo Search Algorithm* (GQICSA) is proposed for obtaining optimized set of coefficients to implement FIR Filter. Adder cost of a filter is estimated after quantizing the filter coefficients followed by Common Sub-expression Elimination (CSE). We found from the simulation results that reduction in word length of coefficients does not make the filters fail to achieve the ideal frequency response. Moreover, filters developed using GQICSA outperform the benchmark filters designed by Parks McClellan Algorithm in terms of stopband attenuation. Analysis of the results reveal that GQICSA not only improves over various conventional algorithms including CSA, it also surpasses modified version of CSA, Quantum Inspired CSA (QICSA) updated using quantum principles, for optimizing filter coefficients to design lower hardware costing filter without compromising the filter responses and efficiency. GQICSA also provides significant improvements compared to

CSA and QICSA in terms of stopband attenuation and execution time for higher order filter design. Efficiency of GQICSA over QICSA and conventional CSA is also proved with 16 benchmark functions.

For the same purpose for implementing hardware efficient FIR filters, a new algorithm namely Fast converging Flower Pollination Algorithm has been proposed. It has been shown by simulation results that reduction neither in word length of coefficients nor in filter order causes the filter implemented using optimized set of coefficients obtained by the proposed algorithm to be incapable of achieving the ideal frequency response. Implemented filters have been demonstrated to be effective to filter noisy Phonocardiogram signals.

1.8. Thesis Organization

In chapter 1, introduction of the thesis is presented. In this section, motivation and background of the research is shown. In chapter 2, a brief literature review is presented to show the present state of the art research which is already done in this area. The objective of the work is decided accordingly. In chapter 3, an innovative approach has been proposed of using Genetic Algorithm (GA) is for obtaining least noisy signal from a set of filtered signals. Kaiser Window with varying passband and stopband ripples are used for filtration of noisy signal, henceforth a set of filtered signals are obtained. GA is then used to find out the least noisy signal by using the set of filtered signals as initial population. For evaluation of fitness values of the solutions an adaptive parameter of the signals are introduced. Based on that parameter a set of signals are then get selected using Roulette Wheel Selection procedure. Genetic operators- crossover and mutation are then applied on the selected signals hence the optimized signal with lowest amount of noise is finally obtained. In the same chapter a new strategy based on the weight lifting strategy of ants has also been proposed for optimizing parameters to design a digital filter using Kaiser Window function. A new innovative objective function has been introduced for optimization that performs based on the signal de-noising capability of the filters implemented by the optimized sets of parameters. A case study was carried out on heart sound signals with a filter designed using Kaiser Window with optimized parameters.

In chapter 4 a comparative study of few traditional optimization algorithms namely GA, (PSO) and BAT algorithm is stated while used to obtain optimized filter coefficients. A new algorithm Global Best Steered Cuckoo Search Algorithm is also proposed for the same purpose. The motivation behind this work is to implement an optimized filter that is capable of generating signal with minimum loss of data. These kinds of filters will be very much useful in bio-medical domain where little loss of data in signals may lead to incorrect diagnosis. Aiming to reduce the execution time to optimize filter coefficients Fast Converging Cuckoo Search Algorithm (FCSA) is also proposed in this chapter. Obtained results are compared with the conventional CSA hence efficacy of the proposed algorithm is proved.

In chapter 5 innovative approaches for implementing hardware efficient FIR filter are proposed. These techniques involved use of fixed length coefficients and to obtain optimized coefficients two innovative algorithms are proposed. In fixed coefficient filter implementation, replacement of the multiplier with the shift and adder circuits is a widespread approach. The adders in this approach are dependent on the number of signed-power-of-two (SPT) terms present in each filter coefficient. A new quantum algorithm namely Global Best Steered Quantum Inspired Cuckoo Search Algorithm (GQICSA) is proposed in this thesis for obtaining optimized filter coefficients capable of acquiring desired filter responses. This approach is also proved to be effective in reducing number of SPT terms. Common sub expression elimination technique reduces the number of adders further. Another new algorithm namely Fast Converging Flower Pollination Algorithm is also used for the same purpose. These new algorithms have been proved to be much more effective compared to their traditional versions which have also been proved to be efficient in the domain of filter coefficient optimization. Another efficient nature inspired optimization technique namely Flower Pollination Algorithm (FPA) is modified and Fast Converging Flower Pollination Algorithm (FFPA) is used for obtaining optimized set of coefficients to implement a hardware efficient FIR filter in much lesser execution time. Efficacy of the proposed algorithm is proved by comparing the filter responses with few other filters designed using several conventional approaches.

Figure 1.2 shows a pictorial view of the thesis organization.

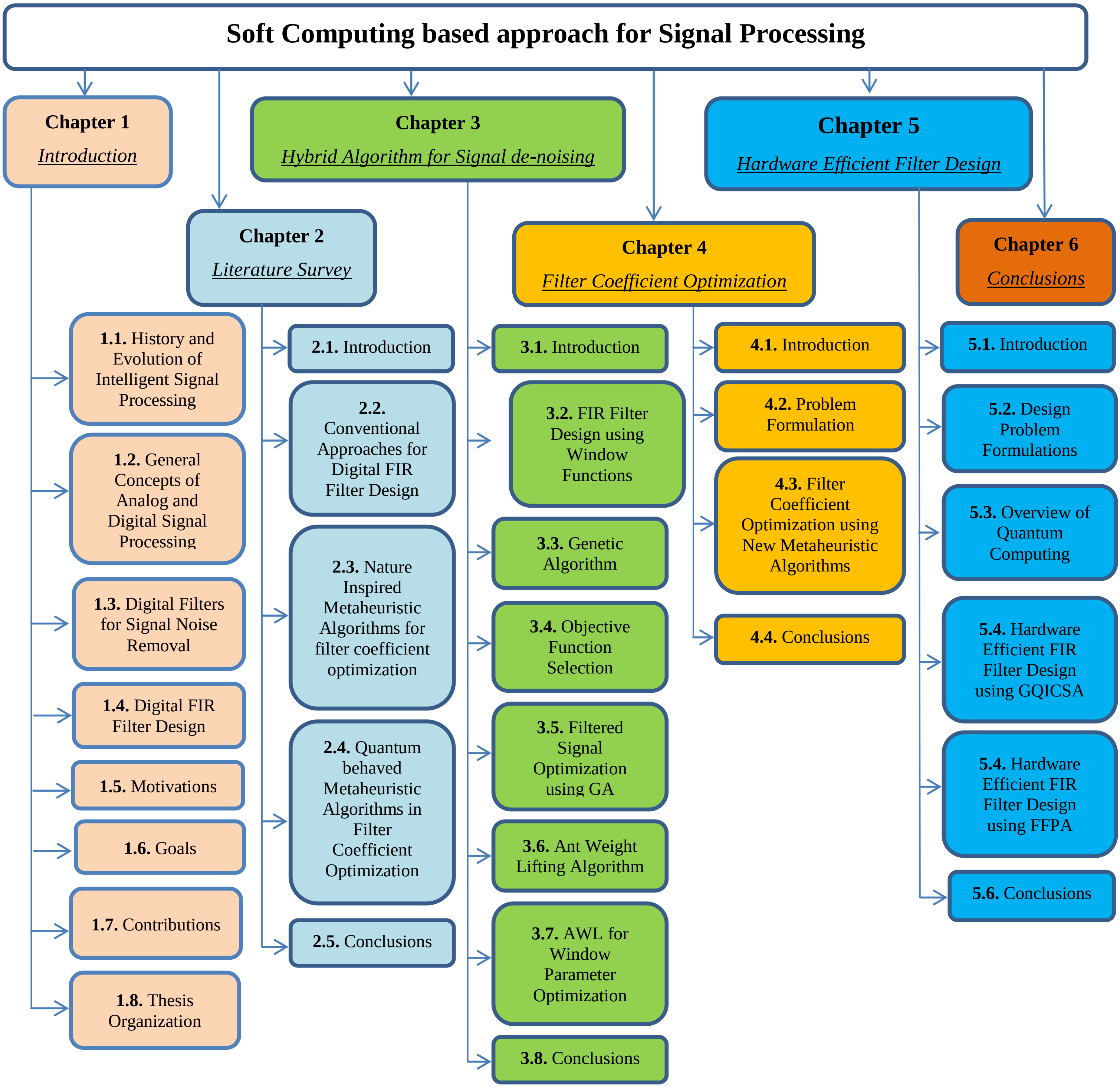

Figure 1.2: Thesis Organization

Chapter 2

Literature Survey

2.1. Introduction

The core aim of this chapter is to provide a survey and analysis of the existing approaches for implementation of digital filters. Digital filters (Antoniou, 1993; Sharma, 2009; Smith, 2011; Mitra, 2013) have been extensively used in the last few decades for noise elimination from signals (Jackson, 1996). Trifling hardware cost and extraordinary behavior of altering characteristics with changes in the discrete values stored in the registers have made the digital filters more efficacious than the analog ones. Digital filters are classified into two types - Finite Impulse Response (FIR) Filter (Sharma, 2009; Mitra, 2013; Proakis & Manolakis, 2015), and Infinite Impulse Response Filter (Sharma, 2009; Mitra, 2013). Due to minimalisms in hardware and fluently attainable linear phase properties, FIR filters are used massively compared to the IIR filters. Section 2.2 gives a brief survey of conventional approaches used for Digital FIR Filter design.

2.2. Conventional Approaches for Digital FIR Filter Design

The foremost step that the design of FIR filters involves is computation of filter coefficients. Among a considerable number of techniques proposed for finding the filter coefficients, Window methods (Kaiser, 1966; Harris & Fredric, 1978; Sharma, 2009), frequency sampling methods (Gold & Jordan, 1969) and optimal methods (Herrmann, 1970; Parks & McClellan, 1972; Reddy & Sahoo, 2015) are most efficiently used.

A traditional method for implementing FIR filters is based on the application of the Fourier series (Deshpande, 2002). Basically frequency of FIR filter is a periodic function of frequency with the period same as the sampling frequency, hence it can be expressed using Fourier series. Though Fourier series does not lead to adequate results but by amalgamation with a special class of functions called as window functions (Harris & Fredric, 1978), satisfactory results can be obtained. Approximation obtained using this method is suboptimal but the design overhead and computation costs are relatively insignificant. The most commonly used fixed window functions are Rectangular (stanford, 2018a; Sharma, 2009), Hanning (stanford, 2018b; Sharma, 2009), Hamming (stanford, 2018c; Sharma, 2009), Blackman (stanford, 2018d; Sharma, 2009) and Bartlett (stanford, 2018e). In these window functions, values of passband ripple δ_p and stopband ripple δ_s are specific as well as same. Henceforth, result will show either too small pass band ripple or too large stop band attenuation. In variable window such as Kaiser (Kaiser, 1966; Salivahanan et al., 2007; Sharma, 2009), the values of δ_p and δ_s are chosen by using the ripple control parameter specified by the designer. Application of the window function also has two effects on the amplitude response of the filter. First, the amplitudes of Gibbs' oscillations (Gibbs, 1898; Hewitt et al., 1979; Chakravarti & Mehra, 2014) in the passbands and stopbands are directly related to the ripple ratio of the window. Second, transition bands are introduced between passbands and stopbands whose widths are directly related to the main-lobe width of the window.

Frequency sampling technique was introduced by Gold and Jordan (Gold & Jordan, 1969) and further developed by Rabiner et al. (Rabiner et al., 1970). This approach involves computation of filter coefficients by sampling the ideal filter in the frequency domain. In this method a specific frequency can be approximated by fixing most of the Discrete Fourier Transform coefficients and rejecting the unspecified coefficients that lie in the range of transition bands. These unspecified coefficients are generally obtained by optimization, henceforth minimizing a weighted approximation error over the desired frequency range. Instead of having very few independent variables, frequency sampling method lacked behind due to presence of a large number of constraints and fixed control over the band edge frequencies or the passband ripples.

In case of optimal approaches (Herrmann, 1970; Parks & McClellan, 1972), filter coefficients are obtained by minimizing the maximum error between the desired and actual frequency response. An optimal approach for FIR filter design using Chebyshev sense (Darlington, 1970) was first developed by Herrmann (Herrmann, 1970). In this method frequency response of the optimal low pass filter was assumed to be equiripple in both the pass band and stop band. The number of ripples in each band was made fixed, a set of non-linear equations describing the filter were also developed using an iterative descent method. Only disadvantage of this technique is restriction in filter length of about 40. This limitation of Herrmann's approach (Herrmann, 1970) was rectified by Hofstetter et al. (Hofstetter et al., 1971), by developing an algorithm, called as "Leminiscent of the Remez exchange algorithm". But both of these methods result in extra ripple or maximum ripple filters, which are restricted subsets of optimal minimax filters. These methods also have the shortcoming of not accommodating the pass band and stop band cut-off frequencies of the filters. Another efficient optimal approach for designing non-recursive digital filter was designed in 1972 using Chebyshev approximation (Parks & McClellan, 1972). Further advancement was introduced the concept of Weighted Chebyshev approximation was used design FIR filter in 1975 (Rabiner et al. , 1975). Necessary and sufficient conditions for the best Chebyshev approximation were obtained from the classical alternation theorem. Remez exchange algorithm was established as an effective tool for the implementation of optimal filters. Subsequently, the method was extended to include all types (Odd order symmetric filter, even order symmetric filter, odd order asymmetric filter and even order asymmetric filter) of linear phase FIR filters. A computer program was developed by McClellan et al. for designing a large class of optimum FIR linear phase digital filters (McClellan & Parks, 1973). This technique comprised multi passband - stop band filters, differentiators and Hilbert transformers, in addition to the conventional low pass, high pass, band pass and band stop filters, but this algorithm does not permit independent selection of passband and stopband ripples whereas it selects a ratio of passband and stopband ripples. Linear programming is another optimal technique for designing FIR filters was introduced by Rabiener et al. (Rabiner et al., 1972) It was used for implementing an equiripple FIR Nyquist filter and equiripple FIR transmit filter. This approach was proved to be capable of avoiding the numerical ill-conditioning problems which commonly occur due to the necessity of sampling the frequency response on a very dense grid of points for high-order filters. Though linear programming is very supple and approximates an extensive variety of preferred filter shapes, it is relatively slow and henceforth restricted to limited length (Rabiner, 1972). In most of the iterative approaches, convergence speed and convergence time intensely depend on the initial estimate of the solution. Another innovative optimal technique was introduced to design constraint based FIR filter (Steiglitz et al., 1992). In this approach the simplex algorithm was used for linear programming to find the best linear-phase FIR filter of minimum length, as well as to find the minimum feasible length itself. Constrained based FIR filter design proved its efficiency to find the shorter filter length that allows the constraints of upper and lower filter responses to be met. Integer programming using branch and bounce algorithm is another effective approach that had been used to design the optimal filter (Kodek, 1980). A Mixed Integer Linear Programming (MILP) method was also used for the same purpose and was proved to be efficient in obtaining desired frequency response compared to simple rounding of coefficient values (Lim & Parker, 1983). Using Local Search Algorithm with powers-of-two coefficients (Zhao & Tadokoro, 1988) show improvement not only in filter response but also minimized the error as compared to MILP (Lim & Parker, 1983). Further improvement in Local Search Algorithm resulted in a Two-stage Local Search Algorithm (Samueli, 1989) for the implementation of multiplier less FIR filters. In this technique coefficients were represented in Canonical Signed Digit (CSD) form. Li et al. proposed a technique involving variable number of SPT terms for each coefficient (Li & Lim, 1993). After two years an approach was proposed for reducing the number of SPT terms as well as required number of adders (Li, 1995). Use of Common Sub-expression Elimination (CSE) enhances the ability of reducing total number of adders required to design optimal filters (Hartley, 1996). Further, the number of adders was reduced by sub-expression sharing using Merge-Search Algorithm (Chen & Wilson, 1999). An innovative approach for SPT term allocation for the set of filter coefficients was proposed by Lim et al (Lim et al., 1999). This technique involved determination of SPT term for each coefficient followed by optimization of coefficients using Integer Programming.

Another two-stage algorithm was used for more reduction in SPT terms by Kaakinen and Saramaki (Kaakinen & Saramaki, 2001). Combination of Local Search and Global Search were used to implement a two-stage algorithm for substantial improvement in filters response (Feng & Teo, 2008). In case of three-stage algorithm, implementation of a prototype FIR filter was performed following scaling of coefficients using a scaling factor (Yao & Chien, 2002). In this approach filter coefficients were represented in CSD form, prototype filter was designed using the most conventional Remez exchange algorithm. Finally a partial MILP algorithm was applied aiming to reduce the total number of SPT terms. An optimal algorithm was introduced in 2007 for implementation of low power multiplier-less FIR filters using Chebyshev criterion (Karakonstantis & Roy, 2007). In the same year FIR filter design using reusable Common Sub-expressions where common SPT terms are scaled and rounded aiming to obtain the CSD coefficients set was proposed by Xu (Xu et al., 2007). In this approach Local Search Algorithm was used to optimize maximum peak ripple. Successively the algorithm also reduced the required number of adders by most frequently used sub-expressions. FIR filter with sub-expression in one stage using MILP algorithm was designed in 2007 (Yu & Lim, 2007). Use of sub-expression in one stage using MILP was also proved to be efficient for implementation of FIR filters (Aktan et al., 2008). A modified branch and bounce algorithm known as FIRGAM was proposed to minimize the total number of SPT terms in a coefficient set. Yu et al. proposed an approach to reuse the CSE for FIR filter design (Yu et al., 2009). Another algorithm was introduced by Shi and Yu for implementing an FIR filter with minimal adder cost (Shi & Yu, 2011). Most of the methods discussed can be categorized into optimal and suboptimal approaches. Optimal approaches involved use of Mixed Integer Linear Programming, hence lead to higher computation time, whereas suboptimal approaches do not assure optimal results.

Heuristic Algorithms (Pearl, 1984; Eiselt & Sandblom, 2000; Kenny et al., 2014; Kokash, 2018;) have an exceptional characteristic of searching within its neighborhood to obtain optimal solution. Hence, using Heuristic optimization algorithms for obtaining filter coefficients is much worthy. A brief review of the designing strategies for the synthesis of hardware efficient FIR filters including conventional approaches and heuristic approaches have been performed by Chandra and Chattopadhyay (Chandra & Chattopadhyay, 2016). But Heuristic Search algorithms do not guarantee sufficiently good quality results to the optimization problems with limited information. Based on this bottleneck of Heuristic Search, an advanced search technique known as Metaheuristic (Bianchi et al., 2009; Yang, 2010a; Yang, 2011) Search had been used to solve hard hitting optimization problems since last decade (Desale et al., 2015). Most of the metaheuristic optimizations techniques are inspired by the astonishing behaviour of nature, hence termed as Nature Inspired Metaheuristics (Yang, 2010b).

Section 2.3 states brief discussions of using Nature Inspire Metaheuristics for filter design. Section 2.4 includes brief description of the Quantum Behaved Nature Inspired Meta-heuristics while used for obtaining optimized coefficients for digital Finite Impulse Response Filter design.

2.3. Nature Inspired Metaheuristic Algorithms for filter coefficient optimization

The term '*Meta-heuristic*' comes from the term '*meta*' which means 'beyond' or 'higher level' and the '*heuristic*' that means 'to find' or 'to discover by trial and error' (Blum and Roli 2003; Gandomi et al., 2017). Heuristics algorithms (Pearl, 1984; Kokash, 2014) are able to provide quality solutions to a hard-hitting optimization problem, but they do not guarantee the optimal solution as an output. Further improvement in 'Heuristic' approach results in 'Meta-heuristic'. All the Metaheuristic algorithms use strategy of randomization and local search. The foremost components of any metaheuristic algorithms are: intensification and diversification, or exploitation and exploration (Blum and Roli, 2003). There are a considerable number of metaheuristic algorithms in the literature that is used for FIR filter design (Gentili et al., 1995; Williams et al., 2005; Najjarzadeh & Ayatollahi, 2008; Hassan & Abbood, 2013; Singh & Josan, 2014; Tsutsumi & Suyama, 2014; Chandra, 2016). However, the well-known algorithms in the field of FIR filter design are addressed in this thesis. Most of the metaheuristic algorithms are implemented based on some natural phenomena, hence called as Nature Inspired Metaheuristic (Yang, 2008; Yang, 2014).

2.3.1. Genetic Algorithm

Genetic Algorithm (GA) (Holland, 1975; De Jong, 1975; Holland J. H., 1992; Yang, 2014; Kumar A.) mimics the mechanism of natural evolutionary principles introduced by Charles Darwin. GA was developed by John Holland and his collaborators during 1960 and 1970s (Holland, 1975; De Jong, 1975; Holland J. H., 1992). GA encodes all the data of a search space into a simple string called as a chromosome, which is usually of a fixed length (Melanie, 1999). GA uses various encoding schemes, such as, binary encoding, integer encoding, Gray encoding, and decimal encoding (Ahmed, 2008). GA is suitable for solving optimization problems. The basic advantage of this algorithm is that it has the capability to handle a number of chromosomes at the same time, where each chromosome presents a different solution to a given problem. The GA evolutionary cycle starts with a randomly selected initial population. The changes to the population occur through the processes of selection based on fitness, and alteration using crossover and mutation (Yang, 2014). The application of selection and alteration leads to a population with a higher proportion of better solutions. The evolutionary cycle continues until an acceptable solution is found in the current generation of population, or some control parameter such as the number of generations is exceeded. Selection operation can be carried out by several schemes such as Rank Selection, Roulette Wheel Selection (Obitko, 1998; Kumar R. & Joytishree, 2012) and so on.

GA had been proved as an efficient approach to design 1D FIR filters (Gentili et al., 1995; Hassan & Abbood, 2013). GA is capable of generating a population of genomes using genetic operators that represent the filter coefficients and compared the amplitude response of each genome to that of the desired amplitude response. As GA directly generated digital coefficients, there is no need to truncate coefficients for digital hardware implementation of the filter. For representing filter coefficients, two types of encoding schemes: ternary encoding and mixed encoding are mostly appreciated (Lee, 2000). To compensate the canonical signed-digit (CSD) constraints, a modified GA structure was proved to be more effective. These constraints were basically imposed by filter coefficients. An efficient Genetic Algorithm was used to design digital FIR filters with coefficients constrained to be the Sums of Power of Two Terms (Gentili et al., 1995). The implemented filter performed better with reduced computational costs. Symmetric properties of 2-D sequences and their applications for designing linear-phase 2-D FIR digital filters using GA were introduced in 2004 (Tzeng, 2004). 16 types of cases were considered according to the symmetry/anti-symmetry of 2-D sequences in both directions. Definitions of quadrantal-plane, half-plane, and full-plane filters were also described along with several numerical design examples illustrated by the GA approach. An influential genetic algorithmic approach can determine the optimal coefficients of McClellan transformation (Tzeng, 2006). It was used to design any arbitrary shape transformation contours to map from one-dimensional prototypes to two-dimensional finite-duration impulse response filters very effectively. Several numerical examples such as fan filters with arbitrary slope, elliptical filters, elliptical filters with arbitrary orientation, circular filters, and diamond-shaped filters were also demonstrated for showing the usefulness and effectiveness of the proposed approach. High throughput 2D FIR filters were implemented using Singular Value Decomposition and GA in early days of the current century (Williams et al., 2005). 1D FIR filter design was performed using Canonical Signed Digit Coefficients that caused decreasing in computation time and increase in throughput. 2D filters were designed using cascaded 1D filter in a parallel structure. A Hybrid Genetic Algorithm (GST) amalgamation of Adaptive Genetic Algorithm (AGA), Simulated Annealing (SA) and Tabu Search was used to design filters with coefficients restricted to the sum of signed powers-of-two (SPT) terms (Cen, 2007). AGA with varying population size and varying probabilities of genetic operations was used as the basis of the hybrid algorithm. SA was used to escape AGA from local optima and prevent premature convergence. The concept of tabu increased the convergence speed by reducing the search space according to the properties of filters coefficients. The authors illustrated that the normalized peak ripples of filters can be largely reduced using GST. A comparative evaluation of GA and PSO was reported while used for FIR filter implementation. Finally the performance analysis based on magnitude and gain plats clearly proved that results with PSO were better than GA (Ababneh & Bataineh, 2008). Adaptive Parameter Adjustment (APA) Genetic Algorithm was used to design FIR filters in the year of 2011 (An-xin et al., 2011). GA‘s parameters were improved based on evolutionary approach to improve the speed of convergence. As real number coding technique was used for chromosome encoding by the authors, chromosomes were represented as vectors. For selection of chromosomes two well-known selection strategies tournament selection and elitist selection were combined. GA based Artificial Neural Network was used for lowpass FIR filter design in 2012 (Thapar et al., 2012). In this approach GA was used basically for optimizing the network.

The network was trained using Multilayer Perceptron in which Error Back Propagation Algorithm had been specifically used to design Low Pass FIR filter. In 2013 Genetic Algorithm was used for searching the optimal coefficients and also to find the minimum number of Taps, and hence minimized the number of multipliers and adders (Hassan & Abbood, 2013). Self-organizing Random Immigrants Genetic Algorithm (SORIGA) was designed aiming to design multiplier-less finite impulse response filter in 2013 (Chandra & Chattopadhyay, 2013). Coefficients of the filter were encoded by binary and Canonic Signed Digit (CSD) number systems and afterwards optimized using SORIGA (Chandra, 2016). In order to prove the efficacy of the algorithm, the performance of the proposed filter was compared with existing filters in terms of impulse response and hardware cost that was measured by means of a number of performance parameters. In 2015 an optimal FIR highpass (HP) filter was implemented using the L1-norm based real-coded genetic algorithm (RCGA) (Aggarwal et al., 2015). A novel fitness function based on L1 norm was used by the authors aiming to enhance the efficiency of the proposed algorithm. Optimized filter coefficients were obtained by defining the filter objective function in L1 sense using RCGA. Simulation analysis in the paper proved efficacy of RCGA by means of signal attenuation ability of the filter, flatter passband and the convergence rate.

2.3.2. Particle Swarm Optimization

Particle Swarm Optimization (PSO) was developed by Kennedy and Eberhart in 1995 (Kennedy & Eberhart, 1995). Among all the swarm intelligence (Kennedy et al., 2001; Engelbrecht, 2005; Yang et al., 2013) based algorithms PSO is used extensively for its simplicity and flexibility. Real number randomness and global communication among the swarm particles, these are the key idea behind this algorithm. PSO quests the search space by updating the paths of each individual, termed as particle, as the piecewise pathways formed by positional vectors following quasi-stochastic manner. The movement of a swarming particle comprises two foremost components: a stochastic component and a deterministic component. Each particle is attracted toward the position of the current global best and its own best location in history, while at the same time it has a tendency to move randomly. When a particle finds a location that is better than any previously found locations, updates that location as the new current best for particle i. There is a current best for all n particles at any time t during iterations. The aim is to find the global best among all the current best solutions until the objective no longer improves or after a certain number of iterations. Different variations of PSO are namely: Accelerated PSO, Binary PSO (Yang, 2014).

PSO was used for digital FIR lowpass and bandpass filter design in 2008. Impacts of different error norms such as Least Mean Square (LMS) and Minimax were surveyed and Minimax strategy was proved to be more efficient in terms of the convergence speed and frequency responses of the implemented filters (Najjarzadeh & Ayatollahi, 2008). The effect of different population and iteration of PSO in filter implementation were also investigated. Filters designed using larger population were proved as more efficient. In the same year combination of PSO and Differential Evolution (DE) was used for FIR filter design and proved to be capable of obtaining optimized result in much lesser time rather than the conventional PSO (Luitel & Venayagamoorthy, 2008). An adaptive design technique of linear phase FIR highpass filters using Improved Particle Swarm Optimization (IPSO) was proposed in the earliest days of the present decade (Mandal et al., 2012). In IPSO a random parameters was used to control local and global searches. A new variation in the velocity was made by splitting the cognitive component into two different components. Digital high pass FIR filters was designed using Particle Swarm Optimization with Constriction Factor and Inertia Weight Approach (PSO-CFIWA) in 2011 (Mandal et al., 2011). In this algorithm velocity of the particles were modified by using inertia weight and constriction factor. Novel Particle Swarm Optimization (NPSO) Algorithm was developed for optimal FIR filter design in 2012 (Mondal et al., 2012). Particle Swarm Optimization improved the solution quality by using modernized velocity vector. Modified definition of the inertia weight had been used in NPSO to enhance the search capability that leads to a higher prospect of obtaining the global optimal solution Craziness based Particle Swarm Optimization was proposed by the same authors in 2012 for linear phase highpass FIR filter design (Kar et al., 2012). Hybrid technique designed in amalgamation of Random PSO with DE is another effective approach used for lowpass and highpass FIR filter design (Vasudhara et al., 2013). Hybrid algorithm was used in order to maintain the diversity and explore the search space more efficiently. Use of hybrid optimization techniques not only helped to reduce the execution time but also in improving the fitness significantly by saving the particles from being trapped in local minima, thus guiding them towards the global solution.

2.3.3. Ant Colony Optimization

Ant Colony Optimization (ACO) was developed by Marco Dorigo in earliest days of 90's (Dorigo & Caro). It is a paradigm for designing meta-heuristic algorithms for optimization problems and is inspired by the foraging behaviour of ant colonies. ACO is an algorithm that finds optimal paths based on the behaviour of ants searching for food. Usually ants stroll randomly, but whenever ants search out a source of food, they walk back to the colony leaving pheromones as indicator all the way leads to the food source. Other ants coming nearer to those indicators follow that path with a convinced probability (Macura, 2018). They too populate the path with their own pheromones. As more ants find and populate the path, it gets stronger until there are a couple streams of ants roving to several food sources near the colony. As the ants drop pheromones each time they travel through the path to food source to bring food, shorter paths are more prospective to be stronger, hence optimizing the "solution." Once a food source gets worn-out, the path is no longer populated with pheromones, hence slowly falloffs. ACO targets discrete optimization problems and can be extended to continuous optimization problems which is useful to find approximate solutions.

ACO applied to design fourth order Low-Pass Butterworth analog Filter Design realized with components selected from different manufactured series (Benhala, 2014). ACO also proved to be an efficient approach for digital filter design using discrete coefficients aiming to reduce execution time (Tsutsumi & Suyama, 2014). Optimization was carried out with the objective of reducing the maximum error. In ACO pheromone update was performed in such away so that pheromones could be added only on the paths of the best individuals. Properties of linear phase characteristics and assured stability allowed the FIR filters to be used with ACO and CSD coefficient (Sasahara & Suyama, 2015). CSD was used to remove non- zero digits and reduction in computation time.

2.3.4. Firefly Algorithm

In mathematical optimization, the firefly algorithm is a metaheuristic proposed by Xin-She Yang and inspired by the flashing behavior of fireflies (Yang, 2008; Yang, 2009; Yang, 2010c). Fireflies produce luminescent flashes as a signal system to communicate with other fireflies, especially to prey attractions. FFA is inspired by the firefly's biochemical and social aspects. The flashing light is produced by a process called Bioluminescence.

For improving the performance of linear phase FIR filters an adaptive designing technique was proposed using an optimization technique namely Firefly Algorithm (Saha et al., 2013). In this paper the design of lowpass, highpass, bandpass and stopband filters were shown.

2.3.5. Differential Evolution

Differential Evolution (DE) was developed by R. Storn and K. Price during 1996 and 1997 (Storn, 1996; Storn & Price, 1997). DE is a vector-based metaheuristic algorithm, which has some similarity with pattern search and GA (Yang, 2014). DE can be considered as an advancement of GA with explicit updated equations. DE is population based stochastic search algorithm. It is capable of using real numbers as solution string, hence encoding and decoding are not required. DE comprises conserving a population of candidate solutions and also involves iterations of recombination, evaluation, and selection. In recombination, a new candidate solution is generated based on the weighted difference between two randomly selected population members added and then to a third population member (Brownlee , 2011). This agitates population members relative to the spread of the wider population. In combining with selection, the perturbation effect self-organizes the sampling of the problem space, by bounding it to already known spaces of interest. The Differential Algorithm is basically proposed for numeric optimization problems.

DE algorithm was applied to the design of digital finite impulse response filters in 2006 (Karaboga & Cetinkaya, 2006). The new DE algorithm based on reserved genes was used to implement digital finite impulse response filters in 2010 (Liu et al., 2010). New vectors were produced by combining the genes of the selected chromosomes. Newly generated vectors were then evolved with other individuals in the population. Besides increasing the diversity of population the algorithm was also proved to be effective in avoiding the local optimal solution.

Use of novel self-adaptive Differential Evolution algorithm for the same purpose was introduced by Chandra and Chattopadhyay in 2011(Chandra & Chattopadhyay, 2011). Critical analysis of different mutation strategy of DE while used for implementing FIR filters had been performed in 2012 by the same authors (Chandra & Chattopadhyay, 2012). Best mutation strategy was chosen based on the convergence speed as well as the frequency response of the designed filters. For higher order filters such as for 29^{th} order filter a mutation scheme could be chosen as best in terms of associated hardware cost. Next year a trigonometric mutation strategy was proposed by them for deigning a multiplier-less FIR filters using DE (Chandra & Chattopadhyay, 2013). In 2015 Differential Evolution Algorithm had been proved to be worth in designing a hardware efficient FIR filter (Reddy & Sahoo, 2015). In this approach using DE algorithm, first a set of filter coefficients with reduced number of signed-power-of-two (SPT) terms had been obtained without compromising on quality of the filter response. Then the Common Sub-expression Elimination Algorithm (CSE) was applied, and the hardware cost was determined in terms of required number of structural as well multiplier adders. The filters were designed using DE for various word lengths, and the same were implemented in transposed direct form (TDF) structure. The implemented filters were synthesized in Cadence RTL compiler using UMC 90 nm technology. Performances of the proposed filters were compared with recently best published works in terms of area, delay, power and power-delay-product (PDP). Use of DE was proved to be effective in highpass FIR filter design in terms of minimizing the magnitude approximation error and ripple magnitudes of pass-band and stop band (Kirandeep & Singh, 2015).

2.3.6. Cuckoo Search Algorithm

Cuckoo Search (CS) is a modern nature inspired metaheuristic algorithm that is broadly used for solving hard-hitting optimization problems. Cuckoo Search Algorithm (CSA) was developed by Yang and Deb in 2009 (Yang & Deb, 2009; Yang & Deb, 2010; Yang & Deb, 2013). CSA is based on the brood parasitism of the cuckoo species. It also uses a balanced composition of a local random walk and global explorative random walks, controlled by a switching parameter p_a. Global random walk is carried out by a superior kind of random walk namely Lévy Flights (Pavlyukevich, 2007). Predefined parameter bounds state the domain to choose the initial population. Lévy flights belong to a class of random walks formulated by Paul Lévy in 1937 (Yang, 2014) by generalizing Brownian motion (Brown & Liebovitch, 2007) and comprising non-Gaussian randomly dispersed step sizes for the distance covered. Lévy Flights are capable of maximizing the probability of resource searches in uncertain surroundings. In Optical science, Lévy flight can be defined as a term used to designate the motion of light (Barthelemy et al., 2008). Sometimes, light follows a random series of shorter and longer steps rather than travelling in a predictable Brownian diffusion. The shorter and longer steps together form a Lévy flights walk. Most of the natural search processes use Lévy flights. Some bee species perform Lévy flights to find the flowers in a new area. Survey says, by performing Lévy flights vaster area can be covered than normal random search. Performing Lévy Flight is also additionally informative than the traditional search methods. Some shark species follow random Brownian motion while searching food; however, if they failed to get food items, they stat following Lévy flight behavior, mixing short random movements with long trajectories.

CSA proved its efficiency in determining the optimal coefficients of FIR filters over other nature inspired algorithms. It is also capable of optimizing coefficients of FIR-fractional order differentiator (FIR-FOD) (Kumar & Rawat, 2014). An adaptive weighted least square (WLS) fitness function was adopted by the authors to improve the response of the FOD. Use of CSA assuage from intrinsic drawbacks of premature convergence and stagnation unlike Genetic Algorithm. Use of Cuckoo Search with adaptive Lévy step size for implementation of computationally efficient lowpass FIR filters was introduced in 2016 (Sengupta & Basak, 2016). An advanced version of CSA namely gbest-guided Cuckoo Search Algorithm (GCSA) was proposed in 2016 to design an IIR filter (Chakraborty et al., 2016). Just after one year GCSA proved its ability to implement higher order two channel filter banks (Dhabal & Venkateswaran, 2017). GCSA performed much better compared to other conventional algorithms in terms of filter responses. Unlike standard CSA, the proposed GCSA used replacement strategy based on global best solution for replacing old nests instead of random walks, and achieved faster convergence. For better exploration of the search space, instead of assuming a fixed value of $\lambda = 1.5$ in Lévy's distribution in GCSA λ was obtained using an equation within a range. Aiming for a faster convergence, the authors adjusted the parameter p_a which stands for probability of choosing worse quality nests to be replaced. To reduce the execution time, the authors modified practical implementation of the algorithm; instead of invoking the cost function separately by individual nest as in standard CSA, the authors make the whole population of nests to invoke the cost function at a time in case of GCSA that reduced functional overhead and hence execution time.

Metaheuristic is actually advanced heuristic are capable of providing an adequately good solution to an optimization problem, especially with incomplete or imperfect information or restricted computation capacity. Heuristic search problem can be defined as follows: Given the ability to execute a probabilistic 'guessing' algorithm A, and a checking function f, such that $\Pr[A\ outputs\ w\ such\ that\ f(w) = 1] = \varepsilon$, output $w\ such\ that\ f(w) = 1$. It can be solved by running the algorithm repeatedly and testing the output using the checking function (Montanaro, 2016). In case of classical algorithms this problem will result in an average case of $O(1/_{\varepsilon})$ evaluations of f. However a quantum algorithm is capable of finding the output w such that $f(w) = 1$ only in $O(1/_{\sqrt{\varepsilon}})$ evaluations of f, hence attaining the speedup (Brassard et al., 2002). Failure probability in this case also tends to 0.

2.4. Quantum Inspired Metaheuristic Algorithms in Filter Coefficient Optimization

Quantum Computing (QC) (Benioff, 1980; Fynman, 1982; Deutsch, 1985; Nielsen & Chuang, 2000; Yanofsky, 2007; Wolf, 2011) originates from the study of quantum mechanics and the functionalities of quantum mechanical devices. This involves operations such as coherence, de-coherence, superposition and entanglement on the essential basis states which are characterized by quantum mechanical properties (Ventura & Martinez, 1997). In quantum mechanical systems the linear combination of each possible solution outputs another solution is declared through the property of superposition. In the present decade QC is a developing field of computer science. QC is capable of bringing parallelism that reduces the complications of the algorithms. This outstanding ability of parallel processing can be used proficiently in finding promising solutions for optimization issues. Quantum behaved metaheuristic algorithms (Dey et al., 2017; Samanta et al., 2017) have proved their efficiency in solving a considerable number of hard hitting optimizations problems.

PSO had been already was proved to be effective for highpass FIR filter design (Mandal et al., 2011; Mandal et al., 2012). In 2015 Quantum Behaved Particle Swarm Optimization (QPSO) proved to be able of performing much better than PSO while used for highpass FIR filter design (Dhabal & Sengupta, 2015). Further improvement in QPSO was accomplished and Weighted Mean Best Quantum behaved Particle Swarm Optimization (WQPSO) was used for the same purpose (Dhabal & Sengupta, 2015). WQPSO proved to be more efficient rather than PSO and also QPSO. Lowpass FIR filter designed using the coefficient optimized by Particle Swarm Optimization had also been proved to efficient in terms of different filter characteristics. Quantum behaved Particle Swarm Optimization proved to be efficient in implementation of highpass FIR filters. Quantum Inspired Multi-objective Cat Swarm Optimization (Q-MCSO) algorithm also proved to be effective for digital FIR filter design (Dwivedi & Patel, 2017). Efficacy of the algorithm was validated by comparing classical Multi-objective Cat Swarm Optimization (MCSO) algorithm and few other standard evolutionary algorithms. Q-MCSO outpaces all other algorithms not only to meet the specification for a filter of a specific order but also by achieving the desired filter responses with minimum power consumption.

2.5. Conclusions

There are a considerable number of techniques proposed by the researchers for digital FIR filters design since the earliest days of signal processing. Most of the earlier approaches can be categorized within three basic types such as window method, frequency sampling method and optimal method. Among these three tactics use of optimal approaches has been spread widely due to the ability of acquiring desired frequency response with minimized hardware complexity. Optimal approaches involve optimization of filter coefficients by minimizing the maximum the error between the desired and actual frequency response. Among a significant number of optimal techniques, Remez exchange algorithm was effectively used as an effective tool for the implementation of optimal filters. Parks McClellan algorithm is another approach for implementation of optimum FIR linear phase digital filters, but this method does not permit independent selection of passband and stopband ripples where as it selects a ratio of passband and stopband ripples.

Most of the previous methods available in the literature can be categorized into optimal and suboptimal approaches. But the limitations of optimal approaches involve use of Mixed Integer Linear Programming hence lead to higher computation time however suboptimal approaches do not assure optimal results. Hence, use of heuristic optimization algorithms for obtaining filter coefficients is much worthy due to the outstanding characteristic of searching in neighborhoods. But Heuristic Search algorithms do not guarantee sufficiently good quality results, to the optimization problems with incomplete information. This limitation of Heuristic Search inspired the researchers to use advanced search technique such as metaheuristic Search used to solve the optimization problems with inadequate information. Most of the metaheuristic algorithms are basically inspired by the natural phenomenon. Nature inspired Metaheuristics are being widely used for filter coefficient optimization since the last few decades. Amalgamation of quantum principle with the nature inspired Metaheuristics not only reduces the computation time of the algorithm, but also enhances the efficacy. In our work an innovative nature inspired metaheuristic algorithm has been used for obtaining optimized set of filter coefficients. Moreover, the algorithm is further improved by incorporating quantum principle, hence used to obtain hardware efficient filters with desired frequency response in lesser execution time.

Chapter 3

Hybrid Algorithms for Signal Noise Removal

3.1. Introduction

Signal transmission habitually includes unwanted components termed as noise in the signal. Exclusion of noise from transmitted signals still remains a research hotspot for the researchers in the field of signal processing. Among different techniques used for noise removal, use of filters has been proved to be most effective. Digital filters (Antoniou, 1993; Smith, 2011; Sharma, 2009; Mitra, 2013) have been used extensively compared to the analog filters. Trifling hardware cost and extraordinary behavior of altering characteristics with changes in the discrete values stored in the registers have made the digital filters more efficacious than the analog ones. Digital filters are classified into two types - Finite Impulse Response (FIR) Filter (Sharma, 2009; Mitra, 2013; Proakis & Manolakis, 2015), and Infinite Impulse Response Filter (Sharma, 2009; Mitra, 2013). Since the earliest days of discrete time systems, both the types of FIR and IIR systems have gone through several advancements in different eras. Design of digital FIR filters involves calculation of filter transfer function coefficients that provide target frequency response. Among a considerable number of techniques proposed for finding the filter coefficients, Window method (Sharma, 2009), frequency sampling and optimal algorithm are most efficiently used.

The most commonly used fixed window functions are Rectangular (Stanford, 2018a), Hanning (stanford, 2018b; Sharma, 2009), Hamming (stanford, 2018c; Sharma, 2009), Blackman (stanford, 2018d; Sharma, 2009) and Bartlett (stanford, 2018e; Salivahanan et al., 2007; Sharma, 2009). In these window functions, values of passband ripple and stopband ripple are specific as well as same. Henceforth, result will show either too small pass band ripple or too large stop band attenuation. In variable size window functions such as Kaiser Window (Kaiser, 1966; Sharma, 2009), the values of passband and stopband ripples are chosen by using the ripple control parameter specified by the Designer.

In this chapter use of Kaiser Window function (Kaiser, 1966; Mitra, 2013) is used for removal of noise from signals in two different contexts. At the initial case, filtration of a noisy heart sound signal was performed using several filters implemented by Kaiser Window function (Sharma, 2009; Mitra, 2013) with varying ripple factors. Henceforth, from the obtained set of filtered signals the optimum signal with lowest amount of noise was achieved using the well-known iterative evolutionary search strategy Genetic Algorithm (GA) (Holland, 1975; De, 1975; Holland, 1992; Yang, 2014; Ye et al., 2013; Chauhan & Arya, 2011). Considering initial population is the starting point of the GA where the initial population comprises set of possible solutions to the specified problem. In this proposed technique for noise removal, obtained set of filtered signals using Kaiser Window is used as the initial population.

At the former case another search technique Ant Weight Lifting (Samanta et al., 2013) algorithm was innovatively used to determine the optimized set of Kaiser Window parameters while used to de-noise the same heart sound signal. A new innovative objective function has been introduced for optimization that performs based on the signal de-noising capability of the filters implemented by the optimized sets of parameters.

3.2. FIR Filter Design using Window Function

The elementary notion behind the use of Window functions to design filters is represented as follows (Singh & Joshan, 2014; Mallick et al., 2014; Neha & Singh, 2014):

$$H_d\left(e^{j\omega}\right) = 1\ for\ 0 < \omega \le \omega_{cutoff}$$

$$= 0\ for\ \omega > \omega_{cutoff} \qquad (3.1)$$

This means in case of ideal filters, frequency response must be equals to 1 in the pass band, and equals to 0 in the stop band.

The unit sample response $h_d(n)$ is related to the desired frequency response $h_d(\omega)$by the Fourier transform relation as follows:

$$h_d(\omega) = \sum_{n=0}^{\infty} h_d(n) e^{-j\omega n} \qquad (3.2)$$

$$h_d(n) = \frac{1}{2\pi}\int_{-\pi}^{\pi} H_d(\omega) e^{j\omega n} d\omega \qquad (3.3)$$

Aiming to implement a FIR filter of length M, unit sample response $h_d(n)$ must be restricted at the specific length of (M-1). It can be performed by multiplying $h_d(n)$ with the basic window function, termed as Rectangular Window function (Sharma, 2009).

A rectangular window of length M can be expressed by the following Eqn (Sharma, 2009).

$$w_R(n) = 1\ for\ n = 0,1,2, \dots \dots .. M-1$$

$$= 0\ elsewhere \qquad (3.4)$$

Hence, the unit sample response $h(n)$ of the FIR filter can be represented by the Eqn. below (Sharma, 2009):

$$h(n) = (h_d(n)) w_R(n) \qquad (3.5)$$

Substituting $w_R(n)$ from Eqn. 3.7, $h(n)$ is limited to the length M,

$$h(n) = h_d(n)\ for\ n = 0,1,2, \dots \dots .., M-1$$

$$= 0\ elsewhere \qquad (3.6)$$

Frequency response of FIR filter is obtained by performing Fourier transform of Eqn. 3.5 (Sharma, 2009),

$$H(\omega) = FT\{h_d(n). w(n)\} \qquad (3.7)$$

Multiplication of the window function $w(n)$ with the unit sample response $h_d(n)$ is equivalent to the convolution of $H_d(\omega)$with $w(\omega)$ which represents the Fourier transforms of the window function (Sharma, 2009).

$$w(n) = \sum_{n=0}^{M-1} w(n)e^{-j\omega n} \tag{3.8}$$

This convolution has the smoothing effect on the frequency response $h_d(\omega)$ of the implemented filter. Increment in the length M causes reduction in the smoothing effect provided by the window function. Overshoots and ripples in frequency response arise due to the high oscillation or side lobes caused by abrupt truncation. Aiming to reduce these effects, few windows were introduced for the implementation of FIR digital filters that might not contain hasty discontinuity in time and frequency domain characteristics. In Table 3.1 different conventional window functions for implementation of FIR filters are shown. These window functions have specific values of passband ripple δ_p and stopband ripple δ_s. Hence, result will show either too small pass band ripple or too large stop band attenuation. Overcoming this limitation a variable size window known as Kaiser (Kaiser, 1966) Window was proposed. Values of δ_p and δ_s are chosen by using the ripple control parameter specified by the Designer in case of Kaiser Window function.

Table 3.1: Comparative study of Window Functions

Name of Window	**Time-domain sequences, $h(n), 0 \le n \le M-1$**	**Transition width of Main Lobe**	**Peak Side Lobes (dB)**
Rectangular (stanford, 2018a; Sharma, 2009)	1	Narrowest main lobe about $\frac{4\pi}{M+1}$	-13 dB
Barlett (Triangular) (Sharma, 2009; stanford, 2018e)	$1-\frac{2\left\|n-\frac{M-1}{2}\right\|}{M-1}$	Medium main lobe of $\frac{8\pi}{M}$, Leakage factor is 0.28%	-25 dB
Blackman (Sharma, 2009; stanford, 2018d)	$0.42-0.5\cos\frac{2\pi n}{M-1}+0.8\cos\frac{4\pi n}{M-1}$	Large main lobe $\frac{12\pi}{M}$, Leakage factor 0%	-57 dB
Hamming (Sharma, 2009; stanford, 2018c)	$0.54-0.46\cos\frac{2\pi n}{M-1}$ for n=0,1,…,M-1	Medium main lobe $\frac{8\pi}{M}$, Leakage factor 0.03%	-41 dB
Hanning (Sharma, 2009; stanford, 2018b)	$\frac{1}{2}(1-\cos\frac{2\pi n}{M-1})$ For n=0,1,….,M-1	Medium main lobe $\frac{8\pi}{M}$, Leakage factor 0.05%	-31 dB
Tukey (Blackman and Tukey, 1959)	$\frac{1}{2}[1+\cos\pi\frac{\|n\|-\alpha\frac{N}{2}}{(1-\alpha)\frac{N}{2}}]\ when\ \frac{\alpha N}{2}\le\|n\|\le\frac{N}{2}$ $=0\ when\ 0\le\|n\|\le\frac{\alpha N}{2}$	Main lobe $-3dB$, Leakage factor 3.57%	-15.1dB

3.2.1. Kaiser Window Function

Kaiser window function was first proposed by James Kaiser at Bell Laboratories (Kaiser, 1966) to design non-recursive digital filters by using the modified zeroth order Bessel function (I_0-sinh) (Kaiser et al., 1980). Later, Kaiser Window (Avci & Nacaroğlu, 2008; Roy & Morshed, 2013; Lin, 1998; Kaur, 2014) has gone through several modifications proposed by different researchers. Kaiser Window permits separate control on the width of the main lobe and attenuation of the side lobes. Kaiser window is defined by the following Eqn.

$$w_k(n) = \begin{cases} \dfrac{I_0\left\{\left[1-\left(\frac{n-\alpha}{\alpha}\right)^2\right]^{\frac{1}{2}}\right\}}{I_0(\beta)} & for\ 0 \le n \le M-1 \\ \end{cases}$$

$$= 0\ elsewhere \tag{3.9}$$

M is the length of the window and I_0 is the first kind 0th order modified Bessel function (Sharma, 2009).

Here, $\alpha = \frac{M-1}{2}$ (3.10)

$$M - 1 = \frac{A-8}{2.285\Delta\omega} \tag{3.11}$$

$$\Delta\omega = \omega_{stop\_band} - \omega_{pass\_band} \tag{3.12}$$

ω_s is stopband edge frequency and ω_p is passband edge frequency. β is the shape of the window which can be selected independently. By changing value of β and the length of the filter main lobe width and side lobes attenuation can be adjusted. There is ripple of $\pm\delta_1$in the passband and $\delta_2$in the stopband. For the FIR filter design using Kaiser Window, minimum ripple of $\delta_1$and δ_2 is considered. Let the minimum ripple be represented by δ. If attenuation is defined in dB,

$$A = -20log_{10}\delta \tag{3.13}$$

Value of β can be found out by using the following Eqns.

$$\beta = 0.1102(A - 8.7)\ for\ A > 50$$

$$= 0.5842(A - 21)^{0.4} + 0.07886(A - 21)\ for\ 21 \leq A \leq 50$$

$$= 0\ for\ A < 21 \quad (3.14)$$

3.3. Genetic Algorithm (GA)

Genetic Algorithm (GA) (Holland, 1975; De Jong, 1975; Holland J. H., 1992; Yang, 2014) obeys the mechanism of natural evolutionary principles introduced by Charles Darwin. GA was developed by John Holland and his collaborators during 1960 and 1970s (Holland, 1975; De Jong, 1975; Holland J. H., 1992). GA encodes all the data of a search space of a problem into a simple strings of fixed length termed as chromosomes (Melanie, 1999). GA uses different encoding schemes, such as, binary encoding, integer encoding, Gray encoding, and decimal encoding (Ahmed, 2008) and so on. GA is suitable for solving hard-hitting optimization problems. The basic advantage of this algorithm is that it has the capability to handle a number of chromosomes at the same time, where each chromosome presents a different solution to a given problem. The GA evolutionary cycle starts with a randomly selected initial population. The changes to the population occur through the application of the genetic operators: selection, crossover and mutation (Yang, 2014). The application of these genetic operators leads to a population with a higher proportion of better solutions. The evolutionary cycle continues until an acceptable solution is found in the current generation of population, or some control parameter such as the number of generations is exceeded. Selection operation can be carried out by several schemes such as Rank Selection, Roulette Wheel Selection (Obitko, 1998; Kumar & Joytishree, 2012) and so on. Mixing of two solutions in the population is performed in crossover operation. It actually causes convergence in a subspace. Crossover (Holland, 1975; Yang, 2014) is performed with a fixed probability termed as crossover probability(p_a), typically a high valued probability in the range of0.7 ~ 1.0.Value of crossover probability should not be too small as it leads to inefficient evolution. Mutation (Holland, 1975; Yang, 2014) is basically performed to increase the diversity of the population by changing a part of a solution randomly. It also helps to get away from local optimum. Mutation is performed with a mutation probability (p_m). Value of mutation probability is usually small, ranging between 0.001 and 0.05.Use of high valued mutation probability may cause diversify even when optimal solution is approaching. Another operator selection, termed as elitism, is basically an operator that passes on the best quality solutions to the next generations. A standard GA for a minimizing optimization can be described as in Algorithm 1 (Holland, 1975; Yang, 2014).

Input: Population size (n), maximum number of iterations($iter_{MAX}$), objective function (f), lower and upper bound.
Output: Global best solution$BEST$.
Begin
Initialize probability of crossover (p_c) and probability of mutation (p_m)
Generate initial population P within bounds and store them in matrices x and y
Evaluate the fitness of the solutions (P) stored in x using f
$BEST = \mathrm{argmin}_i\, f(x_i)$; $fitness_{BEST} = \min_i f(x_i)$; $t = 1$
while($t < iter_{MAX}$)
  Generate new solutions by crossover and mutation with probability $p_c$and p_m respectively
  Replace the old solutions stored in y by the newly generated solutions
  Evaluate the fitness of the solutions stored in y using$f$
  for $i = 1{:}n$
    if ($f_{t+1}(x_i) < f_t(x_i)$)
      $x_i^{t+1} = y_i^{t+1}$
    Else
      $x_i^{t+1} = x_i^t$
    End if
  End for
  Evaluate the fitness of the solutions using f
  $CURRENT\_BEST = \mathrm{argmax}_i\, f(x_i)$; $fitness_{CURRENT\_BEST} = \max_i f(x_i)$
    if ($fitness_{CURRENT\_BEST} < fitness_{BEST}$)
      $BEST = CURRENT\_BEST$; $fitness_{BEST} = fitness_{CURRENT\_BEST}$
    End if
    $t = t + 1$
End while
End

Algorithm 3.1: *Genetic Algorithm*

3.3.1. Selection

In selection parents are selected based on the fitness values aiming to generate off-springs for the next generation with better fitness values. Selection of parents is very vital to the convergence rate of the algorithm, as good parents steer individuals to a better and fitter solutions. Among a considerable number of selection procedure Roulette Wheel Selection is the most effectively used.

3.3.1.1.Roulette wheel Selection

Roulette Wheel Selection (Atanassov et al., 2009) was introduced by Holland. In this selection strategy parents are selected based on the fitness values, fitter individuals have more chances to be selected. At first a circular wheel divided into n pies must be considered, where n is the number of individuals in the population. Each individual must get a portion of the circle proportional to its fitness value. Next, a specific point must be chosen on the wheel circumference before rotating the wheel. The area of the wheel that comes in front of the fixed point is chosen as the parent. For choosing the second parent also, similar process is repeated. Fitter individual has greater pie within the wheel, hence has more chance to be selected when the wheel rotates.

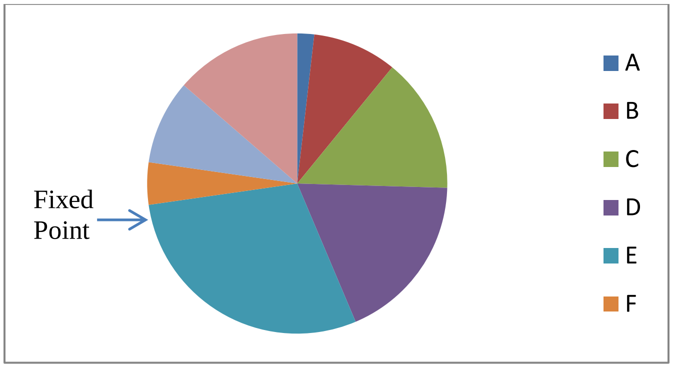

Figure 3.1: Roulette Wheel selection

Probability of each individual in this selection procedure can be described by the following Eqn.

$$P[Individual\ i\ is\ chosen] = \frac{F_i}{\sum_{j=1}^{Pop\ size} F_j} \tag{3.15}$$

F_i and F_j stand for fitness values of the i^{th} and j^{th} individuals respectively. Algorithm 3.2 describes Roulette Wheel Selection procedure in detail.

Step 1: The fitness function is calculated for each chromosome, providing fitness values which are then normalized.

Step 2: The population is arranged in ascending order according to the fitness values.

Step 3: Accumulated normalized fitness values are obtained (Accumulated fitness value of a chromosome = Fitness value of that chromosome + the fitness values of all the previous chromosomes). The accumulated fitness of the last individual must be 1.

Step 4: A random value should preferably be chosen between 0 and 1.

Step 5: The selected chromosome will be the first one whose accumulated normalized value is greater than the randomly chosen value.

Step 6: Step 1 to step 5 are repeated until the initial population converges.

Algorithm 3.2: *Roulette Wheel Selection*

3.3.2. Crossover

Genetic operator crossover generates new off-springs from the selected pair of parents for the next generation. Crossover is performed in a fixed probability p_a. Strategy of crossover can be classified based on the number points of splicing. In case of single point crossover two individuals of the current generation are spliced at a crossover point and swapping the spliced parts are performed. Aim is to combine good characteristics of one gene of an individual may with some good genes of another individual to create a better solution represented by the new off-spring.

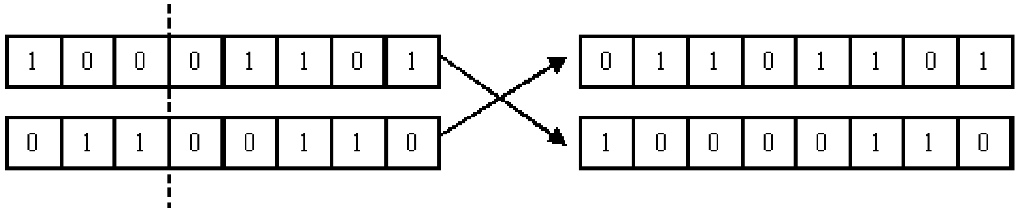

Figure 3.2: Single Point Crossover

3.3.3. Mutation

In mutation genetic composition is adjusted randomly aiming to introduce new characteristics in a population that has not been achieved through the crossover. Genetic operator mutation current value of a gene changes to a different one. For binary string individual values of a gene flipped from 0 bit to 1 or vice versa. Mutation is also performed with a fixed probability for a problem.

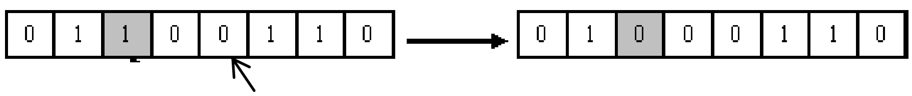

Figure 3.3: Mutation

3.4. Objective Function Selection

In this chapter an innovative objective function is introduced aiming to obtain signal with much reduced amount of noise. This function is termed as the β factor of the signal. Finally the value of $10\log_{10}(\beta)$ has been used as the fitness value, signal with higher value of $10\log_{10}(\beta)$ states signal with lesser amount of noise.

$$\beta = \frac{(\text{Corrupted Signal} - \text{Filtered Signal})}{\textit{Filtered Signal}} \tag{3.16}$$

3.5. Filtered Signal Optimization using GA

In our proposed technique for obtaining optimized filtered signal using GA, following steps are followed:

Step 1: Filters are implemented using Kaiser Window with varying passband and stopband ripples (passband ripple varies from 0.01 to 0.40 and stopband ripple varies from 0.09 to 0.49). Corrupted signal is then filtered using the implemented filters. Set of filtered signals is considered as initial population. Each filtered signal is termed as a chromosome.

Step 2: For determination of fitness values of chromosomes, β factor is used. Finally the value of $10\log_{10}(\beta)$ has been used as the fitness value.

Step 3: Based on fitness values a set of filtered signals has been selected from the initial population using Roulette Wheel Selection procedure.

Step 4: Single point Crossover is performed with 100% probability in between the selected set of chromosomes and off springs are generated. (100% probability of crossover has shown the best results.

Step 5 Mutation is performed on the offspring chromosomes with 25% probability. (25% probability of mutation has shown the best results).

Step 6: Replacement of parent signals by off-spring signals with better fitness values than the parent signals.

Step 7: Signal with highest fitness value has been obtained as best offspring signal.

Step 8: Repeat Step 4 to Step 7 N (N=10) times.

Algorithm 3.3: GA for Filtered Signal optimization

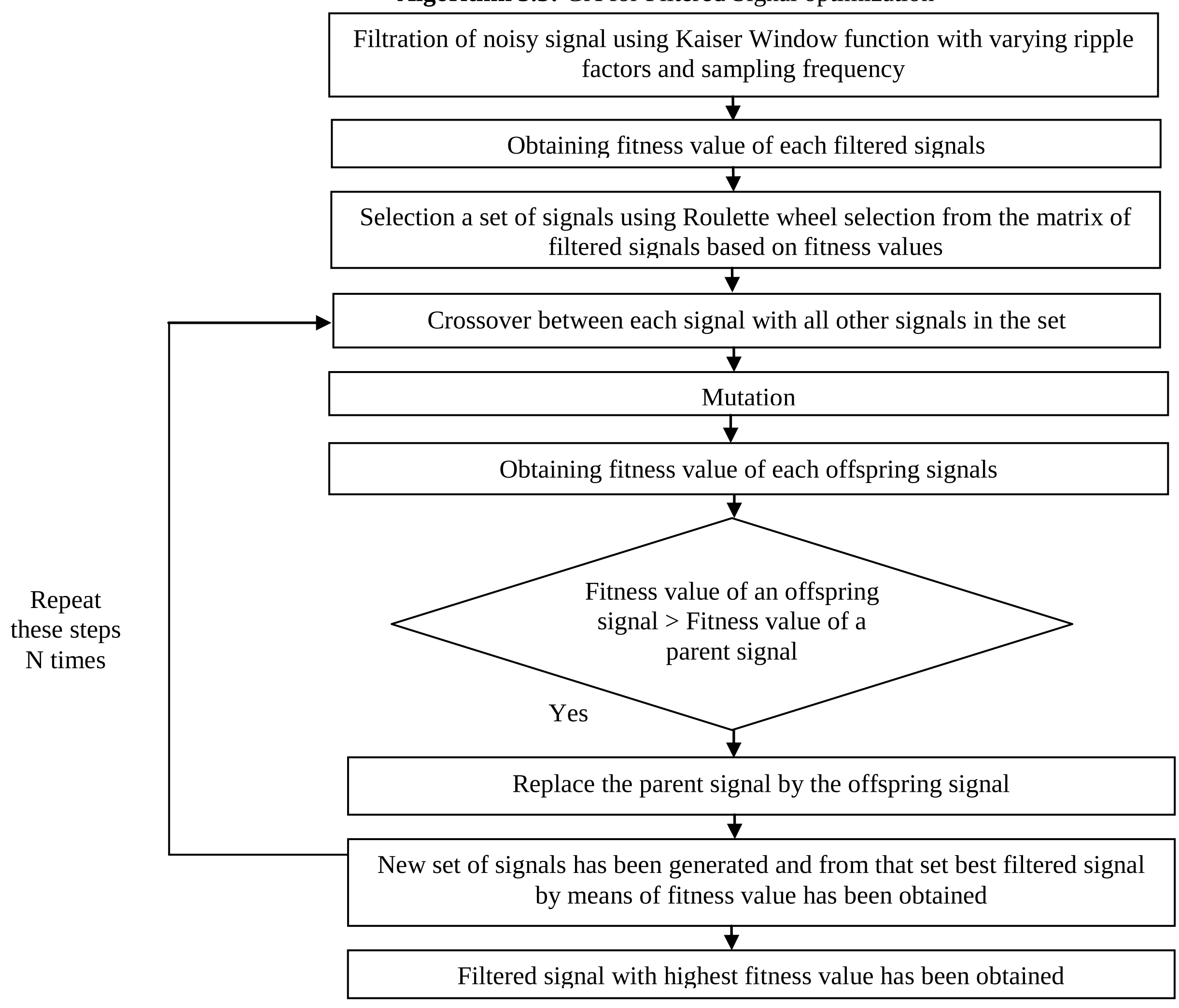

Figure 3.4: Flowchart of filtered signal optimization *using GA*

Figure 3.4 presents flowchart of the proposed technique.

Proposed approach is capable of identifying little amount of noise present in the signal and excluding it from the signal. Hence, it is very much useful for de-noising biomedical signals, where very little amount of noise may cause erroneous diagnosis.

3.5.1. Case Study

A heart Sound Signal without any noise has been collected from a diagnostic center. Random noise has been incorporated in the original heart sound signal. Original Heart Sound Signal and Noisy Heart Sound Signal have been shown in Figure 3.5 (a) & (b). SNR (Signal to Noise Ratio) of the corrupted signal is 2.9126 and correlation of the corrupted signal is 0.8015.

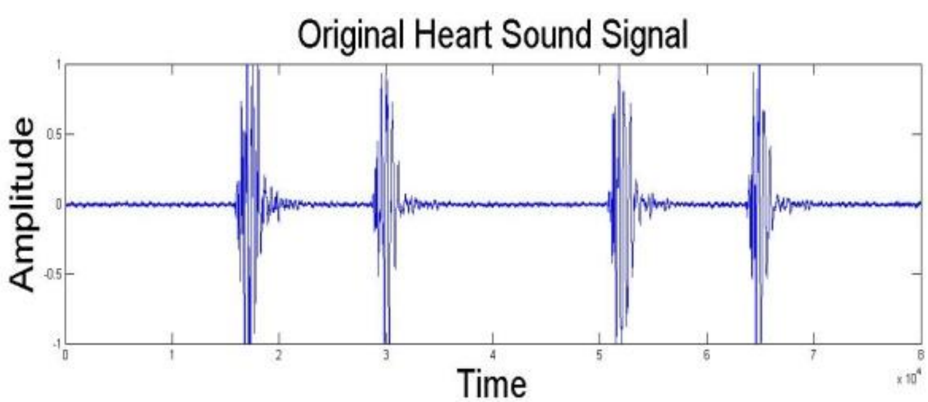

Figure 3.5(a)

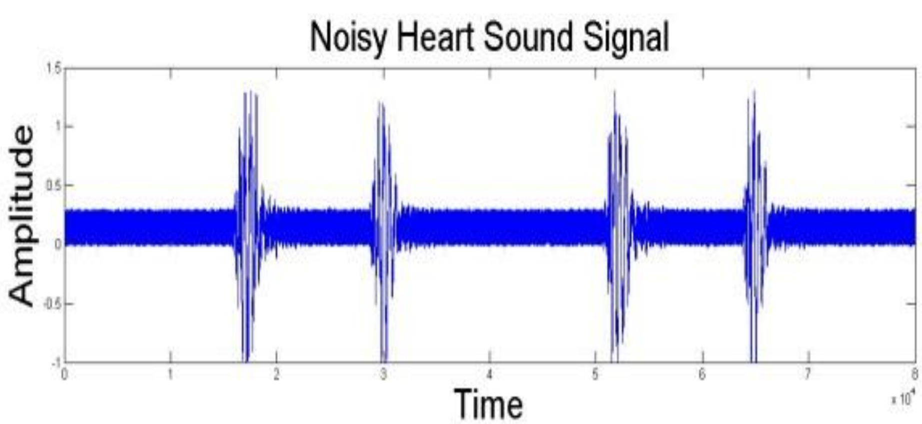

Figure 3.5(b)

Figure 3.5: (a) Original Heart Sound Signal, (b) Noisy Heart Sound Signal

Proposed Algorithm has been performed over the corrupted heart sound signal for a range of sampling frequency (5000-12000) Hz. Filtered heart sound signals obtained by the proposed algorithm for different sampling frequencies are then compared with the original signal. SNR and Correlation values of the filtered signals are shown in Table 3.2. From Table 3.2 it can be seen that the filtered signal obtained by the algorithm at sampling frequency of 7000 Hz has the highest SNR (Signal to Noise Ratio) value and the filtered signal obtained by the algorithm at sampling frequency of 8000 Hz has the highest Correlation value.

Table 3.2: Variation of SNR and Correlation values of Filtered Heart Sound signals with Sampling Frequency

Sampling frequency	**SNR**	**Correlation**
5000	10.4374	0.9327
6000	11.0880	0.9566
7000	**12.0018**	**0.9734**
8000	**11.3383**	**0.9979**
9000	8.0536	0.9975
10000	5.5888	0.9970
11000	3.6374	0.9965
12000	2.0305	0.9959

Variations of SNR and Correlation value of the filtered signals for different sampling frequencies are shown in Fig. 3.6 (a) & (b).

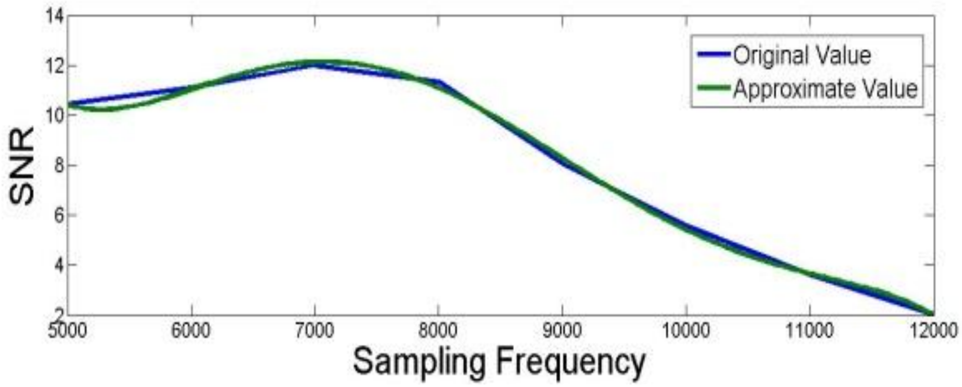

Figure 3.6(a)

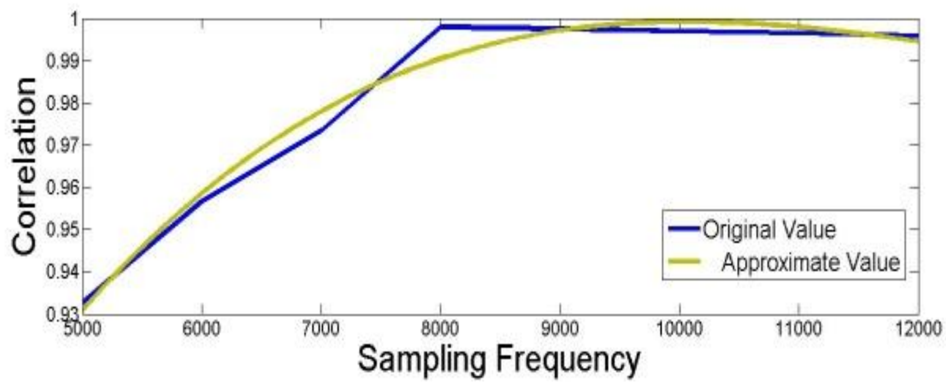

Figure 3.6(b)

Figure 3.6: (a) Plot of SNR-Sampling Frequency, (b) Plot of Correlation value-Sampling Frequency

Filtered heart sound signals obtained by the proposed algorithm with sampling frequency 7000 Hz and 8000 Hz have been shown in Fig. 3.7 (a) & (b).

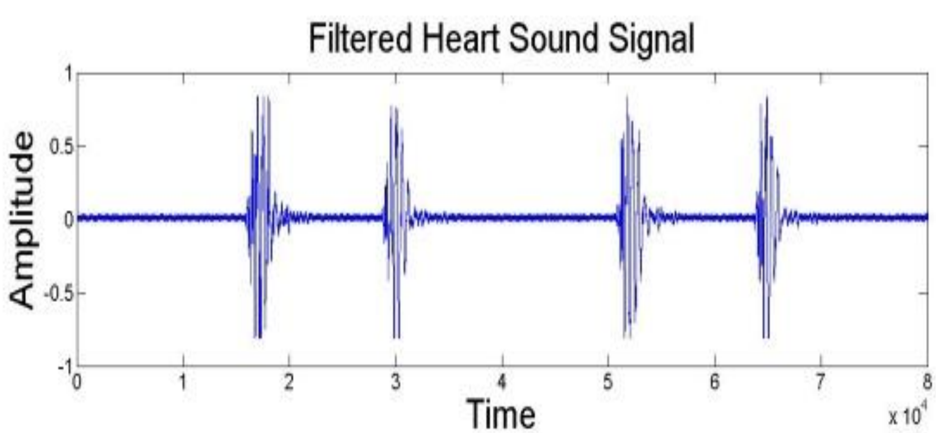

Fig 3.7(a)

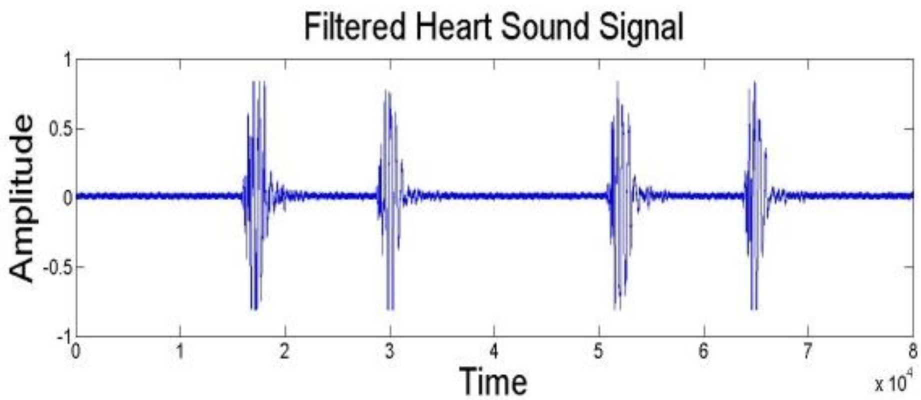

Fig 3.7(b)

Figure 3.7: (a) Filtered (Best offspring) Signal at Sampling Frequency 7000Hz, Filtered (Best offspring) Signal at Sampling Frequency 8000Hz

3.6. Ant weight Lifting Algorithm (AWL)

Colonies of ants (Family: Formicidae, Order: Hymenopetra (a-z-animals, 2018)) are gigantic with the largest size so far recording to 6000 km (scribol, 2018). Though an ant is very tiny in size (2mm to 25mm approximately), in recent times it has been observed that the neck joint of a common American field ant can withstand pressures up to 5,000 times of its own weight (entomologytoday, 2018). According to the bio-physicists smaller organisms have high strength to weight ratio a smuscle strength changes proportionally with the change in muscle cross-sectional area where as the mass of the organism changes proportionally with its volume (antweb, 2018.).According to Dr. Thomas Endlein ants are capable of changing the size and shape of the pads on their feet depending on the load they are carrying. At the time of carrying heavy loads they increase the contact area, and when they need to run they decrease it (ftexploring, 2018). In proposed approach an innovative algorithm namely Ant Weight Lifting (AWL) Algorithm has been introduced using the prospective of weight lifting capability of ants.

3.6.1. Idealized Rules

I. The entire input data set $X_{ij}[n]$ is considered as a food source where each column($j = 1\ to\ 5$) of the data set represents a type of food or parameters and row (i=1 to 200) represents values of the corresponding type of parameters.

II. The weight(a_ω) of each ant is assumed to be one unit i. e. $a_\omega \sim 1$.

III. Weightage(ω) of each cell of $[X_{ij}]$is fixed at 1000.

IV. An ant chooses only one food grain of each type j.

V. An ant can pick up 5 different types of food grains at most. Finally it leaves the food space after collecting total food grains weighing 5,000 times of its own weight i.e. total weight of picked up food grains ($\sum_{j=1}^{5} \omega_j$) will be equals to $5{,}000 \times a_\omega$.As $a_\omega = 1$, ,it is equal to 5000.

3.7. Ant Weight Lifting Algorithm for Window Parameter Optimization

Step1: A data set X_{ij} containing sets of filter parameters will be generated. Each columnof X_{ij}contains set of a specific parameter whereas j represents no. of columns. Here, $j = 5$.

Step 2: Consider a fixed number of ants, suppose t. Each ant a_p (p ranges from 1 to t) is considered to be capable of carrying at most 5 food grains at a time.

Step 3: Ants start collecting food grains (elements of data set) randomly from each column of the food space X_{ij}.

Step 4: Each time an ant a_p collects an element from a cell of X_{ij}, total weight of the visited cells by the ant will be increased by the specified weight (ω)of the cell.

Step 5: Each ant a_p will collect only one element from a column of X_{ij}. An ant will leave the food space and will deposit the collected grains (data set elements) to the Central Food Repository (CFR) after collecting 5 types of different grains from the 5 different columns of the food space and will not enter into the food space further.

Step 6: Digital Filter will be designed by the set of parameters collected by each ant. Filter will be used for filtration of a noisy signal$x'(n)$. φ Factorof the Filtered signal $z(n)$ will then be computed.

Step 7: Filters, which will yield the best value of φ factor, is reported to be the best filter of the iteration. Values of those parameters will be noted down as the optimized values of the parameters.

Step 8: Step 1 to Step 8 will be repeated until the considered number of iterations has been completed.

Step 9: In each of the iterations a best filter and a set of optimized parameters is reported and finally among all these filters the best one will be selected based on the criterion mentioned in Step 7.

Algorithm 3.4: *Filter parameter optimization using Ant Weight Lifting Algorithm*

3.7.1. Case Study

A Heart Sound Signal is collected from ISO 9001:2000 certified *Jeevan Rekha Diagnostic Centre*, India. The truncated signal has been mixed with random noise. Noisy signal is passed through Kaiser Window based proposed filter where optimization of the parameters are carried out using Ant-Weight Lifting Algorithm. Optimized set of parameters determines the order of the filter to be implemented by the Kaiser Window function. Result is compared with the Filter implemented using Chebyshev Window function (Dolph, 1946), Butterworth function (Butterworth, 1930) and Parks McClellan algorithm (McClellan & Parks, 1975) of the similar order for the same noisy heart sound signal. In Table 3.3, optimized set of parameters has been shown.

Table 3.3: Optimized Parameters

Name of Parameter	**Value**
Passband Ripple (δ_p)	0.2800
Stopband Ripple (δ_s)	0.4700
Passband Cut-off Frequency (f_p)	100 Hz
Stopband Cut-off Frequency (f)	1000Hz
Sampling Frequency (f_s)	71000Hz

In Figure 3.8 magnitude Response of the filter implemented using the optimized parameters has been shown.

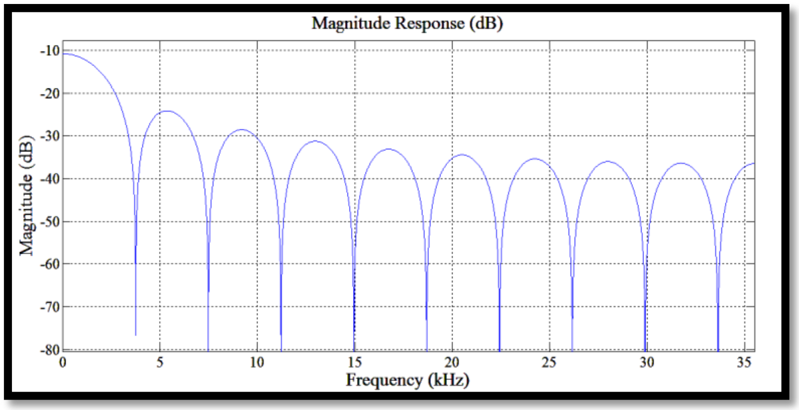

Figure 3.8: Magnitude Response of the filter implemented with optimized parameters

Signal to Noise Ratio (SNR) of the noisy signal = -4.2150, SNR and Correlation values of the signals filtered using different filters have been shown in the Table 3.4 and Table 3.5 respectively.

Table 3.4: Signal to Noise Ratio of the filtered signals

Filter	**SNR**
Filter implemented using Chebyshev Function (Dolph, 1946; Sharma, 2009)	-5.3591
Butterworth Filter (Butterworth, 1930)	-5.0629
Filter implemented using Parks McClellan Algorithm (Parks & McClellan, 1972)	-0.3291
Proposed Filter (using Kaiser Window Function and AWL Algorithm)	3.7297

Table 3.5: Correlation of the filtered signals

Filter	**Correlation**
Filter implemented using Chebyshev Function (Dolph, 1946; Sharma, 2009)	0.2621
Butterworth Filter (Butterworth, 1930)	0.3890
Filter implemented using Parks McClellan Algorithm (Parks & McClellan,	0.8705

1972)	
Proposed Filter (using Kaiser Window Function and AWL Algorithm)	0.9508

Comparing the Signal to Noise Ratio and Correlation values of optimized filtered signals, it has been proved that the optimized signal designed using AWL algorithm and Kaiser Window is least noisy. Figure 3.9 (a) and (b) shows original heart sound signal and Figure 3.9 (c) and (d) show signals filtered using conventional filtration technique of Parks McClellan Algorithm and the proposed technique.

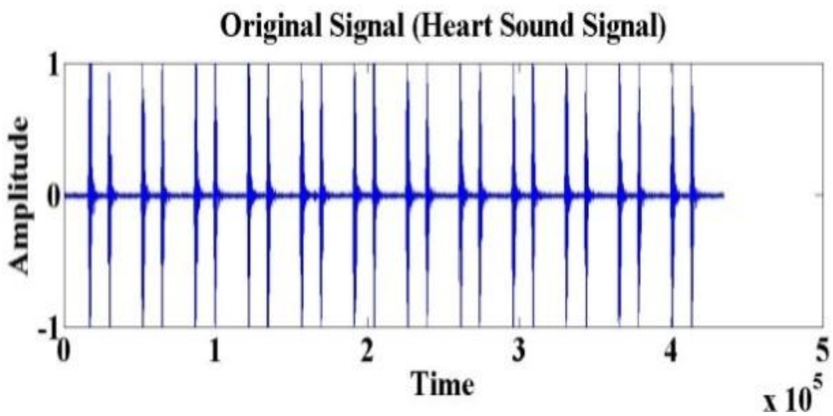

Figure 3.9(a)

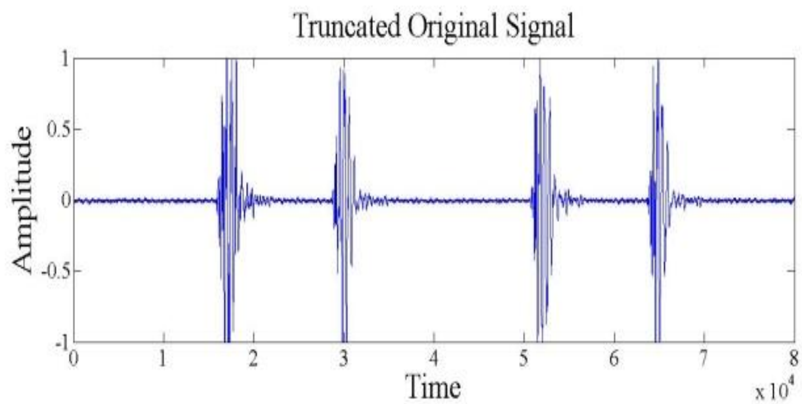

Figure 3.9(b)

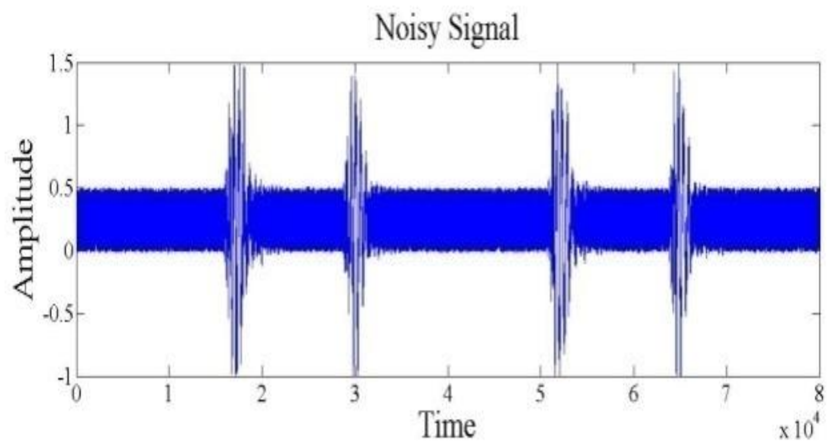

Figure 3.9(c)

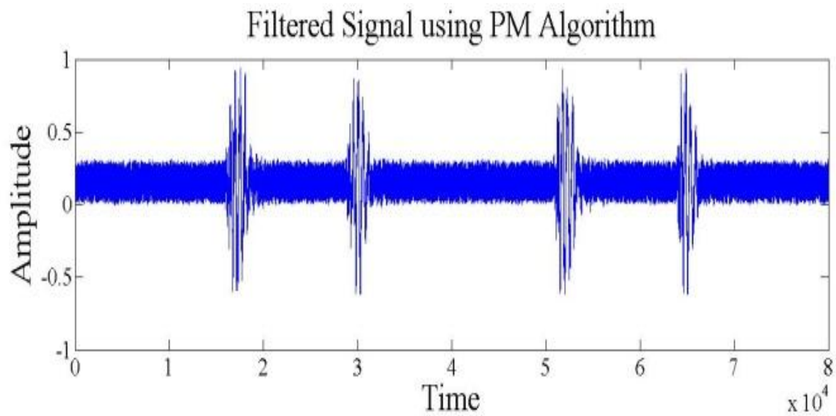

Figure 3.9(d)

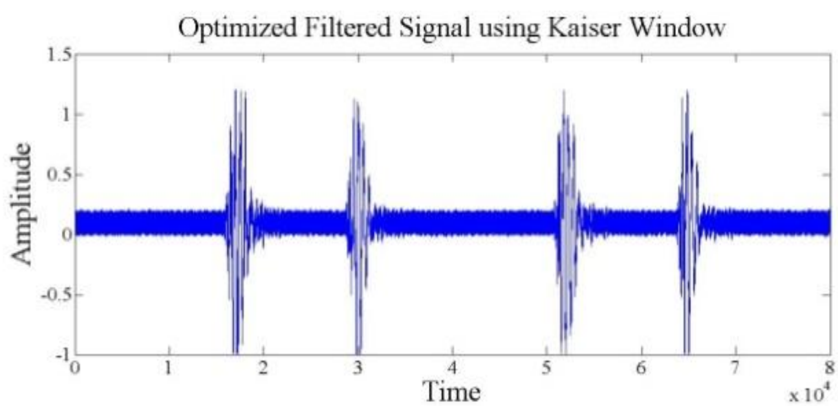

Figure 3.9(d)

Figure 3.9: (a) Original Heart Sound Signal, (b) Truncated Heart Sound Signal (c) Noisy Heart Sound Signal (d) Signal filtered by the filter designed using Parks McClellan Algorithm, (e) Signal filtered by the proposed filter

Feature analysis of the original signal, noisy signal and the optimized filterd signal is performed and shown in Table 3.6.

Table 3.6: Feature analysis of Original Signal, Noisy Signal and Filtered Signal

Features	Original Signal	Noisy Signal	Optimized Filtered Signal
Variance	0.0315	0.0522	0.0348
Skewness	-0.2021	-0.0572	-0.1456
Kurtosis	19.8973	8.9237	16.7892
Standard Deviation	0.1775	0.2286	0.1865
Maximum Peak Amplitude Value	1	1.4986	1.1998
Maximum Peak Amplitude Position Value	16966	51754	51799
Maximum Valley Amplitude Value	-1	-0.9997	-0.9994
Maximum Valley Amplitude Position Value	16706	65049	65049

3.8. Conclusions

In the past few decades, rapid growth of digitization and globalization has prejudiced the medical field as well. For improvement of diagnostic results and mutual availability of therapeutic case studies most of the fabled hospitals and diagnostic centers all over the world have started sharing medical information via different transmission media. At the time of transmission via any media, signals get affected by unwanted components; which are hostile but inescapable. In this chapter, two different techniques have been proposed for removal of noise from bio-medical signals (heart sound signal). In both the cases Kaiser Window function has been used for designing the digital FIR filters. At the first approach least noisy signal is obtained using Genetic Algorithm from a set of signals filtered using Kaiser Window with varying passband and stopband ripples. In the former approach for optimization of the parameters required to implement filter using Kaiser Window function, an innovative nature inspired algorithm based on the weight lifting strategy of ants is used. For optimization a simple adaptive objective function has been introduced that performs based on the ability of the designed filter with the optimized parameters to de-noise signals.

Chapter 4

Filter Coefficient Optimization

4.1. Introduction

Advancement in Digital Signal Processing enhanced the use of Digital Filters for removing noise from signals. Smaller physical dimension, higher reliability, and reduced sensitivity allow the digital filters to dominate over their analog counterparts. Based on impulse response characteristics digital filters are distinguished in following two types: (a) Finite Impulse Response (FIR) Filter (Salivahanan et al., 2007; Sharma, 2009; Mitra, 2013), (b) Infinite Impulse Response (IIR) Filter (Salivahanan et al., 2007; Sharma, 2009; Mitra, 2013; Singh & Arya, 2012; Karaboga & Cetinkaya, 2014). Due to minimalisms in hardware and fluently attainable linear phase properties, FIR filters are used massively compared to the IIR filters. Design of digital FIR filters involves calculation of filter transfer function coefficients that provide target frequency response (Sharma, 2009). The process of determining appropriate set of filter coefficients can be perfectly characterized as an optimization problem with an objective of minimizing the error function. This error function is conceptualized as an approximation function signifying the deviancy between the designed filter responses from the ideal filter responses. Based on this approach Parks and McClellan (Parks & McClellan, 1972) proposed an algorithm aiming to receive exact linear phase response. The algorithm does not permit independent selection of passband and stopband ripples where as it selects a ratio of passband and stopband ripples. In order to obtain the linear response likely to the ideal one, in this chapter different metaheuristic algorithms (Yang, 2014) have been used for obtaining optimized sets of filter coefficients.

In the ground of optimization algorithms a considerable number of metaheuristic algorithms (Storn, 1996; Storn & Price, 1997; Baghel et al., 2012; Kennedy & Eberhart, 1995; Dorigo & Caro; Yang, 2008; Yang, 2009; Yang, 2010b; Yang & Deb, 2009) have attracted researchers since past two decades by solving different challenging optimization problems in different fields as well. Behavior of biological systems and/or physical systems in nature stood as the motivation behind majority of the heuristic (Kokash, 2005; Aickelin & Clark, 2011) and metaheuristic algorithms (Holland, 1975; De Jong, 1975; Yang, 2014). Metaheuristic algorithms perform much better compared to heuristic algorithms to reach the global optima even in presence limited information about the problem.

In the initial part of this chapter BAT algorithm (Yang, 2010d) has been used to design both the lowpass and highpass filter. A comparative study of the performances of the filters designed by BAT Algorithm, GA and PSO are presented.

In the trailing part, design of even order low pass FIR filter and odd order bandpass FIR filter using the coefficients optimized by an innovative algorithm namely Global Best steered Cuckoo Search Algorithm (gbest CSA) has been proposed.

4.2. Problem Formulation

FIR filter of order N i.e. length M (M=N+1) with input $x(n)$ and output $y(n)$ can be represented as follows (Sharma, 2009):

$$y(n) = c_0 x(n) + c_1 x(n-1) + \cdots + c_{M-1} x(n-M+1)$$

$$= \sum_{k=0}^{M-1} c_k x(n-k) \quad where, \{c_k\} \text{ represents set of filter coefficients.} \qquad (4.1)$$

Alternatively, $y(n)$ can also be represented by Eqn. 4.2,

$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k)\ where\ y(n)$ is the convolution of the unit sample response $h(n)$ with the input signal $x(n)$. (4.2)

A FIR filter has linear phase if the unit sample response obeys the following Eqn. 4.3 (Sharma, 2009):

$$h(n) = \pm h(M-1-n), n = 0,1,\dots,M-1 \quad (4.3)$$

For symmetric $h(n)$, number of filter coefficients(d) must be as follows (Sharma, 2009):

$d = \frac{M+1}{2}$; when M is odd $d = \frac{M}{2}$; when M is even (4.4)

Based on the filter order and symmetricity of filter coefficients filters are categorized in four types: (a) Type I- Even order and symmetric coefficients, (b) Type II- odd and symmetric coefficients, (c) Type III- Even order and asymmetric coefficients, (d) Type IV- Odd order and asymmetric coefficients.

In traditional equiripple method for obtaining optimized set of filter coefficients, a desired filter response is estimated with varying degree of success of an objective function which minimizes the error between the approximated filter response and the desired filter response. This method uses an approach for minimizing errors in both the passband and stopband (Pei & Wang, 2002). Following error function, which computes the weighted difference of the ideal and approximated frequency responses in passband as well as in stopband, is used in most of the cases (Singh & Josan, 2014):

$$E(\omega) = W(\omega)[H_d(e^{j\omega}) - H_a(e^{j\omega})] \quad (4.5)$$

Where $H_d(e^{j\omega})$ and $H_a(e^{j\omega})$ are the desired or ideal and approximated frequency response of the filter. Ideal frequency response of a lowpass filter is stated as following (Sharma, 2009; Mitra, 2013):

$H_d(e^{j\omega}) = 1 \qquad for\ \ 0 \le \omega \le \omega_p$

$$= 0 \qquad for\ \ \omega_s \le \omega \le \pi \tag{4.6}$$

Ideal frequency response of a highpass filter is defined as following (Sharma, 2009; Mitra, 2013):

$H_d(e^{j\omega}) = 1 \qquad for\ 0 \le \omega \le \omega_s$

$$= 0 \qquad for\ \omega_p \le \omega \le \pi \tag{4.7}$$

For bandpass filter ideal frequency response can be stated by the following equation (Sharma, 2009; Mitra, 2013):

$H_d(e^{j\omega}) = 1 \quad for\ \omega_{p_1} \le \omega \le \omega_{p_2}$

$$= 0 \quad for\ 0 \le \omega \le \ \ \omega_{s_1} \&\ for\ \omega_{s_2} \le \omega \le \pi \tag{4.8}$$

ω_{p_1}, ω_{p_2} stand for lower passband edge frequency and upper passband frequency respectively. ω_{s_1}, ω_{s_2} are lower stopband frequency and higher stopband edge frequencies respectively.

The weighing factor $W(\omega)$ controls the minimization of error in both of the frequency bands. Using the key concept of the equiripple methodology Parks and McClellan proposed an efficient algorithm for optimal filter design. Best approximation of the Parks and McClellan algorithm is subjected to minimize the maximum bound of error $E(\omega)$. Only limitation of this strategy is the fixed value of $\frac{\delta_p}{\delta_s}$. Aiming to overcome this limitation in present work mean square error (Dhabal & Sengupta, 2015) based objective function has been adopted:

$$\varphi = \mu E_p + (1-\mu)E_s \qquad 0 < \mu < 1 \tag{4.9}$$

For lowpass filter Ep and Es are stated by the following two equations (Dhabal & Sengupta, 2015):

$$E_p = \frac{1}{\pi}\int_0^{\omega_p}(1 - H(\omega) - \delta_p)^2 d\omega \tag{4.10}$$

$$E_s = \frac{1}{\pi}\int_{\omega_s}^{\pi}(0 - H(\omega) - \delta_s)^2 d\omega \quad (4.11)$$

To implement highpass filter Ep and Es can be defined by the following two equations (Dhabal & Sengupta, 2015):

$$E_p = \frac{1}{\pi}\int\int_{\omega_p}^{\pi}\left(1 - H(\omega) - \delta_p\right)^2 d\omega \quad (4.12)$$

$$E_s = \frac{1}{\pi}\int_{0}^{\omega_s}(0 - H(\omega) - \delta_s)^2 d\omega \quad (4.13)$$

In case of designing bandpass filter Ep and Es are stated as following (Dhabal & Sengupta, 2015):

$$E_p = \frac{1}{\pi}\int_{\omega_{p_1}}^{\omega_{p_2}}(1 - H(\omega) - \delta_p)^2 d\omega \quad (4.14)$$

$$E_s = [\frac{1}{\pi}\int_{0}^{\omega_{s_1}}(0 - H(\omega) - \delta_s)^2 d\omega] + [\frac{1}{\pi}\int_{\omega_{s_2}}^{\pi}(0 - H(\omega) - \delta_s)^2 d\omega] \quad (4.15)$$

H is the magnitude response of the approximated filter. The objective function φ is the weighted sum of mean square errors $E_p$andE_s. By minimizing φ better performance can be obtained by the approximated filters.

4.3. Filter Coefficient Optimization using New Metaheuristic Algorithms

4.3.1. Filter Coefficient Optimization using BAT Algorithm

For optimizing filter coefficients of a lowpass filter using Bat Algorithm following inputs are taken into account: (a) Order of the filter (N) =20, (b) Passband edge frequency (ω_p) = 0.55, (c) Stopband edge frequency (ω_s) = 0.65, (d) Passband ripple(δ_p) = 0.1, (e) Stopband ripple(δ_s) = 0.01, (f) No. of Iteration (I) = 3000.

4.3.1.1. BAT Algorithm

Echolocation behavior of bat was inspiration of intending bat algorithm, it was first proposed by Yang in 2010 (Yang, 2010d). Term echolocation refers to the use of sound waves and produced echoes to trace the location of the objects. Expertise in finding victims and classifying different types of beetles even in complete darkness makes the echolocation ability of bats most enthralling. Bats are mainly categorized in two types: (a) Mega-bats and (b) Microbats based on their size. Among different types of bats of different size, microbats (size: 2.2 to 11cm) use echolocation most widely (Richardson, 2008; Richardson). They use a variety of sonar to elude obstacles, sense victim and trace their nestling chinks in the dark. These bats produce a flamboyant sound pulsation and snoop to the echo that bounces back from the surrounding stuffs. Depending on the species and the tactics they use for hunting, type of produced audio signal and resultant bandwidth also vary. Majority of microbats use petite, frequency-modulated signals, however others use constant-frequency signals for echolocation. Though bulk amount microbats emit frequencies ranges 25 kHz to 100 kHz, some species is capable of producing frequencies up to 150 kHz. Usually a microbat emits about 10 to 20 sound bursts per second each lasting for 5 to 20 ms, however at the time of hunting for prey the pulse emission rate may increase to 200 per second. Pulse emission rate typically increases based on the distance of the species from the victim. The travelling ranges of such short pulses are usually a few meters, subjected to the actual frequencies (Richardson, 2008). Loudness of an emitted pulse can be 110 dB. While searching for prey volume decreases between loudest to quietest based on the distance of the species from the prey. Microbats have a mystical capability to sense the distance, positioning, moving speed and even the category of the victim. Overall three dimensional scenario of the surrounding can be easily sensed by a microbat based on the time difference between emission and detection of the echo, the time difference between their two ears and variations of loudness. Basically the microbats use all its senses eyesight (usually very low), smell sense to detect prey efficiently. But only the echolocation behavior could be utilized to formulate in such a way so that it could be used to solve an optimization problem. For implementation of Bat algorithm based on the echolocation behavior of bats, following idealized rules are formulated based on the rules proposed by Yang (Yang, 2010d):

(a) All bats have advanced ability of echolocation to perceive distance from the preys.

(b) Bats are having some magical power to distinguish between their victim and neighbouring obstructions.

(c) For flying bats use random velocity V_i at position P_i with a fixed frequency f where maximum value for f will be denoted by f_{max} and minimum value for f will be denoted by f_{min}.

(d) Bats fly with varying wavelength λ. They spontaneously modify the wavelength λ of their emitted pulses and tune the pulse emission rate r depending on the propinquity of their target. r is any variable in the range 0 to 1.

(e) A frequency range $[f_{min}, f_{max}]$ corresponds to a fixed range of wavelength $[\lambda_{min}, \lambda_{max}]$. For the affluence of algorithm design it has been approximated that any wavelength can be used.

(f) Bats use varying parameter loudness, whereas the loudness can vary in different customs, maximum value of the loudness will be a large positive L_{max} and minimum value will be a constant value L_{min}.

Frequency (f_i) of a solution is computed using the following equation (Yang, 2010; Yang, 2014):

$$f_i = f_{min} + (f_{max} - f_{min})\alpha \quad (4.16)$$

α is a random number and $\alpha \in [0, 1]$

Velocity of a solution computed by the following equation:

$$v_i^{t+1} = v_i^t + (s_i^{t+1} - BEST)f_i \quad (4.17)$$

Generating a new solution is performed using the following equation:

$$s_i^{t+1} = s_i^t + v_i^{t+1} \quad (4.18)$$

A random value $r \in [0,1]$ is generated and checked if it is greater than the pulse rate r then newly generated solution will be adjusted using the following equation:

$$S_i = BEST + 0.01\beta \quad (4.19)$$

β is a randomly generated array of length d, $\beta_j \in [0, 1]$ j=1,2,….,d.

A typical BAT Algorithm can be described as follows (Yang, 2014):

Input: Total No. of Iteration ($iter_{MAX}$), size of the population (n), objective function (f), lower bound, upper bound
Output: Global Best Solution.
Begin
Define objective function $f(x), x = (x_1, x_2, \ldots, x_d)$
Initialize the bat population $S_i (i = 1,2, \ldots, n)$, frequency (f_i), pulse rate (r_i), loudness (A_i)
Evaluate the fitness of the solutions using f and store in an array $fitness$
Find the best quality solution and store in $BEST$; Store fitness value of $BEST$ in $fmin$
$t = 1$
while($t < iter_{MAX}$)
    for $i = 1: N$
        Generate new solution by adjusting frequency
        Update velocities and solutions using Equations 4.16 to 4.18
        Generate a random number r_1
        if($r_1 > r_i$)
            Select a solution among the best solutions
            Generate a local solution around the selected best solutions using Eqn. 4.19
        end if
        Evaluate the fitness of the new solutions using f and store in $fnew$
        if($fnew < fitness(i)$)
            Update old solution with new one; $fitness(i) = fnew$
        end if
    end for
    Evaluate the fitness of the solutions using f
    Find the current best nest c_{best} and store its fitness value in $cmin$
    if ($cmin < fmin$)
        Update $BEST$ and $fmin$
    End if
    $t = t + 1$
End while
End

Algorithm 4.1: *BAT Algorithm*

4.3.1.2. Simulation Results

Passband ripple and stopband attenuation of the filters implemented using the optimized filter coefficients are shown in Table 4.1.

Table 4.1: Comparison of filter responses

Algorithms	**Passband Ripple**	**Stopband Attenuation**
BAT	**0.10**	**31.4**

GA	1.0486	22.2
PSO	0.06	25.2
PM	0.1098	24.9

Tables 4.1 summarized the performances of BAT Algorithm, GA, PSO, and Parks McClellan for implementing the low pass FIR filter. It can be observed that the stop-band attenuation for 20th order low pass filter using BAT Algorithm is 31.4 dB while the same using GA, PSO and Parks-McClellan, are 22.2 dB, 25.2 dB and 24.9 dB. The normalized stop-band ripple obtained using BAT Algorithm is 0.03. This proves superiority in performance of the BAT Algorithm at the cost of little increase in pass-band ripple compared to PSO (0.10 vs. 0.06). The ripple performances are improved by 3.00%, 14.27% and 3.14% compared to PSO, GA and PM respectively in stopband.

Table 4.2 shows optimized sets of filter coefficients obtained using different algorithms.

Table 4.2: Optimized Filter Coefficients

Algorithm	BAT	GA	PSO	PM
h(1) = h(21)	-0.0099625	-0.0102313	-0.0139	-0.0389
h(2) = h(20)	0.0022065	0.0016687	-0.0012	0.0026
h(3) = h(19)	0.012395	0.0241687	0.0254	0.0302
h(4) =h(18)	-0.0079118	0.04558075	-0.0078	-0.0181
h(5) = h(17)	-0.031176	-0.0351313	-0.0344	-0.0356
h(6) = h(16)	0.036761	-0.0538313	0.0343	0.0393
h(7) = h(15)	0.056262	-0.0378313	0.048	0.0451
h(8) = h(14)	-0.08271	0.0371687	-0.0839	-0.0924
h(9) = h(13)	-0.062742	0.1549687	-0.05	-0.0471
h(10)= h(12)	0.30795	0.2514687	0.3139	0.3119
h(11)	0.55297	0.2925687	0.532	0.5512

Figure 4.1 shows the magnitude response in dB vs. normalized frequency and phase response vs. normalized frequency plot of the filter.

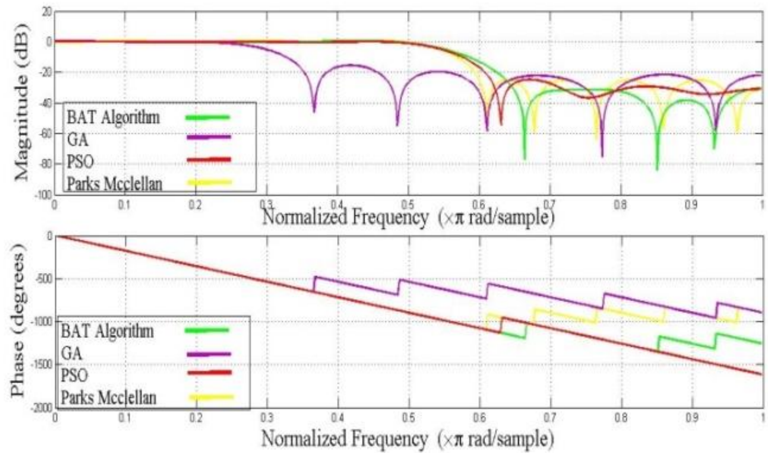

Figure 4.1: Magnitude Response in dB and Phase Response

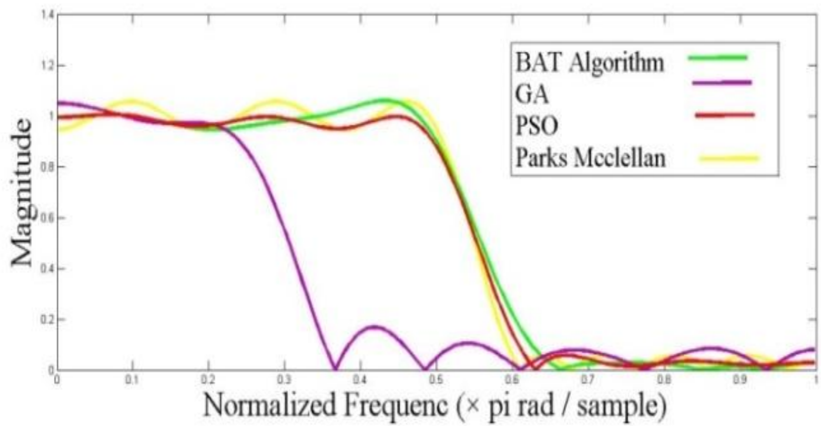

Figure 4.2: Magnitude Response

4.3.2. FIR Filter Design using Global Best Steered Cuckoo Search Algorithm

The present work includes design of odd order symmetric lowpass and bandpass FIR filters using a new algorithm namely Global Best Steered Cuckoo Search Algorithm (gbest CSA). We compared the results of the proposed system with the results of the standard Cuckoo Search Algorithm and also with the results of conventional algorithmic strategy of Parks

McClellan Algorithm. To design 21st order lowpass filter, total numbers of iterations are fixed at 1000. Passband edge frequency (ω_p) is taken as 0.4π and Stopband edge frequency (ω_s) is taken as 0.5π. For implementation of 25th order bandpass filter, total number of iterations is fixed at 4000. Passband edge frequency (ω_p) is taken as 0.3π and Stopband edge frequency is taken as (ω_s) 0.6π. For both types of lowpass and bandpass filters, Passband ripple (δ_p) is fixed at 0.05 and Stopband ripple is fixed at (δ_s) 0.03.

4.3.2.1. Cuckoo Search

Cuckoo Search Algorithm (Yang & Deb, 2009) is a modern nature inspired metaheuristic algorithm that has been broadly used for solving hard-hitting optimization problems. CS is based on the brood parasitism of the cuckoo species. It also uses a balanced composition of a local random walk and global explorative random walks, controlled by a switching parameter p_a (Yang & Deb, 2009; Yang, 2014). The local random walk can be defined by following Eqn.

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \epsilon) \otimes (x_j^t - x_k^t) \tag{4.20}$$

where, x_j^t and x_k^t are two different candidate solutions selected by random permutation, $H(u)$ stands for Heaviside function, α is step size scaling factor, ϵ denotes a random number drawn from a uniform distribution, p_a is probability of abandoning worst nests and s is the step size. Here, $\otimes$ stands for the entry-wise product of two vectors. Global random walk is carried out by a superior kind of random walk namely Lévy Flights. Predefined parameter bounds state the domain to choose the initial population.

4.3.2.1.1. Cuckoo Breeding Behaviour

Some cuckoo species like Ani and Guria cuckoos lay eggs in communal nests (Yarizadeh-Beneh et al., 2016). Sometimes they destroy host birds' eggs to enhance the probability of hatching their eggs. Thus they involve the host birds into rearing their progenies and dedicate more time in the process to lay more eggs instead of devoting time and energy in parental care. Host birds are either other individuals of same species or some other species. If a cuckoo chooses nest of another individual of same species to lay eggs then it is called "*Intra specific Brood Parasitism*" (Payene et al., 2005). If host birds become successful in identifying any egg as not their own, they either simply abandon the egg or moves away from the nest to build a new nest elsewhere.

4.3.2.1.2. Lévy Flights

Lévy flights belong to a class of random walks formulated by Paul Lévy in 1937 by generalizing Brownian motion and comprising non-Gaussian randomly dispersed step sizes for the distance covered (Yang & Deb, 2009; Yang & Deb, 2010). Lévy Flights are capable of maximizing the probability of resource searches in uncertain surroundings. In Optical science, Lévy flight can be defined as a term used to designate the motion of light. Survey says, by performing Levy flights vaster area can be covered than normal random search. Some shark species follow random Brownian motion while searching food; however, if they failed to get food items, they start following Lévy flight behavior, mixing short random movements with long trajectories. Lévy flight is defined by the following Eqn.

$$x_i^{t+1} = x_t^i \alpha L(s, \lambda) \tag{4.21}$$

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, (s \gg s_0 > 0);\ \alpha > 0 \text{ is the step size scaling factor.} \tag{4.22}$$

4.3.2.1.3. Cuckoo Search via Lévy Flights

Three idealized rules have been implicated by Yang.et.al to simplify the Cuckoo Search Algorithm (Yang & Deb, 2009; Yang & Deb, 2010):

i. Each cuckoo lays a single egg at a time and dumps it in a communal nest which is chosen by them randomly.

ii. The best nests containing high quality eggs will be carried over to the succeeding generation.

iii. The number of available host nests is always fixed. Probability of an egg (laid by a cuckoo) to be discovered by the host bird is $p_a \in (0,1)$. In that case, the host bird can either abandon the egg or simply destroy the nest and build a new nest.

For a minimization problem, the quality or fitness of a solution can be reversely proportional to the objective function value. In another practice, fitness functions can be defined following the same way used in genetic algorithms where the tactic "fittest chromosome (solution) survives" is used. In Cuckoo Search Algorithm each possible solution to the problem is assumed as an egg in a nest. A new solution is assumed as a newly hatched cuckoo egg. The aim is to replace worse solutions in the nests by new and potentially better solutions (cuckoo eggs). A new solution x_i^{t+1} is generated by selecting a cuckoo i using Lévy flights. Following equation described generation of new solution.

$$x_i^{t+1} = x_i^t + \alpha \oplus \text{Lévy}(\lambda) \tag{4.23}$$

Where $\alpha > 0.\alpha$ is the step size, value of α is problem specific. In most of the cases, $\alpha = 1$ is used. The Lévy flight basically provides a random walk where the random step length is drawn from Lévy distribution defined by the following equation: Lévy$\sim u = t^{-\lambda}$ $(1 < \boldsymbol{\lambda} \leq \mathbf{3})$ having an infinite variance along with an infinite mean.

Cuckoo Search Algorithm is detailed in Algorithm 4.2.

Input: Total No. of Iterations (*Max_Iteration*), size of population (n), objective function (f), lower bound, upper bound
Output: Global Best Solution
begin
Initialize a population of n nests randomly within bound
Quality of each solution is evaluated using the objective function f & store best quality solution in $BEST$
Initialize a probability $p_a = 0.25$ for discovering worse quality nests & initialize λ=1.5
Initialize t = 1
while $(t < Max\_Iteration)$
  for $i = 1{:}n$
    Generate new nest using Lévy Flights defined in Eqn. 4.22 & evaluate the quality of the new nest
    if $\left(f_i^{t+1} < f_i^t\right)$
      Replace the old nest by the new one
      (f_i^{t+1}represents fitness value of nest i at iteration $t + 1$ & f_i^t represents the same at iteration t)
    End if
  End for
  Find the current best nest c_{best}
  if $(c_{best} < BEST)$
    Update $BEST$
  End if
  Discover worse quality nests with probability p_a & generate new nests using the Eqn. 4.19
  Compute the fitness of the new nests & replace the worse nests by the new nests with better quality
  Find the current best nest c_{best}
  if $(c_{best} < BEST)$
    Update $BEST$
  End if
  $t = t + 1$
End while
End

Algorithm 4.2: *Cuckoo Search Algorithm*

4.3.2.2.Global Best Steered Cuckoo Search Algorithm (gbest CSA)

In Global Best Steered Cuckoo Search Algorithm (gbest CSA) a modification was performed in replacement strategy (Dhabal & Venkateswaran, 2017); another modification was in choice of a parameter (λ) in Lévy Distribution; one more modification was in choosing the probability of abandoning nests(p_a).

In reality higher similarity of a cuckoo egg with the host's eggs decreases the probability of the cuckoo egg to be distinguished by the host from its own eggs. As the fitness of a candidate solution termed as cuckoo egg is related to its difference with the solutions of the latest iteration, therefore, it will be better to perform a random walk in a biased way with random step sizes. Similar like Cuckoo search, Global best steered cuckoo search also uses random step sizes but with different function set for computing the step size. In Cuckoo Search Algorithm, step size is computed using 4.18. For gbest CSA step size is computed using the following Eqn.

$$r * nest[[best] - nest\big[permute[i]\big][j] \tag{4.24}$$

Where r is a random number lying within the range [0, 1].*nest* denotes a matrix containing candidate solutions along with their variables, *permute* represents row permutation functions applied on *nest* matrix (Tuba et al., 2012). In case of gbest CSA instead of using another permute function, best nest till the latest iteration is used. This specialty of gbest CSA keeps the selection pressure towards the better solutions, hence assures better result. Moreover, this advancement of the Cuckoo Search Algorithm does not flood the population by the high fitness solutions. In gbest CSA instead of using $\lambda = 1.5$, λ is determined by the following Eqn (Dhabal & Venkateswaran, 2017):

$$\lambda = (\lambda_{max} - \lambda_{min}) \times \frac{(iter_{total} - iter)}{iter_{total}} + \lambda_{min}\ where\ \lambda_{max} = 1.5 \text{ and } \lambda_{min} = 1 \tag{4.25}$$

In case abandoning worst quality nests standard CSA uses fixed probability of 0.25 (Yang & Deb, 2009; Yang, 2014).

In gbest CSA, following equation is used for obtaining the probability of abandoning worst nests:

$$p_a = p_a + rand/D;\ \text{Initially } p_a \text{ is considered as 0.25.} \tag{4.26}$$

These modifications exhibits better filter responses compared to the standard CSA.

Input: Total No. of Iteration (*Max_Iteration*), size of population (n), objective function (f), lower bound, upper bound
Output: Global Best Solution
begin
Initialize a population of n nests randomly within bound
Level the nests in terms of quality evaluated using the objective function f
Find the best quality nest and store in $BEST$
Initialize a probability p_a for discovering worse quality nests
Initialize t = 1
while ($t < Max\_Iteration$)
    for $i = 1{:}n$
        Compute λ computed by Eqn. 4.25
        Generate new nest using Lévy Flights defined in Eqn. 4.23
        Evaluate the quality of the new nest by the objective function
        if $\left(f_i^{t+1} < f_i^t\right)$
            Replace the old nest by the new one
            (f_i^{t+1}represents fitness value of new nest i at iteration $t+1$ & f_i^t represents fitness value of old nest i at iteration t
        End if
    End for
        Find the current best nest c_{best}
        if ($c_{best} < BEST$)
            Update $BEST$
        End if
        Discover worse quality nests with probability p_a
        Abandon worse nests and generate new nests using the Eqn. 4.24
        Compute the fitness of the new nests using the objective function
        Replace the worse nests by the new nests with better quality
        Update p_a using Eqn. 4.26
        Find the current best nest c_{best}
        if ($c_{best} < BEST$)
            Update $BEST$
        End if
        $t = t + 1$
End while
End

Algorithm 4.3: *Global Best Steered Cuckoo Search Algorithm (gbest CSA)*

4.3.2.3.Simulation Results

Figure 4.3 and 4.4 show the responses of lowpass Type II FIR Filters implemented using gbest CSA, CSA and PMA.

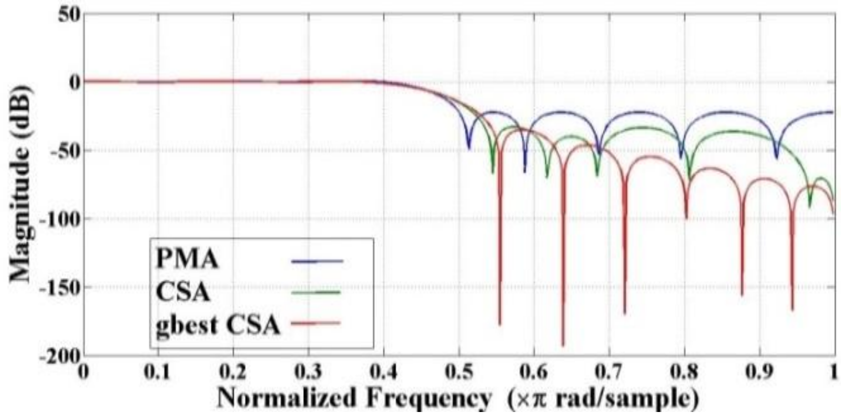

Figure 4.3: Normalized Frequency vs. Magnitude in dB of Lowpass Type II FIR Filter

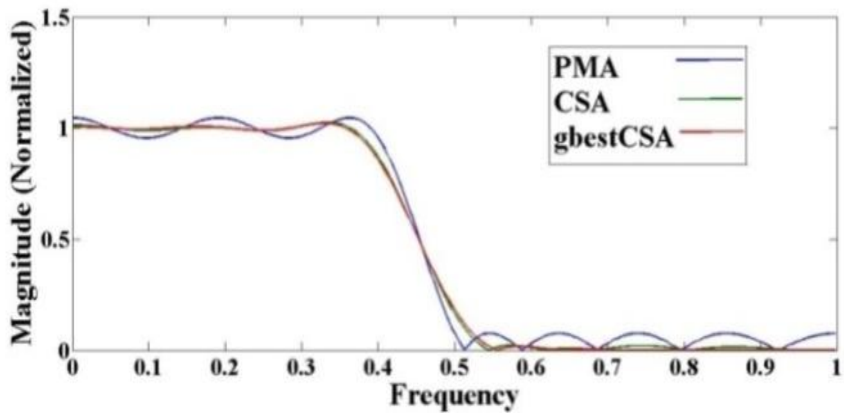

Figure 4.4: Frequency vs. Normalized Magnitude of Lowpass Type II FIR Filter

Similarly Figure 4.5 and Figure 4.6 show the responses of bandpass Type II FIR Filter implemented using gbest CSA, CSA and PMA.

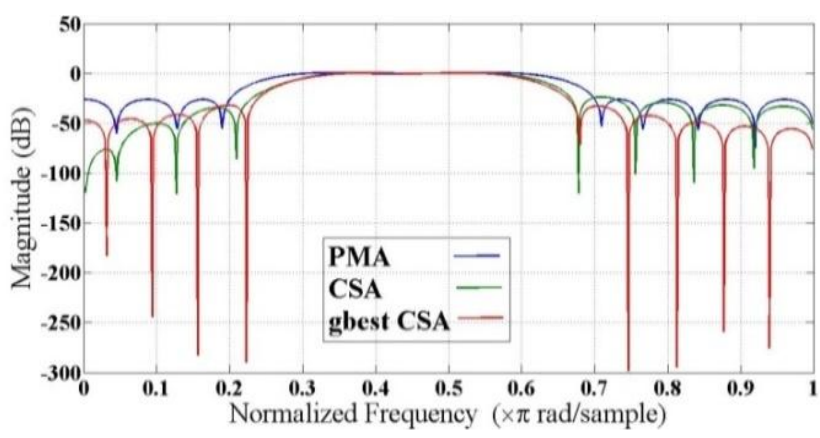

Figure 4.5: Normalized Frequency vs. Magnitude in dB of Bandpass Type II FIR Filter

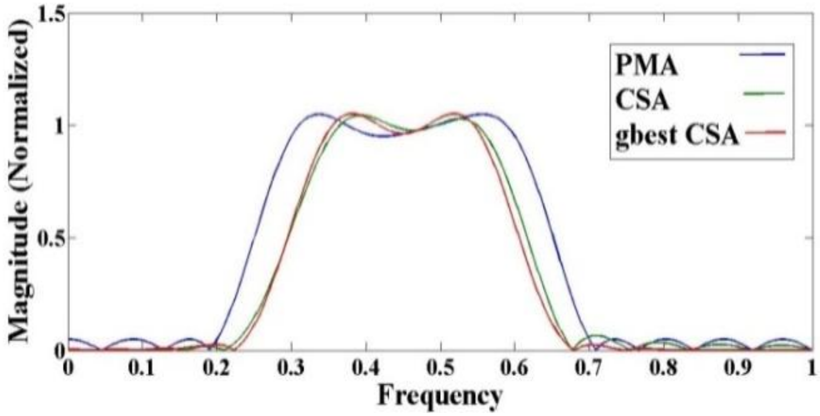

Figure 4.6: Frequency vs. Normalized Magnitude of Bandpass Type II FIR Filter

Table 4.3 and Table 4.4 show the comparative results of the Type II lowpass filters and Type II bandpass filters, respectively, in terms of passband ripple and also stopband attenuation.

Table 4.3: Comparison of Type II Lowpass filter responses

Algorithms	Passband Ripple	Stopband Attenuation
PM	0.11	25.6738
CS	0.14	33.4493
gbest CS	0.14	35.4566

Table 4.4: Comparison of Type II Bandpass filter responses

Algorithms	Passband Ripple	Stopband Attenuation
PM	1.04	26.5911
CS	1.04	26.1504
gbest CS	1.05	33.2805

Table 4.5 and Table 4.6 show the optimized coefficients of the Type II lowpass filters and Type II bandpass filters, respectively.

Table 4.5: Optimized Coefficients for Type II Lowpass Filter

Coefficients	PMA	CSA	gbest CSA
h(1) = h(22)	0.0168746929652027	0.00450653302996857	0.00475128080146216
h(2) = h(21)	0.0356224310492992	0.0120498836574256	0.00797044714101550

h(3) = h(20)	-0.022156462421912	-0.0085544060336254	-0.0073030927757500
h(4) = h(19)	-0.027740395668699	-0.0237797886423163	-0.0216310263118251
h(5) = h(18)	0.00836919266504365	0.00845929736049457	0.00515781230658275
h(6) = h(17)	0.0514379005698228	0.0433690256212844	0.0427007916209370
h(7) = h(16)	0.00476715377763214	0.00945993358253513	0.00742551715593949
h(8) = h(15)	-0.0840368108939566	-0.0798672291567218	-0.0779638435113317
h(9) = h(14)	-0.0474310194466021	-0.0492676037318842	-0.0489386676900673
h(10) = h(13)	0.179330187119499	0.176445843218714	0.175486596403889
h(11) = h(12)	0.412951663757719	0.414121107386855	0.415050096328490

Table 4.6: Optimized Coefficients for Type II Bandpass Filter

Coefficients	PMA	CSA	gbest CSA
h(1) = h(26)	0.0120473066682598	0.0244983137005205	0.00213779222996536
h(2) = h(25)	-0.024641695172901	-0.0177667680011737	-0.0150746230332790
h(3) = h(24)	-0.00706697543646	-0.0432125105296717	-0.0207567965513314
h(4) = h(23)	-0.017244142843961	0.0136902047811129	0.0279765100823963
h(5) = h(22)	-0.0435225995555	0.0110915911276413	0.0398014194638810
h(6) = h(21)	0.0283656122926086	-0.00345537332826511	-0.0129296523374115
h(7) = h(20)	0.0684688094178119	0.0975801731046881	-0.0050789025396310
h(8) = h(19)	-0.004395099347699	0.0477530362920501	-0.0033011712771962
h(9) = h(18)	0.0371415618982372	-0.215581707983335	-0.103467507414961
h(10) = h(17)	0.0344294788509787	-0.155035354335177	-0.0405896706941112

h(11) = h(16)	-0.228061298530215	0.240439572317793	0.216558804718472
h(12) = h(15)	-0.179254394516737	0.240439572317793	0.148925047855674
h(13) = h(14)	0.299278656698308	-0.155035354335177	-0.236423535445386

Results prove the efficacy of the adaptive technique of using gbest CSA and the mean square error based objective function for optimizing filter coefficients. Proposed method outperforms conventional technique of using Parks McClellan Algorithm for obtaining optimized coefficients set of Type II lowpass and bandpass FIR filters. Efficiency of the proposed algorithm compared to the conventional CSA in terms of optimized results (Best), mean, and standard deviation (Std.) is proved using seven standard benchmark functions (Jamil & Yang, 2013; Naik et al., 2015) and shown in Table 4.7.

Table 4.7: Performance Evaluation of gbest CSA and CSA using Benchmark Functions

Functions		Best	Mean	Std.
Sphere model (Jamil & Yang, 2013; Naik et al., 2015)	**gbest CSA**	6.0760e-15	4.6707e-13	4.9057e-13
	CSA	5.1925e-11	3.6508e-10	3.6669e-10
Schwefel's problem 2.22 (Jamil & Yang, 2013; Naik et al., 2015)	**gbest CSA**	6.4948e-08	3.7832e-07	2.2497e-07
	CSA	2.4363e-05	6.0125e-05	2.6264e-05
Schwefel's problem 1.2 (Electric Power Systems Analysis and Nature Inspired Optimization Algorithms, 2015a;	**gbest CSA**	0.0506	0.4197	0.3163
	CSA	2.5935	14.0311	10.6556

Naik et al., 2015)				
Schwefel's problem 2.21 (Jamil & Yang, 2013; Naik et al., 2015)	**gbest CSA** **CSA**	0.0425 3.7385	0.1199 6.5092	0.0549 1.6751
Generalized Rosenbrock's function (Naik et al., 2015)	**gbest CSA** **CSA**	0.3037 16.1695	24.7142 33.8772	18.1265 24.8391
Generalized Rastrigin's function (Surjanovik & Bingham, 2013; Naik et al., 2015)	**gbest CSA** **CSA**	16.6182 29.5893	33.4733 53.3390	9.6165 11.2181
Ackley's function (Surjanovik & Bingham, 2013; Naik et al., 2015)	**gbest CSA** **CSA**	9.6237e-08 3.2896e-04	4.3725e-06 0.4842	7.5285e-06 0.6344

4.3.3. FIR Filter Design using Fast Converging Cuckoo Search Algorithm

Present study involves design of odd length and even length symmetric lowpass, highpass and bandpass FIR filters using Fast Converging Cuckoo Search Algorithm (FCCSA). Responses of the implemented filters are compared with the responses of the filters designed by the Parks McClellan Algorithm and the conventional Cuckoo Search Algorithm.

In FCCSA step size is computed using the Eqn. 4.23. This modification keeps the selection pressure towards the better solutions, hence assures better results in lesser execution time. FCCSA is detailed in Algorithm 4.4.

4.3.3.1.Fast Converging Cuckoo Search Algorithm

Input: Total No. of Iteration (*Max_Iteration*), size of population (n), objective function (f), lower bound, upper bound
Output: Optimized set of filter coefficients.
begin
Initialize a population of n nests randomly within bound
Evaluate the quality of the nests using the objective function f and store the fitness values in fit
Find the best quality nest and store in $BEST$
Initialize a probability p_a for discovering worse quality nests;
Initialize t = 1
while $(t < Max\_Iteration)$
 for $i = 1{:}n$
  Generate new nest using Lévy Flights defined in Eqn. 4.23
  Evaluate the quality of the new nest by the objective function
  if $\left(fit_i^{t+1} < fit_i^t\right)$
   Replace the old nest by the new one
  End if
 End for
  Find the current best nest c_{best}
  if $(c_{best} < BEST)$
   Update $BEST$
  End if
  Discover worse quality nests with probability p_a
  Abandon worse nests and generate new nests using the Eqn. 4.24
  Compute the fitness of the new nests using the objective function
  Replace the worse nests by the new nests with better quality
  Find the current best nest c_{best}
  if $(c_{best} < BEST)$
   Update $BEST$
  End if
  $t = t + 1$
End while

Algorithm 4.4: *Fast Converging Cuckoo Search Algorithm (FCCSA)*

4.3.3.2.Simulation Results

Required filter parameters and input parameters to the algorithm used to implement Type I and Type II lowpass, highpass and bandpass filter are stated in Table 4.8, Table 4.9 and Table 4.10 respectively. In the following case studies responses of the implemented filters are shown.

Table 4.8: Input Parameters to design Type I and Type II Lowpass Filter

Type of filter	I	II
Total number of nests (n)	30	30
Filter order (N)	20	19
Size of each nest (candidate solution)	11	10
Total no. of iteration (MaxIteration)	1000	1000
Upper bound (U_B)	-1	-1
Lower bound(L_B)	1	1
Passband frequency (ω_p)	0.25	0.25
Stopband frequency (ω_s)	0.4	0.4
Passband ripple (δ_p)	0.0575	0.0575
Stopband ripple (δ_s)	0.0316	0.0316

Table 4.9: Input Parameters to design Type I and Type II Highpass Filter

Parameter Name	**Parameter Value**	**Parameter Value**
Type of filter	I	II
Total number of nests (n)	30	30
Filter order (N)	20	19
Size of each nest (candidate solution)	11	10
Total no. of iteration (MaxIteration)	1000	1000
Upper bound (U_B)	-1	-1
Lower bound(L_B)	1	1
Passband frequency (ω_p)	0.35	0.35
Stopband frequency (ω_s)	0.20	0.20
Passband ripple (δ_p)	0.1	0.1
Stopband ripple (δ_s)	0.05	0.05

Table 4.10: Input Parameters to design Type I and Type II Bandpass Filter

Parameter Name	Parameter Value	Parameter Value
Type of filter	I	II
Total number of nests (n)	30	30
Filter order (N)	20	19
Size of each nest (candidate solution)	11	10
Total no. of iteration (MaxIteration)	1000	1000
Upper bound (U_B)	-1	-1
Lower bound(L_B)	1	1
Passband frequency (ω_p)	0.3 ; 0.5	0.3 ; 0.5
Stopband frequency (ω_s)	0.25 ; 0.55	0.25 ; 0.55
Passband ripple (δ_p)	0.1	0.1
Stopband ripple (δ_s)	0.01	0.01

Figure 4.7 (a) shows plot of normalized Frequency vs. magnitude in dB and Figure 4.7 (b) shows plot of frequency vs. normalized magnitude of lowpass Type I FIR Filters implemented using FCCSA, CSA and PMA. Table IV contains comparative results of the filters in terms of passband ripple, stopband ripple and stopband attenuation. Table 4.8 contains optimized coefficients of the filters.

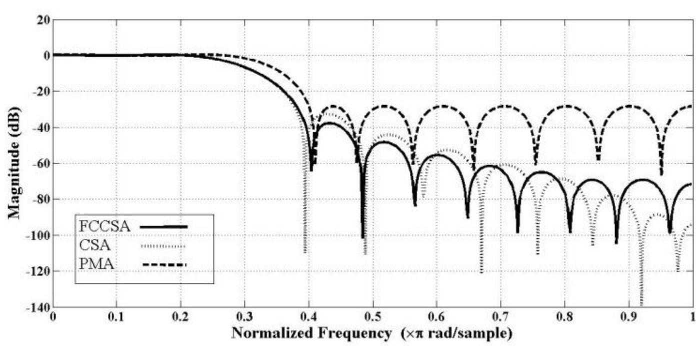

Figure 4.7(a)

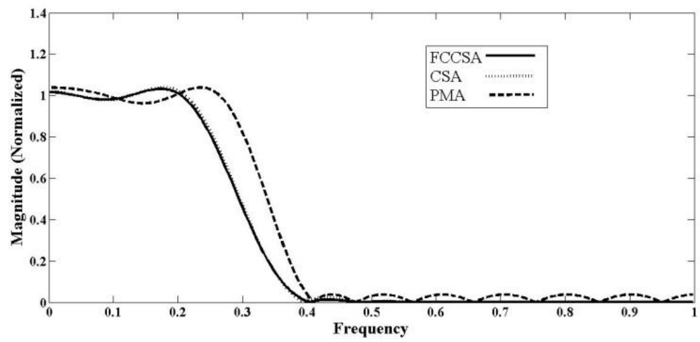

Figure 4.7(b)

Figure 4.7: (a) Normalized Frequency vs. Magnitude in dB of Lowpass Type I FIR Filter, (b) Frequency vs. Normalized Magnitude of Lowpass Type I FIR Filter

Table 4.11: Passband Ripple and Stopband Attenuation of Lowpass Type I FIR Filter

Algorithm	Passband Ripple	Stopband Ripple	Stopband Attenuation
PMA	0.0719	0.038	-28.87
CSA	0.0628	0.0232	-32.8578
FCCSA	0.0535	0.0128	-37.85

Table 4.12: Optimized Coefficients of Lowpass Type I FIR Filter

Coefficients	Parks McClellan	CSA	FCCSA
h(1) = h(21)	-0.0138732222373371	0.00339674726820915	0.00375547208188730
h(2) = h(20)	0.0159878573458010	0.0130678311138362	0.0122307398677153
h(3) = h(19)	0.0282770406935350	0.0190220208204059	0.0168130043548852
h(4) = h(18)	0.0225889779640064	0.00672641652179830	0.00495801361885702
h(5) = h(17)	-0.0107974229093833	-0.0245172923945074	-0.0241664826671393

h(6) = h(16)	-0.0508639686377598	-0.0517848590918863	-0.0495093765178696
h(7) = h(15)	-0.0535763648648696	-0.0396666696661371	-0.0372314416561971
h(8) = h(14)	0.0143400172180110	0.0327142756484805	0.0333347347450519
h(9) = h(13)	0.140903161291734	0.148094705278335	0.146754169973728
h(10) = h(12)	0.265901718602978	0.254461223430382	0.252951053281185
h(11)	0.318133616052643	0.297663846691431	0.296341702935941

Figure 4.8(a) shows plot of normalized Frequency vs. magnitude and Figure 4.8(b) shows plot of frequency vs. normalized magnitude of lowpass Type II FIR Filters implemented using FCCSA, CSA and PMA. Table 4.13 contains comparative results of the filters in terms of passband ripple, stopband ripple and stopband attenuation. Table 4.14 contains optimized coefficients of the filters.

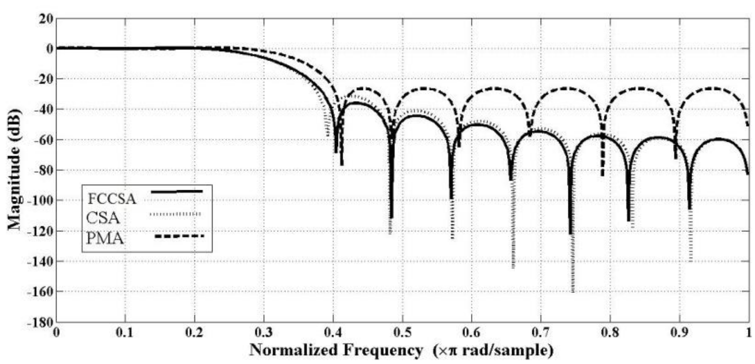

Figure 4.8(a)

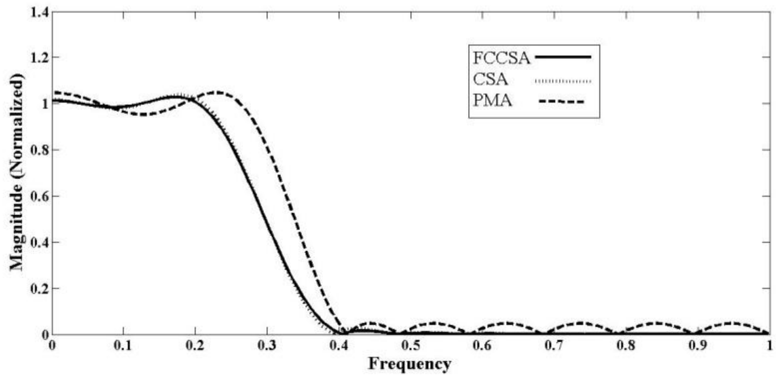

Figure 4.8(b)

Figure 4.8: (a) Normalized Frequency vs. Magnitude in dB of Lowpass Type II FIR Filter, (b) Frequency vs. Normalized Magnitude of Lowpass Type II FIR Filter

Table 4.13: Passband Ripple and Stopband Attenuation of Lowpass Type II FIR Filter

Algorithm	Passband Ripple	Stopband Ripple	Stopband Attenuation
PMA	0.072	0.05	-27.5795
CSA	0.061	0.019	-31.5
FCCSA	0.05	0.015	-36.08

Table 4.14: Optimized Coefficients of Lowpass Type II FIR Filter

Coefficients	PMA	CSA	FCCSA
h(1) = h(20)	-0.00837047290389881	0.00687139505578740	0.00597070222241318
h(2) = h(29)	0.0281336733236659	0.0174423137821560	0.0150826049318690
h(3) = h(18)	0.0285731449708783	0.0164209103714373	0.0143565509043312
h(4) = h(17)	0.00799066379637620	-0.00689170411939148	-0.00645038318579679
h(5) = h(16)	-0.0328713342200233	-0.0410816601622469	-0.0382539281101766
h(6) = h(15)	-0.0595899744082957	-0.0531555008435412	-0.0506179746679018
h(7) = h(14)	-0.0287496288147971	-0.0114653443951876	-0.0111865240708885

h(8) = h(13)	0.0732588077730955	0.0870279193390558	0.0856874381096096
h(9) = h(12)	0.208267409694031	0.205913120961259	0.204852594690420
h(10) = h(11)	0.304204477078734	0.287073221032930	0.287096880235132

Figure 4.9(a) shows plot of normalized Frequency vs. magnitude and Figure 4.9(b) shows plot of frequency vs. normalized magnitude of highpass Type I FIR Filters implemented using FCCSA, CSA and PMA. Table 4.15 contains comparative results of the filters in terms of passband ripple, stopband ripple and stopband attenuation. Table 4.16 contains optimized coefficients the filters.

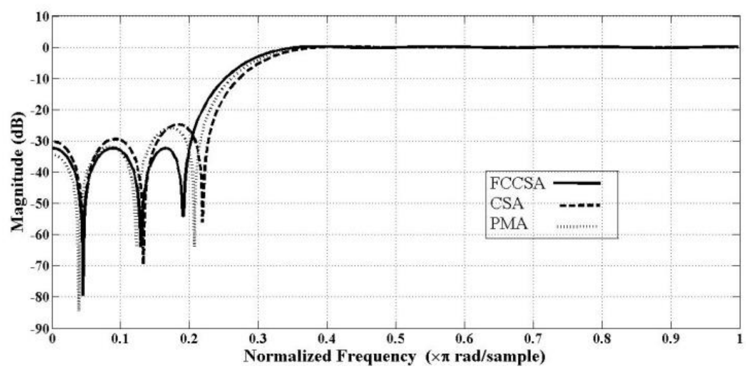

Figure 4.9(a)

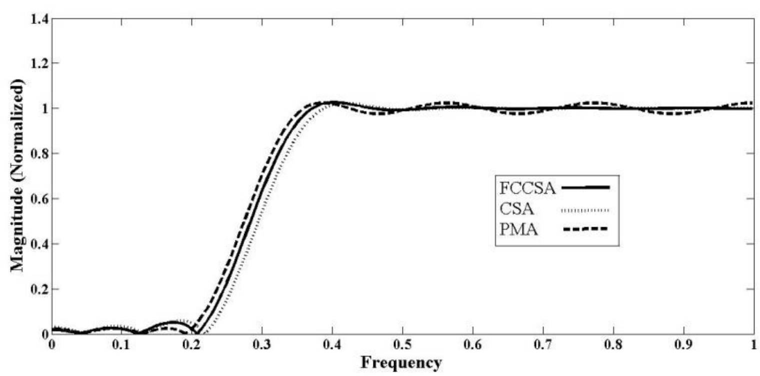

Figure 4.9(b)

Figure 4.9: (a) Normalized Frequency vs. Magnitude in dB of Highpass Type I FIR Filter, (b) Frequency vs. Normalized Magnitude of Highpass Type I FIR Filter

Table 4.15: Passband Ripple and Stopband Attenuation of Highpass Type I FIR Filter

Algorithm	Passband Ripple	Stopband Ripple	Stopband Attenuation
PMA	0.06	0.034	- 25.907
CSA	0.0014	0.032	-29.3349
FCCSA	0.0010	0.03	-32.38

Table 4.16: Optimized Coefficients of Highpass Type I FIR Filter

Coefficients	PMA	CSA	FCCSA
h(1) = h(21)	-0.000137726459964331	0.00748623582299282	-0.00558576985965225
h(2) = h(20)	-0.0176699545194173	0.0178027101280496	-0.0169493946104454
h(3) = h(19)	-0.0157693103654503	0.0179660074472004	-0.0218993002674030
h(4) = h(18)	-0.000873411544119435	-0.00155813005259396	-0.00667383452407309
h(5) = h(17)	0.0277785986917579	-0.0337294146442447	0.0269681742099115
h(6) = h(16)	0.0482280179796081	-0.0533244004058117	0.0542909006449092
h(7) = h(15)	0.0305241752696865	-0.0312704642826736	0.0402215568173479
h(8) = h(14)	-0.0407854938629945	0.0438755984369263	-0.0335158906933499
h(9) = h(13)	-0.148761211833621	0.152002790037944	-0.148729534819408
h(10) = h(12)	-0.247353255131035	0.247398385882968	-0.254767813362851
h(11)	0.712730949395183	-0.714547010307068	0.702234115682767

Figure 4.10(a) shows plot of normalized Frequency vs. magnitude and Figure 4.10(b) shows plot of frequency vs. normalized magnitude of highpass Type II FIR Filters implemented using

FCCSA, CSA and PMA. Table 4.17 contains comparative results of the filters in terms of passband ripple, stopband ripple and stopband attenuation. Table 4.18 contains optimized coefficients the filters.

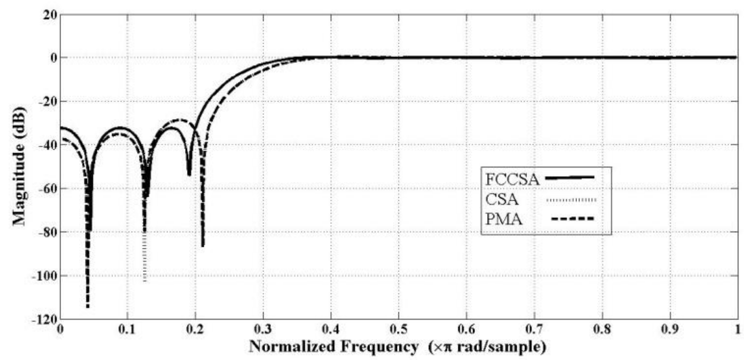

Figure 4.10(a)

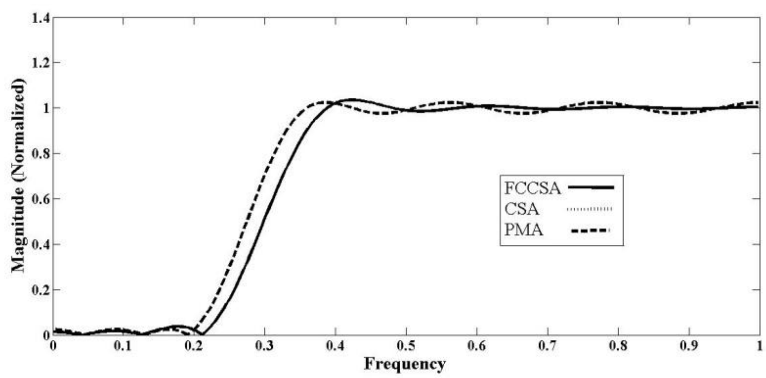

Figure 4.10(b)

Figure 4.10: (a) Normalized Frequency vs. Magnitude in dB of Highpass Type II FIR Filter,

(b) Frequency vs. Normalized Magnitude of Highpass Type II FIR Filter

Table 4.17: Passband Ripple and Stopband Attenuation of Highpass Type II FIR Filter

Algorithm	Passband Ripple	Stopband Ripple	Stopband Attenuation
PMA	0.089	0.042	-28.6
CSA	0.0408	0.03	-28.665
FCCSA	0.0316	0.028	-32.385

Table 4.18: Optimized Coefficients of Highpass Type II FIR Filter

Coefficients	PMA	CSA	FCCSA
h(1) = h(20)	0.0141216880680729	-0.0117393678914356	0.0118262307392677
h(2) = h(29)	-0.0217911062176947	-0.0203840519806205	0.0204249674053444
h(3) = h(18)	-0.0305505111468436	-0.00863290411005589	0.00858560157465317
h(4) = h(17)	-0.0205184606557532	0.0241924970731447	-0.0242855458833939
h(5) = h(16)	0.0150824802749166	0.0529777501522605	-0.0530189768149152
h(6) = h(15)	0.0530512697697476	0.0410011916503982	-0.0409576044392026
h(7) = h(14)	0.0511481151956367	-0.0318457187749407	0.0319667786146751
h(8) = h(13)	-0.0191936287477738	-0.147697576195291	0.147803814344124
h(9) = h(12)	-0.143319978325906	-0.255195274474480	0.255103909379504
h(10) = h(11)	-0.263316454182693	0.701146228086199	-0.701323247021478

Figure 4.11(a) shows plot of normalized Frequency vs. magnitude and Figure 4.11(b) shows plot of frequency vs. normalized magnitude of bandpass Type I FIR Filters implemented using FCCSA, CSA and PMA. Table 4.19 contains comparative results of the filters in terms of passband ripple, stopband ripple and stopband attenuation. Table 4.20 contains optimized coefficients the filters.

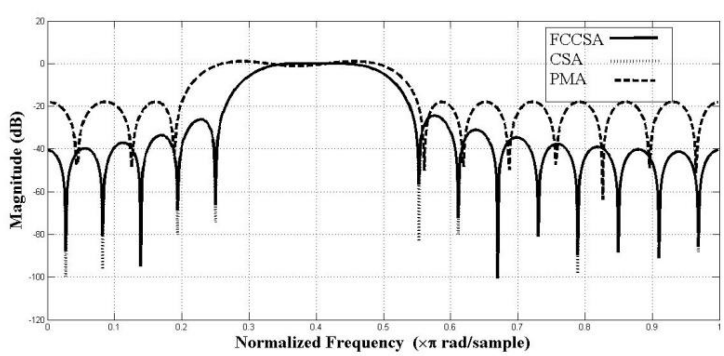

Figure 4.11(a)

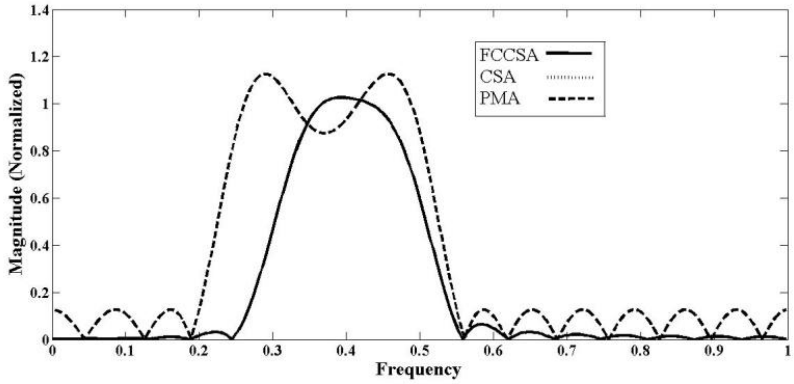

Figure 4.11(b)

Figure 4.11: (a) Normalized Frequency vs. Magnitude in dB of Bandpass Type I FIR Filter, (b) Frequency vs. Normalized Magnitude of Bandpass Type I FIR Filter

Table 4.19: Passband Ripple and Stopband Attenuation of Bandpass Type I FIR Filter

Algorithm	Passband Ripple	Stopband Ripple	Stopband Attenuation
PMA	0.31	0.18	- 18
CSA	0.0014	0.015	- 23. 7236
FCCSA	0.0011	0.012	-25.78

Table 4.20: Optimized Coefficients of Bandpass Type I FIR Filter

Coefficients	PMA	CSA	FCCSA
h(1) = h(31)	0.0225649236810043	0.0170748379280340	-0.0171182588203891
h(2) = h(30)	-0.0327527368769161	0.0100473207003458	-0.0100800756054404
h(3) = h(29)	0.0191196025661108	-0.0197092061247583	0.0196846104850894
h(4) = h(28)	-0.0118614163673383	-0.0209122078111174	0.0208556074098390
h(5) = h(27)	-0.0685961131249631	0.00450117963550519	-0.0045137269539507
h(6) = h(26)	-0.0548250284969489	0.00467044992938622	-0.0044814386209027
h(7) = h(25)	0.0236265390156190	-0.00534638469469250	0.00547177112844060

h(8) = h(24)	0.0481289716354942	0.0279758972855318	-0.0281481453655971
h(9) = h(23)	0.00461899095644996	0.0523972218776362	-0.0525501244375481
h(10) = h(22)	0.0192837900135152	-0.0246383472064071	0.0247317917166476
h(11) = h(21)	0.0797591204518425	-0.119121517080092	0.119156014177336
h(12) = h(20)	0.000748978916006141	-0.0485874002306499	0.0484933299022815
h(13) = h(19)	-0.191343945434930	0.134611759973402	-0.134509118622160
h(14) = h(18)	-0.181706024639650	0.151894367388243	-0.151644775033331
h(15) = h(17)	0.110250881888867	-0.0597700804054681	0.0596924629328150
h(16)	0.300178237382188	-0.200904398738736	0.200606151617971

Figure 4.12(a) shows plot of normalized Frequency vs. magnitude and Figure 4.12(b) shows plot of frequency vs. normalized magnitude of bandpass Type II FIR Filters implemented using FCCSA, CSA and PMA. Table 4.21 contains comparative results of the filters in terms of passband ripple, stopband ripple and stopband attenuation. Table 4.22 contains optimized coefficients the filters.

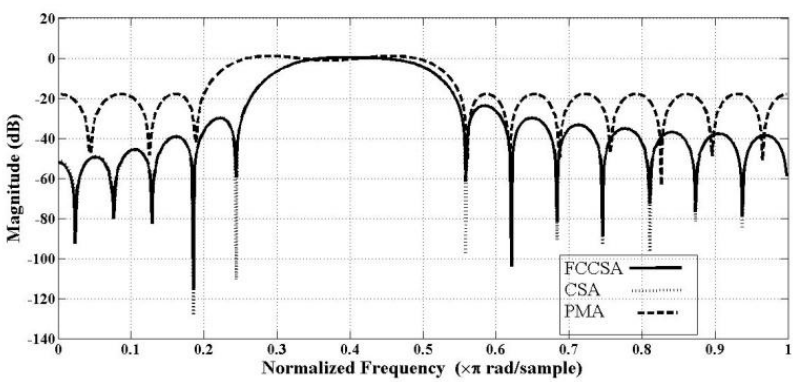

Figure 4.12(a)

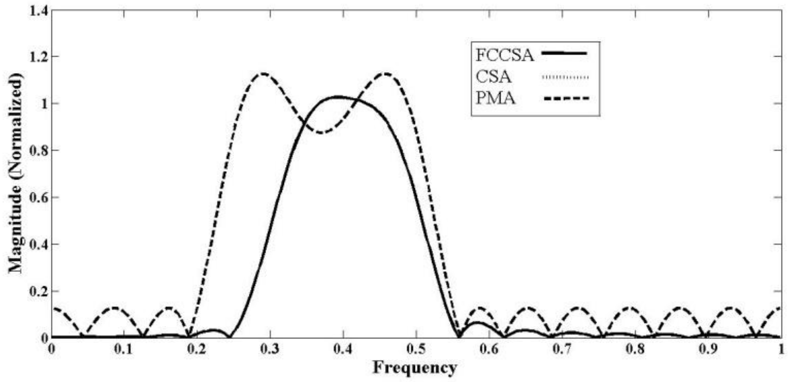

Figure 4.12(b)

Figure 4.12: (a) Normalized Frequency vs. Magnitude in dB of Bandpass Type II FIR Filter, (b) Frequency vs. Normalized Magnitude of Bandpass Type II FIR Filter

Table 4.21: Passband Ripple and Stopband Attenuation of Bandpass Type II FIR Filter

Algorithm	Passband Ripple	Stopband Ripple	Stopband Attenuation
PMA	0.31	0.18	- 18
CSA	0.0015	0.016	- 23.72375
FCCSA	0.0012	0.014	-25.96

Table 4.22: Optimized Coefficients of Bandpass Type II FIR Filter

Coefficients	PMA	CSA	FCCSA
h(1) = h(30)	0.0225649236810043	0.0150907875477844	-0.014806418259073
h(2) = h(29)	-0.0327527368769161	0.000367212966049854	-0.000304532592591
h(3) = h(28)	0.0191196025661108	-0.0211835862366085	0.0207795937981213
h(4) = h(27)	-0.0118614163673383	-0.00853565514888823	0.00835755808540606
h(5) = h(26)	-0.0685961131249631	0.00777110981867764	-0.0074445267265577
h(6) = h(25)	-0.0548250284969489	-0.00419399734853405	0.00442874495494476
h(7) = h(24)	0.0236265390156190	0.00326937520040729	-0.0034143509220554
h(8) = h(23)	0.0481289716354942	0.0474662694866716	-0.0475627931284042
h(9) = h(22)	0.00461899095644996	0.0302721677204632	-0.0300986953182150

h(10) = h(21)	0.0192837900135152	-0.0786238434760165	0.0786466882788071
h(11) = h(20)	0.0797591204518425	-0.110914227140650	0.110650753237896
h(12) = h(19)	0.000748978916006141	0.0414172519237242	-0.0414924784132797
h(13) = h(18)	-0.191343945434930	0.177217303827776	-0.176986933958765
h(14) = h(17)	-0.181706024639650	0.0629555995504433	-0.0627636078912966
h(15) = h(16)	0.110250881888867	-0.160995161493941	0.160783271805496

Above mentioned case studies proved efficacy of the Fast Converging Cuckoo Search Algorithm and the proposed fitness function for optimizing filter coefficients. Proposed method outpaces conventional technique of using PMA for obtaining optimized coefficients set of Type I and Type II lowpass, highpass as well as bandpass FIR filters.

Table 4.23, 4.24 and 4.25 show comparative studies of the execution time of GA, PSO, BAT Algorithm, CSA and FCCSA while used for lowpass, highpass and bandpass FIR filters coefficients optimization respectively.

Table 4.23: Comparison of execution time (in seconds) of different algorithms to design 19th and 20th order Lowpass FIR Filter

Algorithm	**Filter Type**	
	Type I (order 20)	**Type II (order 19)**
GA	48.3400	48.2000
PSO	46.2060	46.0002
BAT Algorithm	44.6422	44.1020
CSA	43.4682	43.2667
FCCSA	43.2362	43.0042

Table 4.24: Comparison of execution time (in seconds) of different algorithms to design 19th and 20th order Highpass FIR Filter

Algorithm	Filter Type	
	Type I (order 20)	Type II (order 19)
GA	49.1420	49.5500
PSO	48.1244	48.0008
BAT Algorithm	47.0022	47.1146
CSA	45.2280	45.1740
FCCSA	43.2362	43.0042

Table 4.25: Comparison of execution time (in seconds) of different algorithms to design 29th and 30th order Bandpass FIR Filter

Algorithm	Filter Type	
	Type I (order 30)	Type II (order 29)
GA	80.1315	80.0142
PSO	78.1232	78.1008
BAT Algorithm	75.1200	75.5532
CSA	72.2520	72.1765
FCCSA	69.4450	69.0004

4.4. Conclusions

Simulation results and discussions have proved that use of BAT algorithm for designing excellent performing lowpass FIR filters is worthwhile compared to GA and PSO. It can be observed that the stop-band attenuation for 20th order low pass filter using BAT Algorithm is -

31.4 dB while the same using GA, PSO and Parks-McClellan, are -22.2 dB, -25.2 dB and -24.9 dB. The normalized stop-band ripple obtained using BAT Algorithm is 0.03. This specifies superior performance of the BAT Algorithm at the cost of little increase in pass-band ripple with respect to PSO (0.10 vs. 0.06). Ripple performances in stopband are improved by 3.00%, 14.27% and 3.14% compared to PSO, GA and PM respectively. BAT algorithm can be further applied to digital highpass FIR filter design. Further modification in BAT algorithm can also be adapted for obtaining optimized filter coefficients to design better performing FIR filters. Another innovative approach of using a new algorithm Global Best Steered Cuckoo Search Algorithm (gbest CSA) to design linear phase symmetric FIR filters is also presented in this chapter. Proposed algorithm is used to implement 21st order lowpass filter and 25th order bandpass filters. For optimization, an adaptive fitness function based on weighted mean square error is used. The result shows that use of gbest CSA along with the mean square error based fitness function for designing Type II lowpass and bandpass FIR filter is worthwhile for improving filter characteristics. Lowpass filter of order 21 implemented by the optimized sets of filter coefficients obtained using gbest CSA offers gain in stopband attenuation of 38% and 6% compared to PMA and CSA respectively, whereas to design 25th bandpass filter gain in stopband attenuation is 25% and 27% in comparison with PMA and CSA respectively. In this chapter an adaptive algorithm namely Fast Converging Cuckoo Search Algorithm (FCCSA) is also proposed and efficacy of the algorithm is proved while implementing linear phase symmetric FIR filters. Lowpass, highpass and bandpass filters of even and odd order have been realized using FCCSA. For optimization weighted mean square error is used. The result shows that use of FCCSA along with the proposed fitness function for designing Type I and Type II lowpass, highpass as well as bandpass FIR filters is sensible not only for improving filter characteristics but also minimizing execution time. Filters implemented by the optimized sets of filter coefficients obtained using FCCSA offers flat passband and higher stopband attenuation. All the above mentioned algorithms can be used to design efficient digital filters. To remove noise from the corrupted bio-medical signals for error free diagnosis these filters are most useful. In the astrophysical signals received by the satellite bit of noise can make huge difference to the data, use of these filters for de-noising astrophysical signals are much worthy.

Chapter 5

Hardware Efficient Filter Design

5.1. Introduction

Digital filters have been extensively used in the last few decades for biomedical signal processing, noise elimination, astrophysical signal processing, etc. Trifling hardware cost and extraordinary behaviour of altering characteristics with changes in the discrete values stored in the registers have made the digital filters more efficacious than the analog ones. Digital filters are classified into two types - (i) Finite Impulse Response (FIR) Filter (Salivahanan et al., 2007; Sharma, 2009; Mitra, 2013), and (ii) Infinite Impulse Response (IIR) Filter (Salivahanan et al., 2007; Sharma, 2009; Mitra, 2013; Singh & Arya, 2012; Karaboga & Cetinkaya, 2014). Minimalisms in hardware and fluently attainable linear phase properties have made FIR filters more worthwhile. FIR filters produce inherent stable response due to absence of poles in the transfer functions. The key steps involved in FIR filter design are multiplication and accumulation of filter coefficients with the input discrete time signal. Due to the consumption of higher amount of power and, it has become necessary to replace the multipliers with shift and adder circuits. For realizing multiplier less filter circuits, coefficients can be represented as sums or differences of signed-power-two (SPT) terms (Solank, 2012). Requirement of adders including Structural Adders (SA) and Multiplier Adders (MA) depend on total the number of SPT terms used to represent a set of filter coefficients (Reddy & Sahoo, 2015; Reddy, 2015). If the number and attributes of filter coefficients vary with the filter specifications, the total number of adders also varies.

The design strategies of FIR filters are classified as window method (Kaiser, 1966; Harris & Fredric, 1978; Sharma, 2009), frequency sampling technique (Gold & Jordan, 1969) and use of optimal filter coefficients (Herrmann, 1970; Parks & McClellan, 1972; Reddy & Sahoo, 2015). Optimization of filter coefficients is characterized as a problem with an objective of minimizing the errors in both passband and stopband. Error in the frequency bands is mathematically conceptualized as an approximation function representing the deviancy between the designed filter responses and the ideal filter responses. In order to obtain appropriate set of filter coefficients, Parks and McClellan (1972) proposed an algorithm namely Parks McClellan Algorithm (PMA) (Parks & McClellan, 1972) aiming to receive exact linear phase response. The algorithm is inefficient in independent selection of passband and stopband ripples as it uses their ratio. In optimization based methods, the design problem of FIR filters is formulated as either single or multi-objective optimization problem which can be solved by either the heuristic approach or the meta-heuristic approach.

Heuristic algorithmic approach has weakness in determining local optimal solutions in terms of convergence speed although it has a unique nature of searching in neighbourhoods aiming to get the optimal solution. Heuristic algorithms (Kokash, 2005; Aickelin & Clark, 2011) determine high quality solutions to tough optimization problems; however, they do not assure the optimum solution. Overcoming these limitations, further research led to the development of the metaheuristic algorithms (Baghel et al., 2012; Yang, 2010a; Yang, 2011). Besides better performance than simple heuristic algorithms, meta-heuristic algorithms have another advanced feature of using certain trade-offs of randomization and local search. Meta-heuristic algorithms perform better to reach the global optima even in presence of limited information about the problem. Behaviour of biological systems and/or physical systems in nature stood as the motivation behind majority of the meta-heuristic algorithms.

Among a considerable number of nature inspired meta-heuristic algorithms, Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995; Yang et al., 2013) exhibits fast convergence in many practical applications, however, it suffers from the problem of premature convergence. Specifically, in case of solving multimodal and nonlinear problems with a huge number of local minima, PSO gets trapped into local optima. Performance of PSO is improved by integrating the principles of quantum superposition and quantum probability, hence accelerating the search for an optimal solution. Some other algorithms are also available in literature that performs better in solving unimodal as well as multimodal problems. Among them, a few newly developed algorithms are Cuckoo Search Algorithm (CSA) (Yang & Deb, 2009; Yang & Deb. 2010), Flower Pollination Algorithm (FPA) (Yang, 2014; Yang, 2013), BAT Algorithm (Yang, 2014; Yang, 2010d; Fister et al., 2015), etc. Amid these algorithms CSA and FPA are efficient not only in terms of performance but also in computational time. Moreover they require less number of parameters to be tuned rather than GA (Oner, 1998; Aggarwal et al., 2015) and PSO (Najjarzadeh & Ayatollahi, 2008). In the proposed work, improvement on standard CSA (Yang & Deb, 2009; Yang, 2014) is achieved by incorporating quantum principles (Ventura & Martinez, 1997) and modifying the replacement strategy of the worse quality nests, resultant a new algorithm which we refer to as Global Best Steered Quantum Inspired Cuckoo Search Algorithm (GQICSA). This algorithm outpaces basic algorithms Quantum Inspired Cuckoo Search Algorithm (QICSA) (Layeb & Boussalia, 2012; Laha, 2015; Djelloul et al., 2015), CSA and FPA in terms of convergence time. In the work, reported in this chapter, we used both QICSA and GQICSA to obtain optimized sets of filter coefficients for the design of FIR filters with responses likely to the ideal filters with minimum adder cost. Reduction in the number of required adders refers to the reduction of SPT terms. The key concept behind the approach used to reduce the SPT terms relies on the notion that the set of filter coefficients are not exclusive for particular filter specifications such as stopband attenuation, passband ripple and order of filter. QICSA performs better for this specific problem compared to GA (Oner, 1998; Aggarwal et al., 2015), PSO (Najjarzadeh & Ayatollahi, 2008), CSA (Yang & Deb, 2009; Yang, 2014), FPA (Yang, 2014; Yang, 2013) and Quantum behaved Particle Swarm Optimization (QPSO) (Sun et al., 2004; Long et al., 2010; Dhabal & Sengupta,2015). The proposed algorithm, GQICSA, shows even better performance compared to all the above mentioned algorithms as well as QICSA (Layeb & Boussalia, 2012; Laha, 2015; Djelloul et al., 2015).

5.2. Design Problem Formulations

Finite Impulse Response (FIR) filters can be described by the system transfer function in Equation 5.1,

$$H(z) = \sum h(n)z^{-n}, n = 0,1, \ldots, N \tag{5.1}$$

Where N signifies the order of the filter, $h(n)$ represents the set of filter coefficients containing $(N + 1)$ elements in the set (Sharma, 2009). For symmetric FIR filters, $h(n)$ can be defined by Equation 5.2.

$$h(n) = h(N - 1 - n), \; n = 0,1, \ldots, N - 1 \tag{5.2}$$

For optimizing coefficients of symmetric filters, only $\lfloor (N/2) + 1 \rfloor$ coefficients must be taken into consideration whereas in case asymmetric filters all the $N + 1$ coefficients need to be optimized. Based on the filter order and symmetricity of filter coefficients, filters are categorized into following four types.

i. Type I- Even order and symmetric coefficients,

ii. Type II- odd and symmetric coefficients,

iii. Type III- Even order and asymmetric coefficients,

iv. Type IV- Odd order and asymmetric coefficients.

In this work, we address design of Type I & Type II lowpass filters.

FIR filter design involves three basic steps:

i. Specifying the filter desires

ii. Obtaining appropriate set of filter coefficients

iii. Design filter architecture

5.2.1. Specification of desired filter characteristics

Filter design initiates with the specification of filter characteristics such as passband frequency (ω_p), stopband frequency(ω_s), passband ripple(δ_p), stopband ripple(δ_s). The magnitude response has a peak deviation of δ_p in passband and maximum deviation of δ_s in stopband. Minimum stopband attenuation (A_s), maximum and minimum passband attenuation $A_{p_1}$and A_{p_2} respectively can be expressed in dB using the following Equations:

$$A_s = -20\log_{10}\delta_s \tag{5.3}$$

$$A_{p_1} = 20\log_{10}\left(1+\delta_p\right) \tag{5.4}$$

$$A_{p_2} = 20\log_{10}\left(1-\delta_p\right) \tag{5.5}$$

5.2.2. Filter coefficients computation

In the traditional equiripple method for obtaining optimized set of filter coefficients, a desired filter response is approximated with varying degree of success of an objective function that minimizes the error between the approximated filter response and the desired filter response. This method uses an approach for minimizing errors in both the passband and stopband (Aggarwal et al., 2013). The error function (cf. Equation 5.3), in terms of the weighted difference of the ideal and approximated frequency responses in both pass-band and stop-band, is used in most of the cases (Singh & Josan, 2014),

$$E(\omega) = W(\omega)[H_d\left(e^{j\omega}\right) - H_a(e^{j\omega})] \tag{5.6}$$

where $H_d\left(e^{j\omega}\right)$ and $H_a(e^{j\omega})$ are the desired and approximated frequency response of the filter. Ideal frequency response of a lowpass filter is stated as in Equation 5.7 (Aggarwal et al., 2013; Singh & Josan, 2014).

$$\begin{aligned} H_d(e^{j\omega}) &= 1 \qquad for\ \ 0 \leq \omega \leq \omega_p \\ &= 0 \qquad for\ \ \omega_s \leq \omega \leq \pi \end{aligned} \tag{5.7}$$

The weighing factor $W(\omega)$offers control over error minimization in both the frequency bands. Using the key concept of the equiripple methodology, Parks and McClellan (Parks & McClellan, 1972) proposed an efficient algorithm for optimal filter design. Limitation of this strategy is the fixed value of $\frac{\delta_p}{\delta_s}$. In a bid to overcome this limitation, we adopted a mean square error (Dhabal & Sengupta, 2015) based objective function as in Equation 5.8.

$$\varphi = \mu E_p + (1-\mu)E_s\ \ ;\ \ 0 < \mu < 1 \tag{5.8}$$

For lowpass filter, and Es are stated by Equation 5.9 and 5.10 respectively.

$$E_p = \frac{1}{\pi}\int_0^{\omega_p}(1 - H(\omega) - \delta_p)^2 d\omega \tag{5.9}$$

$$E_s = \frac{1}{\pi}\int_{\omega_s}^{\pi}(0 - H(\omega) - \delta_s)^2 d\omega \tag{5.10}$$

Here, H is the magnitude response of the approximated filter. The objective function φ is the weighted sum of mean square errors $E_p$and E_s. By minimizing φ, better performance can be obtained by the approximated filters in terms of passband ripples and stop band attenuation.

5.2.3. Design filter architecture

The basic operations involved in implementing FIR filter architecture are multiplication and accumulation of filter coefficients with the input discrete time signal. High amount of power and area consumption by the multipliers causes them to be replaced with shift and adder circuits. For implementing multiplier less filter circuits, coefficients are represented as sums or differences of signed-power-two (SPT) terms (Solank, 2012). The transposed direct form of FIR filter can be defined by Equation 5.11,

$$Y(n) = \sum_{k=0}^{M-1} C_k X(n-k) \tag{5.11}$$

where, $X(n)$ represents input to the filter of order N,$C_0, C_1, \ldots\ldots, C_{M-1}$represent the filter coefficients, length of the filter is denoted by M (M=N+1) and $Y(n)$ represents the filter output. Transpose direct form of filter implementation consists of structural and multiplier adders along with the delay elements. Structural adders are used to add the input signal $X(n)$, multiplied by the filter coefficient product value, along with the stored value in delay element. The number of structural adders equals to the total number of addition required to obtain filter coefficients after Common Sub Expression elimination. Multiplier adders are used for coefficient multiplication and the number of multiplier adders equals the total number of SPT terms or the nonzero bits required to represent filter coefficients neglecting the repeating terms or bit positions. If the number and attributes of filter coefficients vary with the filter specifications, the total number of adders also varies. Reduction in number of required adders refers to the reduction of SPT terms. The key concept behind the approach used to reduce the SPT terms relies on the notion that the set of filter coefficients are not exclusive for particular filter specifications such as stopband attenuation, passband ripple and order of filter. Suppression in the word length of the filter coefficients and representing them in Canonical Signed Digit (CSD) form lead to reduction in SPT terms and hence in total number of adders. Hardware cost of a filter is very much dependent on the total number of adders required to design a filter.

The hardware requirement for implementing the Eqn. 5.11 can be stated as follows:

Multipliers: Multiplication between $X(n)$ and filter coefficients $c[k]$.

Adders: For accumulating purpose.

Delay Elements: For storing the previous inputs or accumulated values.

Eqn. 5.8 requires (M-1) numbers of adders, (M-1) numbers of delay elements and M numbers of multipliers. Among several techniques for FIR filter design direct form and transposed direct form structures are shown below (Reddy, 2015):

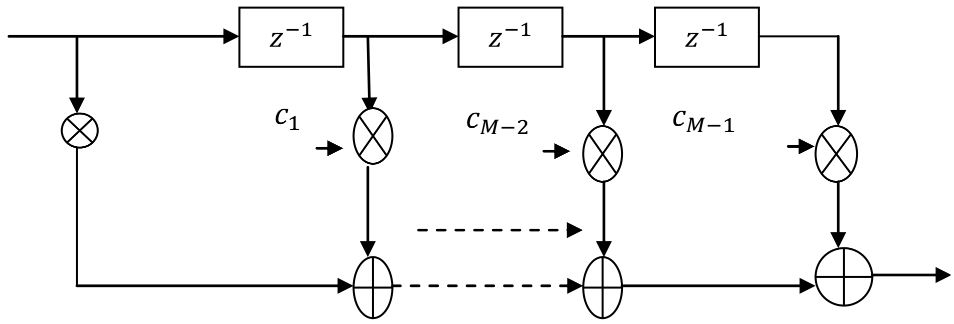

Figure 5.1: Direct Form Structure

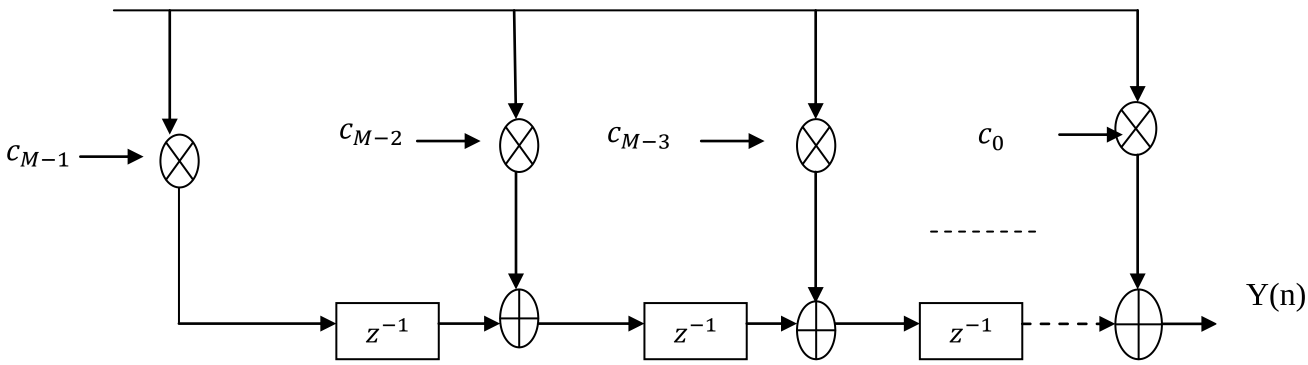

Figure 5.2: Transposed Direct Form Structure

In this chapter, at first conventional Cuckoo Search is updated by quantum principles and the quantum inspired Cuckoo Search algorithm is used to obtain optimized coefficients set of Finite Impulse Response Filter. After that an adaptive replacement strategy is incorporated with the QICSA and resultant algorithm namely Global Best Steered QICSA (GQICSA) is used for the same purpose. Word length of each optimized coefficient of a filter is then fixed at 10. Hardware efficiency of the filter implemented using the coefficients having fixed word length of 10 is then measured by estimating the required number of adders including structural adder and multiplier adder. In simulation results, it has been shown that reduction in word length of the coefficients does not make the filters incapable of achieving the desired frequency response.

5.3. Overview of Quantum Computing

Quantum computing (Kaye et al., 2007) is a latest theory which has appeared as a result of amalgamating computer science and quantum mechanics. The term "quantum computer" can be defined as a computer system, designed on the basis of quantum theory, was first proposed by Nobel prize-winning physicist Richard Feynman in 1982 (Fynman, 1982). In quantum computers, computations are performed on the basis of the laws of quantum mechanics (Benioff, 1980). Initially the concept of a quantum computer was of theoretical concern only, but topical developments like invention of Shor's algorithm (Bone and Castro) to factor large numbers on a quantum computer, by Peter Shor (Bell Laboratories), geared up the researchers to win the race for implementing a practical quantum computer (Beckman et al., 1996). Shor's algorithm, makes the quantum computer capable to crack codes quicker than any classical computer could.

Recently quantum computing has drawn attention of the researchers of various modern areas since it appears more powerful than its classical counterpart. Indeed, the characteristic of parallelism that the quantum computing provides decreases the algorithmic complexity. This makes the Quantum-inspired algorithms more acceptable compared to the classical algorithms for some hard hitting problem solving. The theories of quantum computation have some interesting implications in the world of artificial intelligence also.

The basic component of quantum computing is "qubit" (Samanta et al., 2017; Konar et al, 2017; Dey et al., 2017). It is a unit vector defined over two-dimensional Hilbert space, where a particular basic state can be indicated by $|0\rangle$ and $|1\rangle$ (Han & Kim 2002; Li & lu, 2015).

Based on the fundamental concept of the superposition principle, if a quantum system can be represented by any one of the two basic states, then it can also be represented as a linear combination of these two states, such as $\alpha_0|0\rangle + \alpha_1|1\rangle$, where the coefficients α_0 and α_1 are the "amplitudes" of the states $|0\rangle$ and $|1\rangle$ respectively (Konar et al., 2017). These coefficients give the probabilistic measure of the occurrence of state $|0\rangle$ and state $|1\rangle$ respectively (Samanta et al., 2017; Dey et al., 2017). The superposition $\alpha_0|0\rangle + \alpha_1|1\rangle$ is the basic, or the smallest, unit of encoded information in quantum computers or quantum systems. The qubit representation is given by Equation 5.12 (Samanta et al., 2017; Konar et al, 2017; Dey et al., 2017).

$$|\varphi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle \quad (5.12)$$

According to the superposition principle, α_0 and $\alpha_1$are arbitrary complex numbers and the squares of their norms add up to 1, as indicated in Equation 5.13 (Samanta et al., 2017; Konar et al, 2017; Dey et al., 2017).

$$|\alpha_0|^2 + |\alpha_1|^2 = 1 \quad (5.13)$$

$\alpha_0$and$\alpha_1$are the probabilistic amplitude of the qubit that may exist in one of the two states (state "0" or state "1") and ensure that the normalization condition is met.

5.3.1. Qubit Representation

The equation for the superposition state of a qubit can be stated by the sum of the two basic states corresponding to their probabilistic amplitude coefficients α_0 and α_1 respectively. Here, α_0 and α_1 are complex but are generally considered real without any loss. A qubit lies in a coherent superposition of states $|0\rangle$ and $|1\rangle$ (Samanta et al., 2017; Konar et al, 2017; Dey et al., 2017). After measurement it must be either at state 0 or state 1. If we are having two qubits, these can be in one of the four computational fundamental states i.e. 00, 01, 10 or 11. Pair of qubits should be in a superposition of four states stated in the following Equation (Samanta et al., 2017; Konar et al, 2017; Dey et al., 2017):

$$|\varphi\rangle = \alpha_{00}|00\rangle + \alpha_{01}|01\rangle + \alpha_{10}|10\rangle + \alpha_{11}|11\rangle \quad (5.14)$$

Normalization condition of these four states can be stated by the following Equation (Samanta et al., 2017; Konar et al, 2017; Dey et al., 2017):

$$|\alpha_{00}|^2 + |\alpha_{01}|^2 + |\alpha_{10}|^2 + |\alpha_{11}|^2 = 1 \quad (5.15)$$

In this way a qubit string i.e. set of some individual qubits can be represented as following (Samanta et al., 2017; Konar et al, 2017; Dey et al., 2017):

$$q = \begin{bmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_l \\ \beta_1 & \beta_2 & \dots & \beta_l \end{bmatrix} \quad (5.16)$$

Here, an individual qubit can be stated as $\begin{bmatrix} \alpha \\ \beta \end{bmatrix}$, l refers to the length of the quantum string q.

Ability of parallel processing of the quantum algorithms can be used to solve combinatorial optimization problems requiring the exploration of large solutions spaces. For obtaining better optimal solutions in quantum inspired optimization algorithms, qubits are updated using quantum gates. Updated qubits may also increase the convergence rate of the algorithm. Usually quantum gates operate on one or two qubits similar like the classical logic gates. Hence, the gates can be symbolized using 2×2 or 4×4 unitary matrices.Key concept of a unitary matrix can be stated by the following Equation (Samanta et al., 2017; Konar et al, 2017; Dey et al., 2017):

$U^{-1} = U^{+1}$, where U^{-1} and U^{+1} are unitary and $U^{+1}U = 1$. (5.17)

This ensures the logically reversible property of the quantum gate.

5.3.2. Quantum Operators

Basic quantum operator is quantum rotational gate which is generally used to update the qubits. The coefficients (α_i, β_i) of the ith qubit can be updated using the following Equation (Han & Kim, 2002; Lu & Li, 2015; Samanta et al., 2017; Konar et al, 2017; Dey et al., 2017):

$$\begin{bmatrix}\alpha_i' \\ \beta_i'\end{bmatrix} = U(\Delta\theta_i) \times \begin{bmatrix}\alpha_i \\ \beta_i\end{bmatrix}$$,where $U(\Delta\theta_i)$ stands for quantum rotation gate. (5.18)

Quantum rotation gate can be stated as following (Samanta et al., 2017; Konar et al, 2017; Dey et al., 2017):

$$U(\Delta\theta_i) = \begin{bmatrix} cos(\Delta\theta_i) & -sin(\Delta\theta_i) \\ sin(\Delta\theta_i) & cos(\Delta\theta_i) \end{bmatrix} \tag{5.19}$$

Rotation angle θ_i is invoked to update the qubit to (α_i', β_i'). There are a considerable number of gates in quantum computing such as NOT gate, controlled NOT gate, controlled phase-shift gate, Toffoli gate, Fredkin gate, Hadamard gate (Dey et al., 2017).

5.4. Quantum Inspired Optimization Algorithms

For solving optimization-related problems and issues, use of population based algorithms inspired by nature have been proved to be efficient in literature. Aiming to reduce the algorithmic complexity researchers has incorporated the quantum principles into traditional nature inspired algorithms. Resultant quantum behaved nature inspired algorithms have proved their efficiency for solving complex optimization problems. Quantum algorithms involve applications of a series of quantum operations successively on a quantum system. Quantum operations can be performed using quantum gates and quantum circuits. As there is not a powerful quantum machine able to execute the developed quantum algorithms, researchers have tried to adapt some properties of quantum computing in the classical algorithms since last two decades. The key notion of quantum inspired optimization algorithms is to represent the possible solutions of a problem as the superposition of all those solutions. This specific type of representing potential solutions is termed as quantum representation. For updating the solutions in a population through generations, quantum operators are used.

5.4.1. Quantum Inspired Cuckoo Search Algorithm

Modification in conventional CSA is performed by collaborating with quantum principles for solving complex, multimodal problems, like higher dimensional filter design. At the beginning of solving an optimization based problem using Quantum Inspired Cuckoo Search Algorithm (QICSA) (Layeb & Boussalia, 2012; Djelloul et al., 2015) a set of solutions, termed as initial population, is generated but within a specific range demarcated by lower and upper bounds. In the next step, quantum representation of all the solutions in the initial population is performed. In QICSA, a balanced composition of local random walks and global explorative random walks is also used for generating new solutions with the aim of replacing the old bad solutions. Local random walk is performed following the same way as performed in CSA using following Equation (Yang & Deb, 2009; Yang & Deb, 2010):

$$x_i^{t+1} = x_i^t + \alpha s \otimes H(p_a - \epsilon) \otimes (x_j^t - x_k^t) \tag{5.20}$$

In QICSA, for obtaining new solutions by updating the old ones, Monte Carlo method defined in Equations 5.21-5.24 are obeyed (Lu & Li, 2015).

$$x_i^{t+1} = p_i^t \pm \frac{1}{2} H_i^t \ln(1/u_i^t) \tag{5.21}$$

Where $H_i^t = 2\alpha |c^t - x_i^t|$ (5.22)

$$c^t = \left(\frac{1}{N}\sum_{i=1}^{N} p_{i,1}^t, \frac{1}{N}\sum_{i=1}^{N} p_{i,2}^t, \dots, \frac{1}{N}\sum_{i=1}^{N} p_{i,d}^t\right) \tag{5.23}$$

$$p_i^t = x_i^t + \gamma L(\lambda)(g_* - x_i^t) \tag{5.24}$$

Here, $L(\lambda)$ stands for Lévy flights, g_* represents the global best solution, γ denotes the scaling factor used to control the step size and λ refers to the step length which can be drawn from a Lévy distribution, α corresponds to the contraction expansion coefficient which can be tuned to control the convergence speed of the algorithms, and u_i^t is a random number uniformly distributed over (0, 1). Value of α can computed using Equation 5.25 (Lu & Li, 2015).

$$\alpha = (1 - t)/iter_{MAX} * 0.5 \tag{5.25}$$

Finally, in order to perform global random walk using Lévy flights, Equations 5.26 and 5.27 are used based on a specific condition as outlined in Algorithm 5.1.

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(g_* - x_i^t) + \alpha|c^t - x_i^t| \ln(1/u_i^t) \tag{5.26}$$

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(g_* - x_i^t) - \alpha|c^t - x_i^t| \ln(1/u_i^t) \tag{5.27}$$

Input: Total No. of Iteration ($iter_{MAX}$), size of the population (n), objective function (f), lower bound, upper bound, λ in Lévy distribution, probability p_a for discovering bad quality nests
Output: Global Best Solution.
Begin
Define objective function $f(x), x = (x_1, x_2, \dots, x_d)$
Initialize a population of n nests
Evaluate the fitness of the solutions using f
Find the best quality nest and store in $BEST$
Store fitness value of $BEST$ in $fmin$
Define a switch probability p_a [0, 1] and scaling factor γ; $t = 1$
while($t < iter_{MAX}$)
  Define c^t by Equation 5.23
  for $i = 1{:}N$
    Generate a random number u and generate α by Equation 5.25
    Draw a (d-dimensional) step vector L which obeys a Lévy distribution
    if ($u > 0.5$)
      Generate new solution using Equation 5.26
    else
      Generate new solution using Equation 5.27
    end if
    Discover bad nests with probability p_a
    Replace bad nests by the new nests generated using the Equation 5.20
  end for
  Evaluate the fitness of the solutions using f
  Find the current best nest c_{best} and store its fitness value in $cmin$
  if ($cmin < fmin$)
    Update $BEST$ and $fmin$
  End if
  $t = t + 1$
End while
End

Algorithm 5.1: *Quantum Inspired Cuckoo Search Algorithm*

5.4.2. Global Best Steered Quantum Inspired Cuckoo Search Algorithm

In our proposed technique named as Global Best Steered Quantum Inspired Cuckoo Search Algorithm (GQICSA) a modification was performed in replacement strategy of standard QICSA; another modification was in choice of a parameter (λ) in Lévy Distribution. In reality higher similarity of a cuckoo egg with the host's eggs decreases the probability of the cuckoo egg to be distinguished by the host from its own eggs. As the fitness of a candidate solution termed as cuckoo egg is related to its difference with the solutions of the latest iteration, therefore, it will be better to perform a random walk in a biased way with random step sizes. Similar like Cuckoo search, Global best steered cuckoo search also uses random step sizes but with different function set for computing the step size. In CSA and QICSA, step size is computed using the following expression:

$$r * nest\big[permute1[i]\big][j] - nest\big[permute2[i]\big][j] \tag{5.28}$$

Where r is a random number lying within the range [0, 1].*nest* denotes a matrix containing candidate solutions along with their variables, *permute1* and *permute2* are different rows permutation functions applied on *nest* matrix (Yang & Deb, 2010). For Global Best Steered Cuckoo Search Algorithm (gbest CSA) step size is computed using the following equation:

$$r * nest[[best] - nest\big[permute[i]\big][j] \tag{5.29}$$

In case of GQICSA instead of using permute1 function, best nest till the latest iteration is used. This specialty of GQICSA keeps the selection pressure towards the better solutions, hence assures better result. Moreover, this advancement of the Quantum Inspired Cuckoo Search Algorithm does not flood the population by the high fitness solutions. In our proposed algorithm instead of using $\lambda = 1.5$, λ is determined by the following equation (Dhabal & Venkateswaran, 2017):

$$\lambda = (\lambda_{max} - \lambda_{min}) \times \frac{(iter_{total} - iter)}{iter_{total}} + \lambda_{min} \tag{5.30}$$

Where $\lambda_{max} = 1.5$ and $\lambda_{min} = 1$.

Input: Total No. of Iteration ($iter_{MAX}$), size of the population (n), objective function (f), lower bound, upper bound, bounds for λ in Lévy distribution, probability p_a for discovering bad quality nests
Output: Global Best Solution.
Begin
Define objective function $f(x), x = (x_1, x_2, \dots., x_d)$
Initialize a population of n nests
Evaluate the fitness of the solutions using f
Find the best quality nest and store in $BEST$
Store fitness value of $BEST$ in $fmin$
Define a switch probability p_a [0, 1] and scaling factor γ
Compute λ using Equation 5.27
$t = 1$
while$(t < iter_{MAX})$
  Define c^t by Equation 5.23
  for $i = 1: N$
    Generate a random number u and generate α by Equation 5.25
    Draw a (d-dimensional) step vector L which obeys a Lévy distribution
    if $(u > 0.5)$
      Generate new solution using Equation 5.26
    else
      Generate new solution using Equation 5.27
    end if
    Discover bad nests with probability p_a
    Replace bad nests by the new nests generated using the Equation 5.30
  end for
  Evaluate the fitness of the solutions using f
  Find the current best nest c_{best} and store its fitness value in $cmin$
  if $(cmin < fmin)$
    Update $BEST$ and $fmin$
  End if
  $t = t + 1$
End while
End

Algorithm 5.2: *Global Best Steered Quantum Inspired Cuckoo Search Algorithm*

To evaluate the performance of the proposed method, we compared the performance of GQICSA, QICSA and conventional CSA with 16 efficient benchmark functions (Naik et al., 2015). These benchmark functions can be categorized into three types: unimodal, multimodal with variable dimension and multimodal with fixed dimension. Among the 16 benchmark functions, the unimodal test functions are Sphere model ($F1$) (Jamil & Yang, 2013; Naik et al., 2015), Schwefel's problem 2.22 ($F2$) (Jamil & Yang, 2013; Naik et al., 2015), Schwefel's problem 1.2 ($F3$) (Electric Power Systems Analysis and Nature Inspired

Optimization Algorithms, 2015a; Naik et al., 2015), Schwefel's problem 2.21 ($F4$) (Jamil & Yang, 2013; Naik et al., 2015), generalized Rosenbrock's function ($F5$) (Naik et al., 2015;), Step function ($F6$) (Electric Power Systems Analysis and Nature Inspired Optimization Algorithms, 2015b; Naik et al., 2015) and quartic function, i.e. noise ($F7$) (Jamil & Yang, 2013; Naik et al., 2015). The multimodal test functions with variable dimension are generalized Schwefel's problem 2.26 ($F8$) (Electric Power Systems Analysis and Nature Inspired Optimization Algorithms, 2015c; Naik et al., 2015), generalized Rastrigin's function ($F9$) (Surjanovik & Bingham, 2013; Naik et al., 2015), Ackley's function ($F10$) (Surjanovik & Bingham, 2013; Naik et al., 2015) and generalized Griewank function ($F11$) (Surjanovik & Bingham, 2013; Naik et al., 2015). The multimodal test functions with fixed dimension are Shekel's foxholes function (F12) (Electric Power Systems Analysis and Nature Inspired Optimization Algorithms, 2015d; Naik et al., 2015), Kowalik's function (F13) (Naik et al., 2015), Six-hump Camelback function (F14) (Surjanovik & Bingham, 2013; Naik et al., 2015), Branin function (F15) (Surjanovik & Bingham, 2013; Naik et al., 2015), Goldstein-Price function (F16) (Surjanovik & Bingham, 2013; Naik et al., 2015). The unimodal ($F1$ - $F7$), multimodal test functions with variable dimension ($F8$ – $F11$) and multimodal test functions with fixed dimension (F12 – F16) are considered to have for performance evaluation. Parameters for CSA and GQICSA were considered as follows:

Size of initial population (n) = 25

Probability for discovering bad quality nests (p_a) = 0.25

Total no. of iterations ($iter$)=2000

Table 5.1: Details of Unimodal Benchmark Functions

Name	Benchmark Function	Search Range
Sphere model (f_1) (Surjanovik & Bingham, 2013; Naik et al., 2015)	$f_1(x) = \sum_{i=1}^{D} x_i^2$	$[-100,100]^D$
Schwefel's problem 2.22 (f_2) (Jamil & Yang, 2013; Naik et al., 2015)	$f_2(x) = \sum_{i=1}^{D} \lvert x_i \rvert + \prod_{i=1}^{D} \lvert x_i \rvert$	$[-10,10]^D$
Schwefel's problem 1.2 (f_3) (Naik et al., 2015;)	$f_3(x) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-100,100]^D$
Schwefel's problem 2.21 (f_4) (Jamil & Yang, 2013; Naik et al., 2015)	$f_4(x) = max \lvert x_i \rvert \, ; i = 1,2, \dots . D$	$[-100,100]^D$
Generalized Rosenbrock's function (f_5) (Surjanovik & Bingham, 2013; Naik et al., 2015)	$f_5(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i)^2 + (x_i - 1)^2]$	$[-30,30]^D$
Step function (f_6) (Naik et al., 2015;)	$f_6(x) = \sum_{i=1}^{D} \lfloor x_i + 5 \rfloor^2$	$[-100,100]^D$
Quartic function (f_7) (Jamil & Yang, 2013; Naik et al., 2015)	$f_7(x) = \sum_{i=1}^{D} i x_i^4 + random[0,1]$	$[1.28,1.28]^D$

Table 5.2: Details of Multimodal Benchmark Functions with variable dimension

Name	Benchmark Function	Search Range
Schwefel's problem 2.26 (f_8) (Naik et al., 2015)	$f_8(x) = -\sum_{i=1}^{D} x_i (\sqrt{\lvert x_i \rvert})$	$[-500,500]^D$
Generalized Rastrigin's function (f_9) (Surjanovik & Bingham, 2013; Naik et al., 2015)	$f_9(x) = \sum_{i=1}^{D-1} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[5.12,5.12]^D$

Ackley's function (f_{10}) (Surjanovik & Bingham, 2013; Naik et al., 2015)	$f_{10}(x) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	[-32,32]^D
Generalized Griewank function (f_{11}) (Surjanovik & Bingham, 2013; Naik et al., 2015)	$f_{11}(x) = \sum_{i=1}^{D}\frac{x_i^2}{4000} - \prod_{i=1}^{D}\cos\left(\frac{x_i}{i}\right) + 1$	[-600,600]^D

Table 5.3: Details of Multimodal Benchmark Functions with fixed dimension

Name	**Benchmark Function**	**Search Range**
Shekel's foxholes function (f_{12}) (Naik et al., 2015;)	$f_{12}(x) = \left[\frac{1}{500}\sum_{j=1}^{25}\frac{1}{j + \sum_{i=1}^{2}(x_i - a_{i,j})^6}\right]^{-1}$	[-65.536 , 65.536]^2
Kowalik's function (f_{13}) (Naik et al., 2015)	$f_{13}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	[-5,5]^4
Six-hump Camelback function (f_{14}) (Surjanovik & Bingham, 2013; Naik et al., 2015)	$f_{14}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	[-5,5]^2
Branin function (f_{15}) (Surjanovik & Bingham, 2013; Naik et al., 2015)	$f_{15}(x) = 1(x_2 - x_1^2\left(\frac{5.1}{4\pi^2}\right) + x_1\left(\frac{5}{\pi}\right) - 6)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos(x_1) + 10$	[[-5,0],[10,15]]^2
Goldstein-Price function (f_{16}) (Surjanovik & Bingham, 2013; Naik et al., 2015)	$f_{16}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)][30 + (2x_1 - 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2]$	[-2,2]^2

Table 5.4: Performance Evaluation using Benchmark Functions

Function	Algorithm	Best	Mean	Std.	Avg. Time
F_1	GQICSA	1.7194e-40	3.3645e-37	1.4765e-36	1.7448
	QICSA	4.9152e-16	6.2337e-15	6.0275e-15	3.0932
	CSA	5.1925e-11	3.6508e-10	3.6669e-10	3.3126
F_2	GQICSA	1.2881e-23	1.6149e-21	3.3248e-21	1.8510
	QICSA	4.4884e-09	1.2879e-08	8.2027e-09	3.2634
	CSA	2.4363e-05	6.0125e-05	2.6264e-05	3.4660
F_3	GQICSA	1.7145e-05	7.1186e-04	7.9723e-04	5.1074
	QICSA	2.1979e-04	0.0018	0.0021	9.8202
	CSA	2.5935	14.0311	10.6556	9.9447
F_4	GQICSA	3.0422e-04	0.0058	0.0049	1.8940
	QICSA	1.0956	5.2957	2.5149	1.9046
	CSA	3.7385	6.5092	1.6751	2.0811
F_5	GQICSA	0.0015	17.8455	19.4371	1.9516
	QICSA	0	0	0	3.2160
	CSA	16.1695	33.8772	24.8391	2.5685
F_6	GQICSA	0	0	0	1.7691
	QICSA	0	0	0	3.2160
	CSA	0	0	0	3.4286
F_7	GQICSA	0.0034	0.0103	0.0036	2.5483
	QICSA	0.0172	0.0250	0.0076	3.8422
	CSA	0.0098	0.0308	0.0145	4.0255
F_8	GQICSA	-1.2198e+04	-1.1169e+04	475.7058	2.0582
	QICSA	-9.4685e+03	-9.0397e+03	358.4852	3.5953
	CSA	-9.6111e+03	-9.2550e+03	241.0039	3.7570
F_9	GQICSA	7.0931	18.0660	6.5278	1.9099
	QICSA	20.0508	29.8919	7.8986	3.4785
	CSA	29.5893	53.3390	11.2181	3.6830
F_{10}	GQICSA	7.9936e-15	1.8420e-09	1.8419e-08	1.9330
	QICSA	5.5159e-08	1.3441	0.6901	3.3927
	CSA	3.2896e-04	0.4842	0.6344	3.6742
F_{11}	GQICSA	0	0.0060	0.0099	2.0691
	QICSA	1.7764e-15	0.0049	0.0070	1.2696
	CSA	5.1778e-09	4.9655e-05	1.5445e-04	3.8954

F_{12}	**GQICSA**	0.9980	0.9980	2.5664e-15	6.0168
	QICSA	0.9980	0.9980	2.5664e-15	6.3875
	CSA	0.9980	0.9980	2.5664e-15	6.6974
F_{13}	**GQICSA**	3.0749e-04	3.6243e-04	2.1856e-04	0.9694
	QICSA	3.0749e-04	3.0749e-04	9.5079e-19	1.0312
	CSA	3.0749e-04	4.3568e-04	3.1933e-04	1.2120
F_{14}	**GQICSA**	-1.0316	-1.0316	1.5434e-15	1.3136
	QICSA	-1.0316	-1.0316	1.5434e-15	1.5524
	CSA	-1.0316	-1.0316	2.0919e-15	0.8608
F_{15}	**GQICSA**	0.3979	0.3979	1.0600e-15	0.8142
	QICSA	0.3979	0.3979	1.0600e-15	1.1250
	CSA	0.3979	0.3979	4.3543e-10	1.1461
F_{16}	**GQICSA**	3.0000	3.0000	9.4470e-16	0.8120
	QICSA	3.0000	3.0000	9.4470e-16	1.1414
	CSA	3.0000	3.0000	7.3745e-16	1.1418

For a single test function each algorithm was executed for 100 times. For comparing the performance of CSA, QICSA and GQICSA, best minima/maxima ('Best'), mean ('Mean'), standard deviation ('Std'), and average time ('Avg. Time') to get the best result in an evaluation of 100 independent runs are shown in TABLE 5.4. The convergence curves of some benchmark functions are shown in Fig. 5.3(a)-(f). The experimental results suggest that GQICSA outperforms CSA and QICSA in terms of 'Best', 'Mean', and 'Std.' for all the benchmark functions.

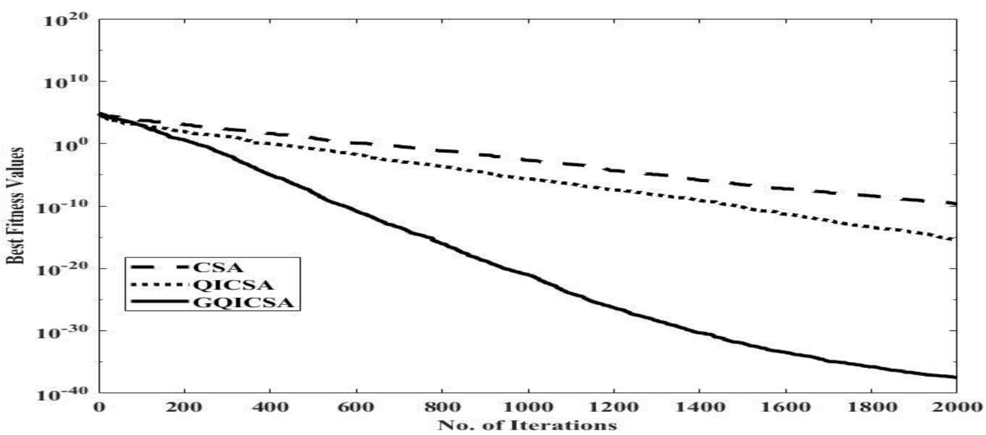

Figure 5.3(a)

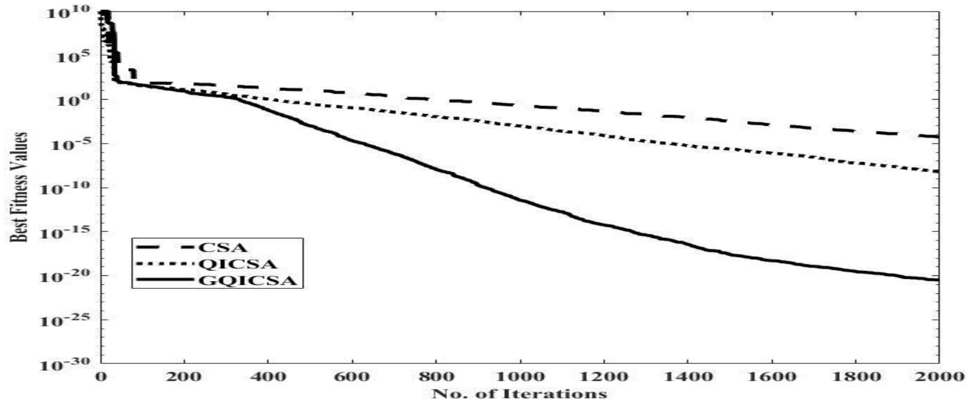

Figure 5.3(b)

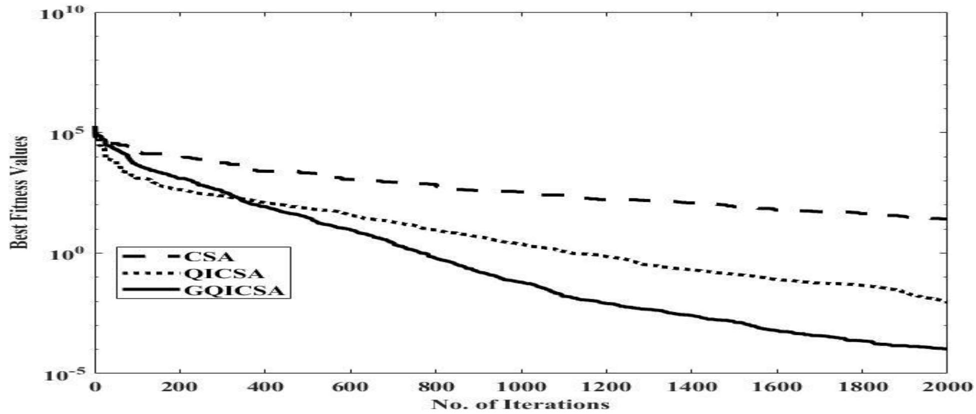

Figure 5.3(c)

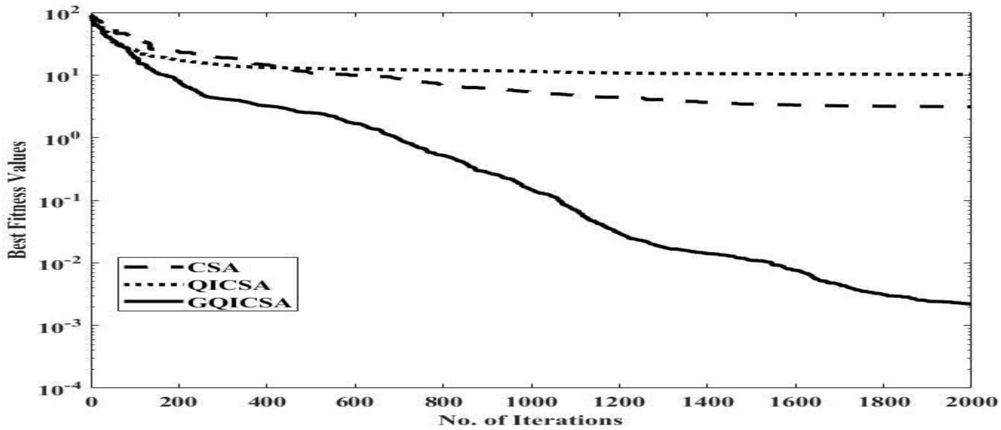

Figure 5.3(d)

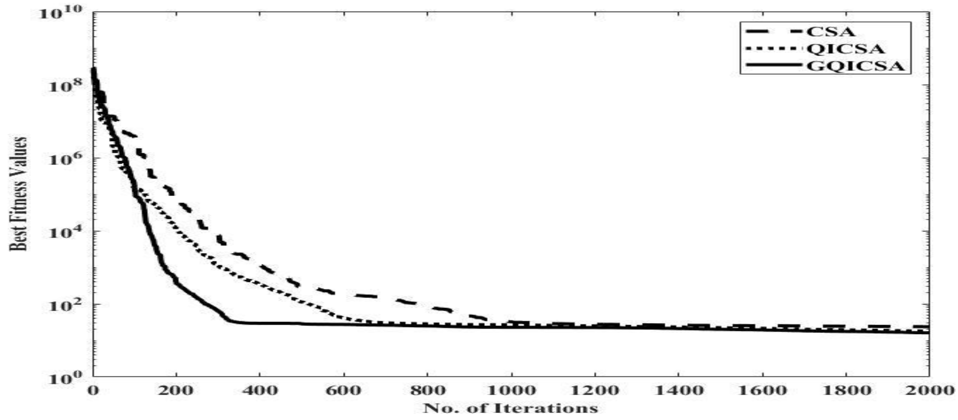

Figure 5.3(e)

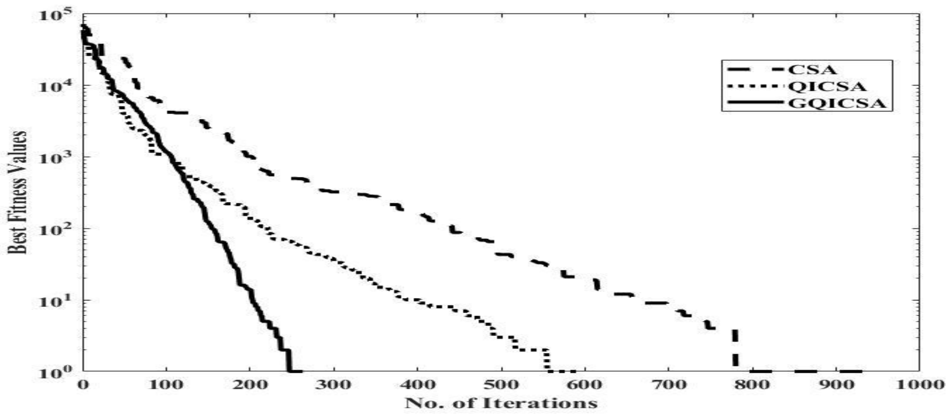

Figure 5.3(f)

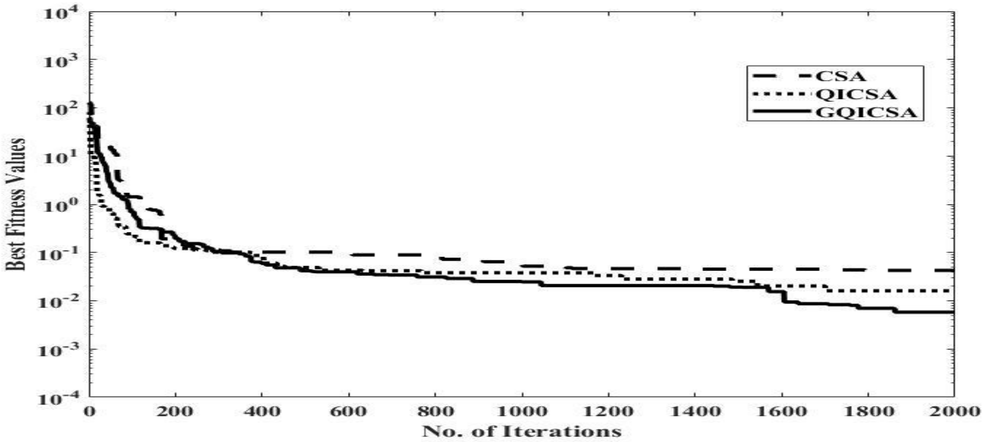

Figure 5.3(g)

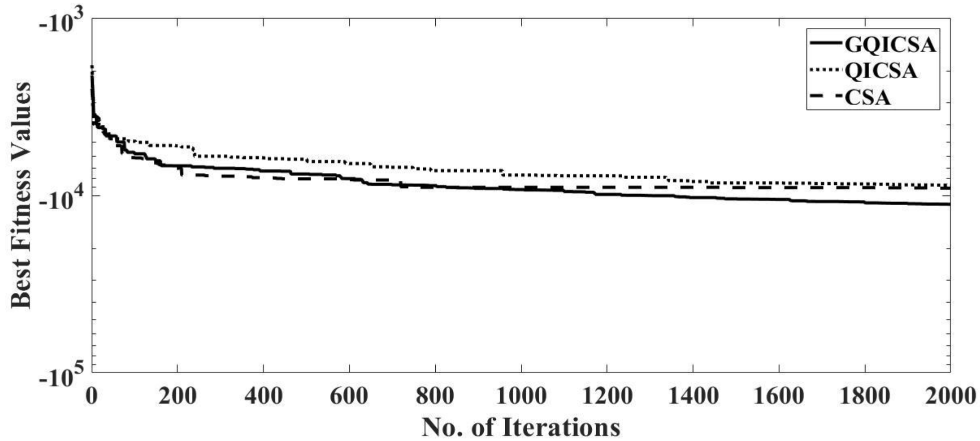

Figure 5.3(h)

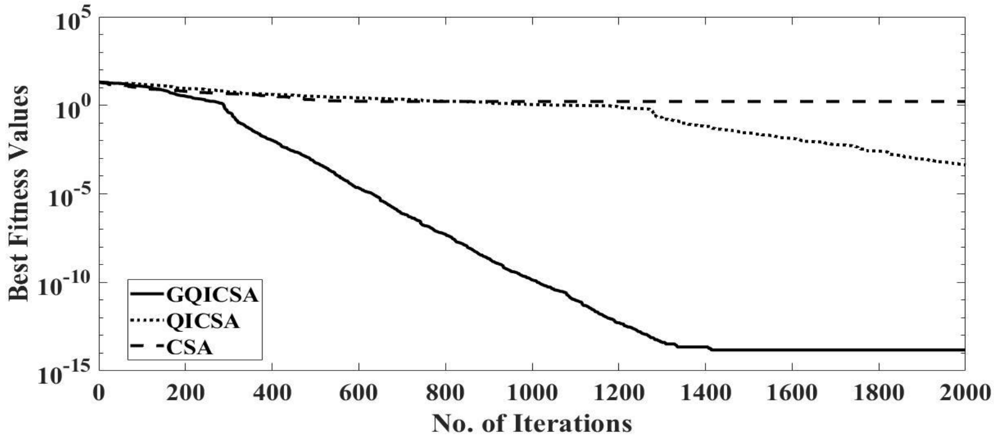

Figure 5.3(i)

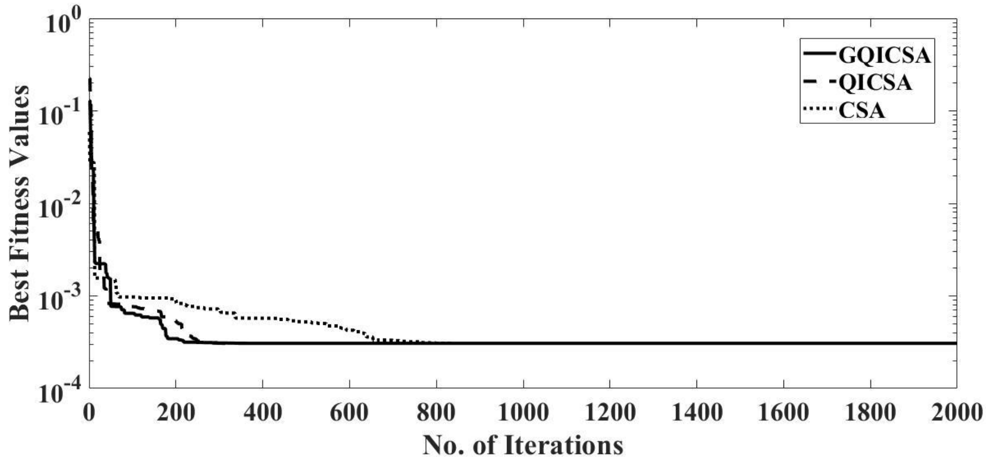

Figure 5.3(j)

Figure 5.3: Fitness Plots for Benchmark Functions (a) Sphere model, (b) Schwefel's problem 2.22, (c) Schwefel's problem 1.2, (d) Schwefel's problem 2.21 (e) Generalized Rosenbrock's function, (f) Step Function, and (g) Quartic function, (h) Schwefel's problem 2.26, (i) Ackley's function, (j) Kowalik's function

5.5. Hardware Efficient FIR Filter Design using Global Best Steered Quantum Inspired Cuckoo Search Algorithm

To design FIR filters using GQICSA or QICSA firstly the filter order (N), word length of filter coefficients (B), passband frequency (ω_p), stopband frequency(ω_s), passband ripple (δ_p), stopband ripple(δ_s), total no. of iterations ($iter_{MAX}$), size of the population (n), lower bound (L_B), upper bound (U_B), switch probability (p_m), and scaling factor (γ) must be specified. Size of each candidate solution will be $d = \lfloor (N/2) + 1 \rfloor$. The primary step of the algorithm is to generate an initial population randomly, but within a specific range demarcated by the lower and upper bounds. In our proposed technique, each host nest specifies the coefficient set of a symmetric FIR filter. Value of each coefficient is bounded in the range of (-1, 1). Fitness value of each solution is computed in terms of the quality of the implemented filter using Equation 5. Finally after obtaining an optimized set of filter coefficients, each coefficient in the set is abbreviated to the specified word length (B). Abbreviated coefficients set are then used to design the FIR filter. Quality of the filter is evaluated in terms of filter responses, i.e., stopband attenuation and passband ripple. Total number of SPT terms present in the coefficients set and the required number of adders to implement the filter are also computed to evaluate the designed filter. Performances of even order and odd order symmetric lowpass FIR filters designed using QICSA and GQICSA are evaluated. We compared the results of the proposed system with the results obtained with the standard Cuckoo Search Algorithm and few other optimization algorithms like GA, PSO, QPSO and FPA. Responses of the filters designed using QICSA and GQICSA were also compared with the results of conventional algorithmic strategy of filter design with Parks McClellan Algorithm. A comparative study of the designed filters in terms of Stopband attenuation (Asb) and Passband ripple $\left(\delta_p\right)$ is presented in TABLE V and VI, for even and odd order symmetric lowpass FIR filters, respectively. To design each of the 16th, 17th, 20th and 21st order lowpass filters using each of the above mentioned optimization algorithms, total numbers of iterations were fixed at 1000. Similarly, for implementation of 24th and 25th order lowpass filters, total numbers of iterations were fixed at 1500. In both cases, size of the initial population is considered as 30. For all the lowpass filters, Passband edge frequency (ω_p) is taken as 0.4π, Stopband edge frequency (ω_s) is taken as 0.6π, Passband ripple (δ_p) is fixed at 0.05 and Stopband ripple is fixed at (δ_s) 0.03.

5.5.1. Simulation Results

Table 5.5: Performance Comparison of different algorithms used to design even order filters

Algorithm	**Filter Order(N)**					
	16		**20**		**24**	
	Asb(dB)	$\boldsymbol{\delta_p}$	**Asb(dB)**	$\boldsymbol{\delta_p}$	**Asb(dB)**	$\boldsymbol{\delta_p}$
PM	32.6684	0.41	38.9590	0.20	45.1764	0.10
GA	4.1150	1.35	5.8206	0.79	2.1524	2.35
PSO	11.1904	3.01	12.6466	0.90	5.1398	2.22
QPSO	18.7527	0.54	12.9994	0.94	11.2887	3.53
FPA	28.9095	0.30	30.9035	0.28	41.5362	0.12
CSA	34.8293	1.80	33.7077	1.48	22.0503	0.49
QICSA	36.0029	0.38	40.0104	0.30	45.2146	0.17
GQICSA	**36.0035**	**0.38**	**40.7590**	**0.29**	**45.2160**	**0.17**

Table 5.6: Performance Comparison of different algorithms used to design odd order filters

Algorithm	**Filter Order(N)**					
	17		**21**		**25**	
	Asb(dB)	$\boldsymbol{\delta_p}$	**Asb(dB)**	$\boldsymbol{\delta_p}$	**Asb(dB)**	$\boldsymbol{\delta_p}$
PM	35.1755	0.31	41.4227	0.14	47.6300	0.07
GA	3.5660	3.43	4.0228	3.58	3.5722	1.91

PSO	6.6600	2.89	9.6912	5.64	5.2304	2.97
QPSO	7.1994	4.09	9.7021	6.29	11.5722	3.23
FPA	29.2285	0.24	39.5807	0.17	34.0203	0.19
CSA	32.7543	0.22	24.6848	0.34	13.8709	0.56
QICSA	38.5122	0.37	42.6822	0.22	47.9356	0.12
GQICSA	**38.6310**	**0.37**	**43.0010**	**0.22**	**48.0280**	**0.12**

From the comparative study presented in TABLE 5.5 & 5.6, it is evident that the most conventional algorithms, GA and PSO, could not achieve good results with such less number of iterations like 1000 and 1500, and even a modified version of traditional PSO, Quantum Behaved Particle Swarm Optimization (QPSO) could not accomplish good results in such a fewer number of iterations. Hence, in Figure 5.4(a)-(f), responses of the lowpass Type I & Type II FIR filters implemented using only GQICSA, QICSA, CSA, FPA and PMA are shown.

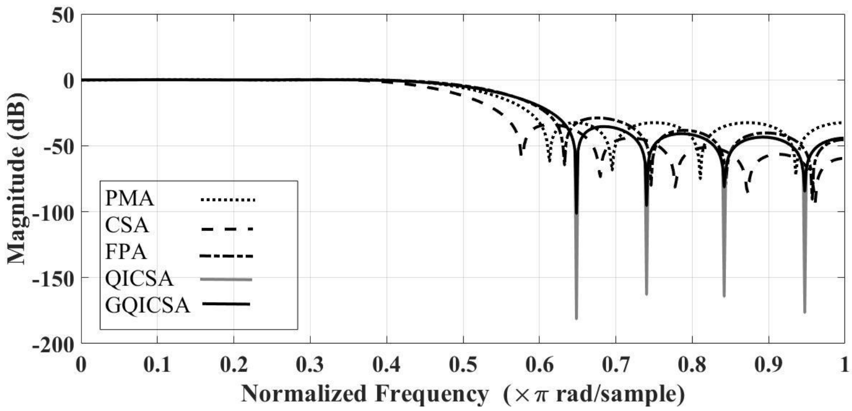

Figure 5.4(a)

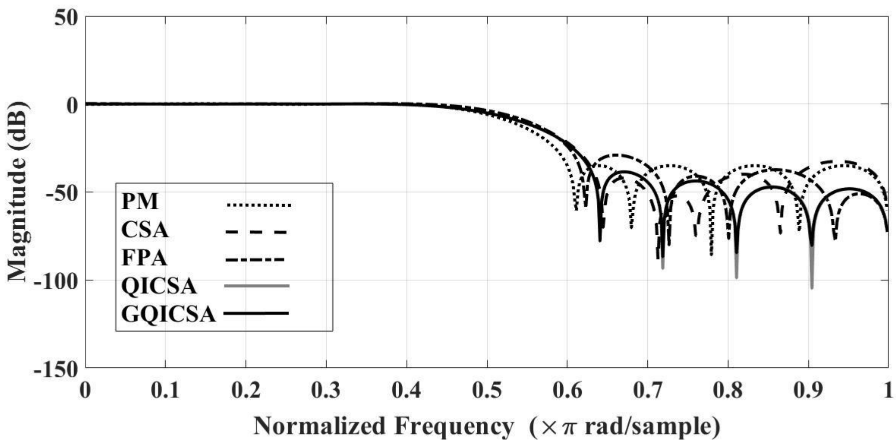

Figure 5.4(b)

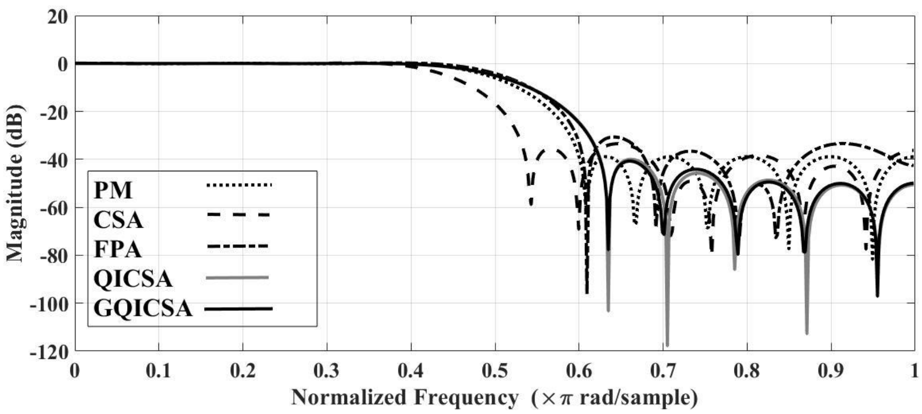

Figure 5.4(c)

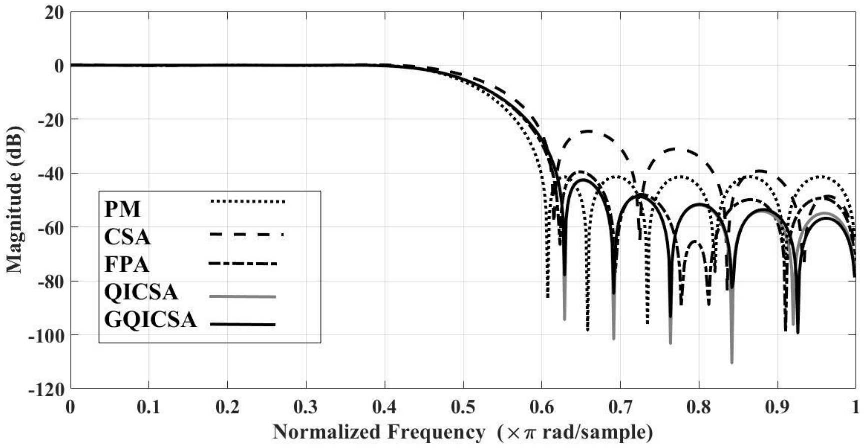

Figure 5.4(d)

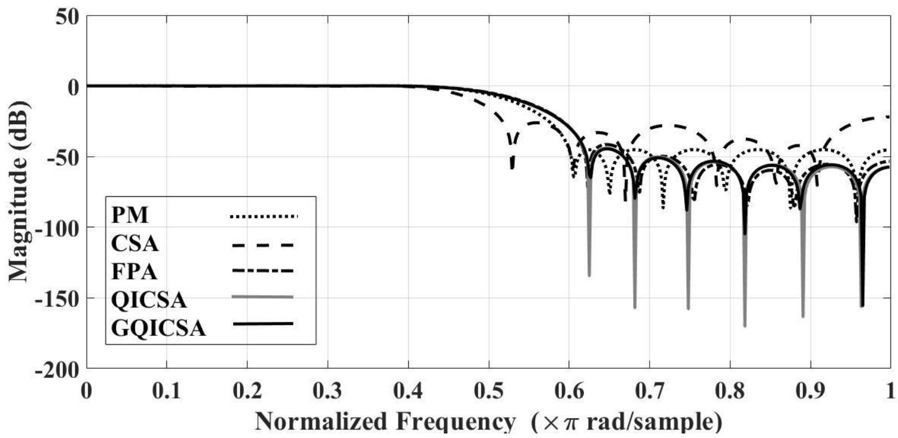

Figure 5.4(e)

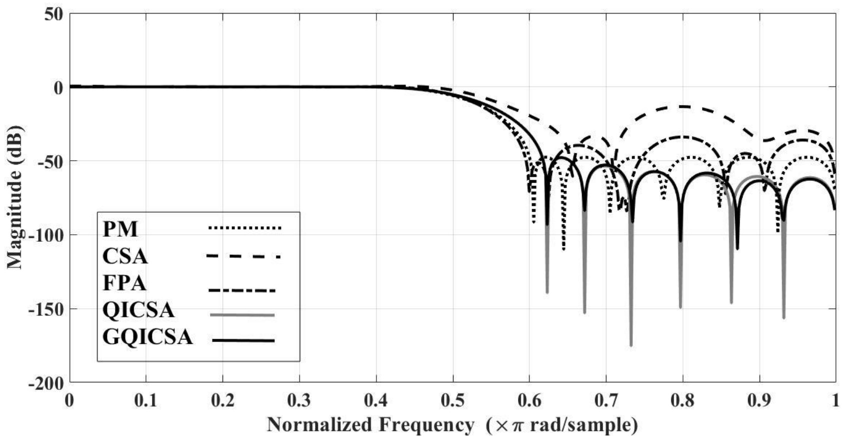

Figure 5.4(f)

Figure 5.4: Magnitude Responses of the filters with (a) order 16, (b) order 17, (c) order 20, (d) order 21, (e) order 24, and (f) order 25

Next, the responses of the filters implemented using the optimized set of coefficients, but with fixed word length is presented. Stopband attenuation and adder costs for implementing different order filters using the coefficients obtained by different algorithms but with suppressed word length are presented in Table 5.7-5.12.

Table 5.7: Performance comparison of 16th order filters in terms of Stopband Attenuation, Passband Ripple and Hardware Cost

Filters	**Filter Order (N)**	**WL**	**Asb (dB)**	**Asb Gain (%)**	**Pass Ripple**	**SPT**	**SPT Gain (%)**	**MA**	**SA**	**TA**	**TA Gain (%)**
PM	16	10	30.2660		0.37	24		9	10	19	
FPA	16	10	28.7172	-5.40	0.34	25	-4	9	10	19	0
CSA	16	10	33.1002	8.56	1.95	28	-14.28	9	15	24	-20.83
QICSA1	16	10	35.9531	15.81	0.54	24	0	9	12	21	-9.52
QICSA2	16	9	35.9502	15.81	0.53	18	33.33	8	9	17	11.76
GQICSA1	**16**	**10**	**36.1056**	**16.17**	**0.57**	**24**	**0**	**10**	**9**	**19**	**0**
GQICSA2	**16**	**9**	**36.2336**	**16.46**	**0.63**	**18**	**33.33**	**9**	**8**	**17**	**11.76**

Table 5.8: Performance comparison of 17th order filter in terms of Stopband Attenuation, Passband Ripple and Hardware Cost

Filters	**Filter Order (N)**	**WL**	**Asb (dB)**	**Asb Gain (%)**	**Pass Ripple**	**SPT**	**SPT Gain (%)**	**MA**	**SA**	**TA**	**TA Gain (%)**
PM	17	10	35.0032		0.33	29		9	15	24	
FPA	17	10	29.6679	-17.98	0.20	25	16.00	9	14	23	4.34
CSA	17	10	21.9020	-59.81	0.21	28	3.57	9	14	23	4.34
QICSA1	17	10	38.1414	8.23	0.47	27	7.40	9	13	22	9.09
QICSA2	17	9	41.0075	14.64	0.49	23	26.08	8	10	18	33.33
GQICSA1	**17**	**10**	**39.7882**	**13.67**	**0.51**	**27**	**7.40**	**9**	**12**	**21**	**14.28**
GQICSA2	**17**	**9**	**39.7882**	**13.67**	**0.52**	**20**	**45**	**8**	**9**	**17**	**41.17**

Table 5.9: Performance comparison of 20th order filters in terms of Stopband Attenuation, Passband Ripple and Hardware Cost

Filters	Filter Order (N)	WL	Asb (dB)	Asb Gain (%)	Pass Ripple	SPT	SPT Gain (%)	MA	SA	TA	TA Gain (%)
PM	20	10	36.6763		0.16	23		9	11	20	
FPA	20	10	31.6205	-16.00	0.24	26	-11.54	9	13	22	-9.09
CSA	20	10	33.6580	-8.97	1.51	36	-36.11	9	16	25	-20
QICSA1	20	10	41.7492	12.15	0.36	21	9.52	9	9	18	11.11
QICSA2	20	9	42.0520	12.78	0.40	17	35.30	8	7	15	33.33
GQICSA1	**20**	**10**	**42.6568**	**14.02**	**0.29**	**21**	**9.52**	**9**	**9**	**18**	**11.11**
GQICSA2	**20**	**9**	**42.2371**	**13.16**	**0.33**	**19**	**21.05**	**8**	**7**	**15**	**33.33**

Table 5.10: Performance comparison of 21st order filters in terms of Stopband Attenuation, Passband Ripple and Hardware Cost

Filters	**Filter Order (N)**	**WL**	**Asb (dB)**	**Asb Gain (%)**	**Pass Ripple**	**SPT**	**SPT Gain (%)**	**MA**	**SA**	**TA**	**TA Gain (%)**
PM	21	10	40.1740		0.16	30		9	14	23	
FPA	21	10	36.2762	-10.74	0.27	28	7.14	9	12	21	9.52
CSA	21	10	25.7162	-56.22	0.31	30	0	9	17	26	-11.54
QICSA1	21	10	40.8480	1.65	0.33	27	11.11	9	13	22	4.54
QICSA2	21	9	36.5475	-9.91	0.30	22	36.36	8	11	19	21.05
GQICSA1	**21**	**10**	**40.9766**	**1.95**	**0.26**	**29**	**3.44**	**9**	**13**	**22**	**4.54**
GQICSA2	**21**	**9**	**40.1875**	**0.03**	**0.25**	**24**	**25**	**8**	**11**	**19**	**21.05**

Table 5.11: Performance comparison of 24th order filters in terms of Stopband Attenuation, Passband Ripple and Hardware Cost

Filters	Filter Order (N)	WL	Asb (dB)	Asb Gain (%)	Pass Ripple	SPT	SPT Gain (%)	MA	SA	TA	TA Gain (%)
PM	24	10	42.5662		0.18	24		8	10	18	
FPA	24	10	38.5329	-10.46	0.15	23	4.34	7	7	14	28.57
CSA	24	10	21.5943	-97.11	0.53	37	-35.13	9	20	29	-37.93
QICSA1	24	10	43.4314	2.00	0.35	31	-22.50	9	13	22	-18.18
QICSA2	24	9	42.8260	0.60	0.23	20	20.00	8	8	16	12.50
GQICSA1	**24**	**10**	**43.8460**	**0.22**	**2..91**	**31**	**-22.50**	**9**	**9**	**18**	**0**
GQICSA2	**24**	**9**	**42.9380**	**0.35**	**0.86**	**20**	**20.00**	**8**	**8**	**16**	**12.50**

Table 5.12: Performance comparison of 25th order filters in terms of Stopband Attenuation, Passband Ripple and Hardware Cost

Filters	Filter Order (N)	WL	Asb (dB)	Asb Gain (%)	Pass Ripple	SPT	SPT Gain (%)	MA	SA	TA	TA Gain (%)
PM	25	10	42.6105		0.08	35		9	16	25	
FPA	25	10	33.9195	-25.62	0.18	41	-14.63	9	14	23	8.69
CSA	25	10	14.1199	-201.77	0.62	37	-5.40	9	12	21	19.04
QICSA1	25	10	46.6903	8.74	0.15	31	12.90	9	13	22	13.63
QICSA2	25	9	42.8985	0.67	0.21	25	40	8	11	19	31.57
GQICSA1	**25**	**10**	**46.8931**	**9.13**	**0.16**	**31**	**12.90**	**9**	**13**	**22**	**13.63**
GQICSA2	**25**	**9**	**43.1619**	**1.27**	**0.19**	**25**	**40**	**8**	**11**	**19**	**31.57**

Efficiency of QICSA is distinctly proved by our comparative study since QICSA1 means optimized coefficients with word length 10 obtains the least stop band attenuation for all the Type I and Type II lowpass filters among all the filters designed by the coefficients having similar word length. In terms of adder costs and SPT terms, QICSA1 also outperforms the benchmark filters for different orders. QICSA2 with reduced word length of 9 also satisfy the required filter characteristics. Efficacy of GQICSA is proved over QICSA by the comparative study since GQICSA1 means optimized coefficients with word length 10 obtains lesser stop band attenuation for Type I and Type II lowpass filters designed by the coefficients having similar word length. In terms of adder costs and SPT terms, GQICSA1 also outperforms the benchmark filters for different orders. GQICSA2 with reduced word length of 9 also satisfy the required filter characteristics.

Magnitude responses of these filters along with the benchmark filters are shown in Figure 5.5(a) – 5.5(f).

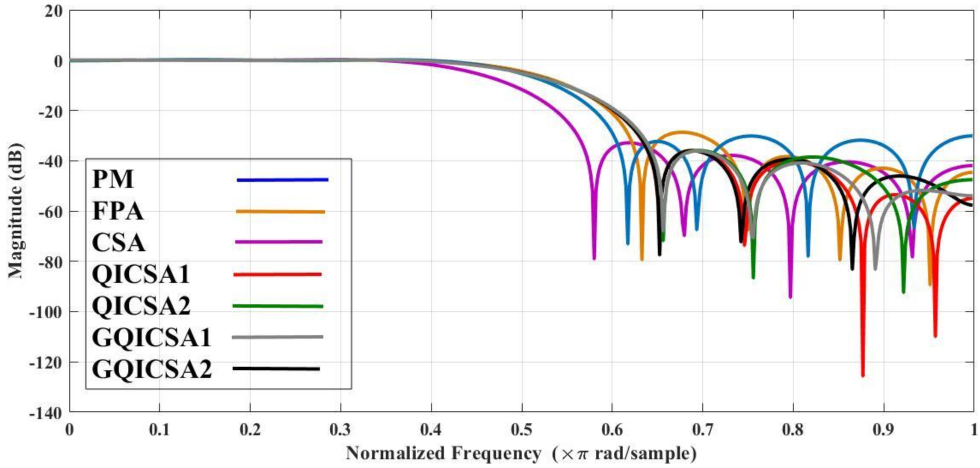

Figure 5.5(a)

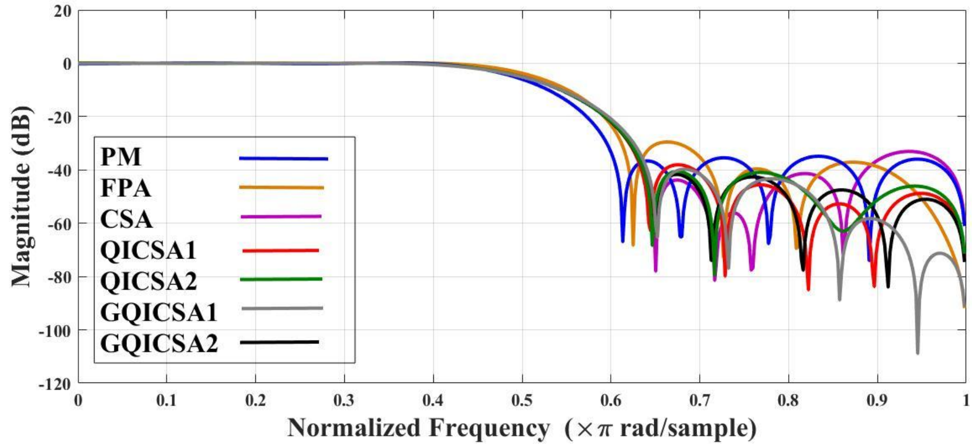

Figure 5.5(b)

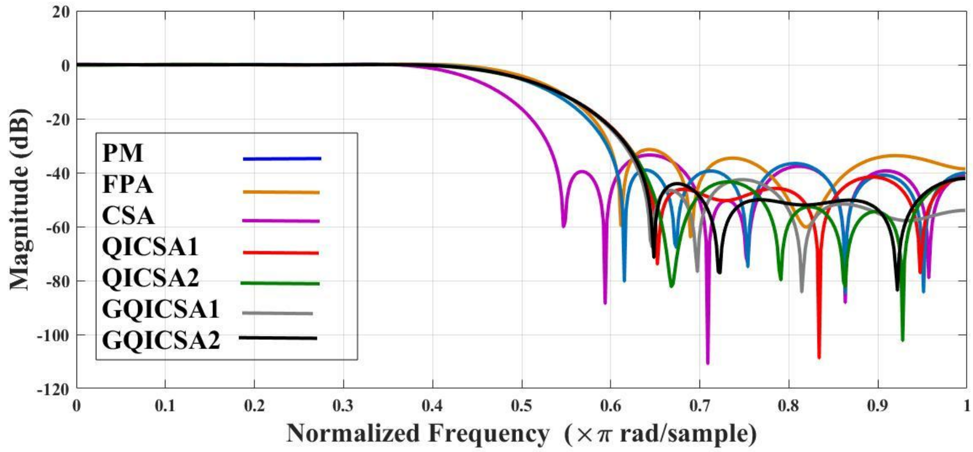

Figure 5.5(c)

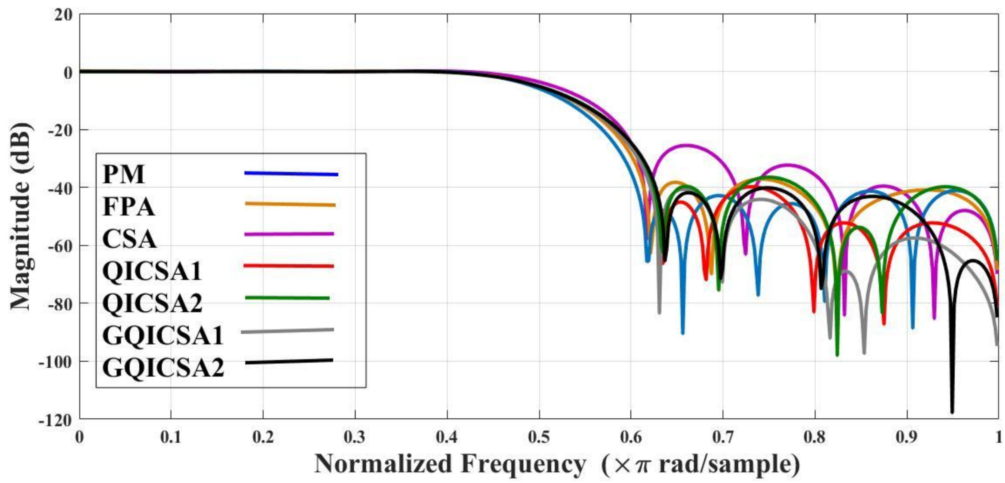

Figure 5.5(d)

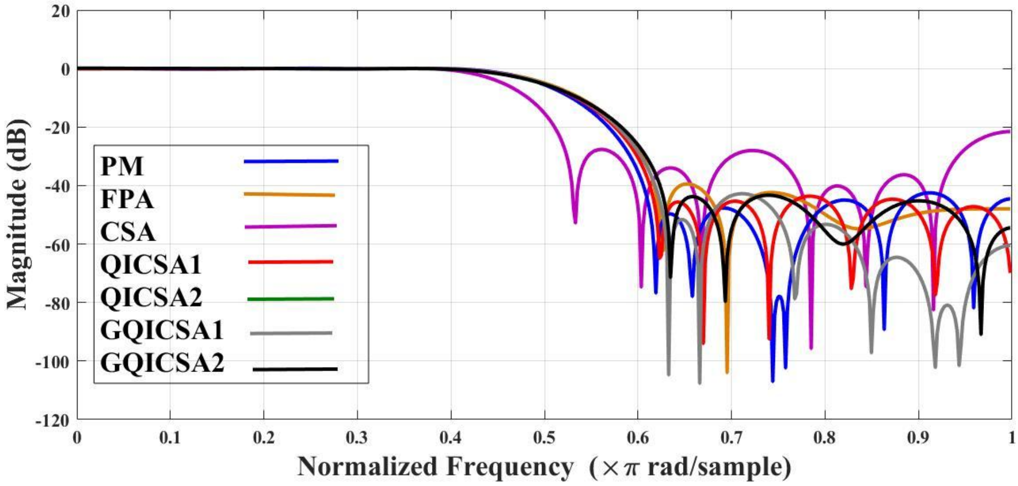

Figure 5.5(e)

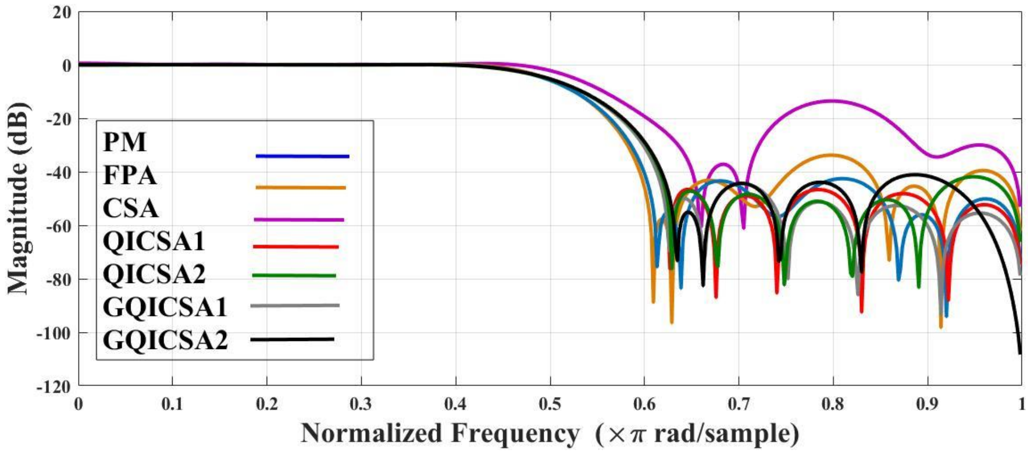

Figure 5.5(f)

Figure 5.5: Magnitude Responses of the filters implemented by the optimized coefficients with reduced word length of (a) order 16, (b) order 17, (c) order 20, (d) order 21, (e) order 24, and (f) order 25

Figure 5.6 shows the column chart representation of the stopband attenuation of different order filters.

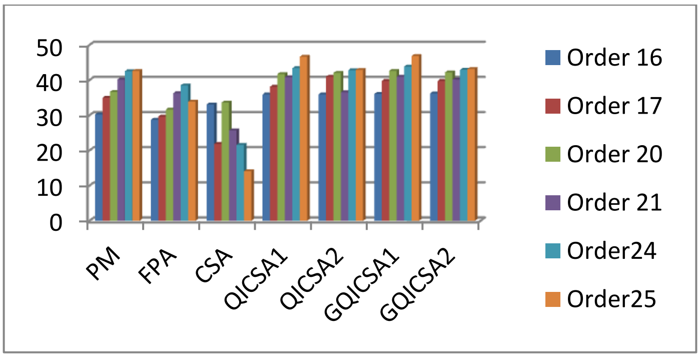

Figure 5.6: Column chart representation of the stopband attenuation of different order filters

Figure 5.7 & 5.8 present the column chart representations of required no. of SPT terms and adders (TA), respectively, for implementing different filters with different orders.

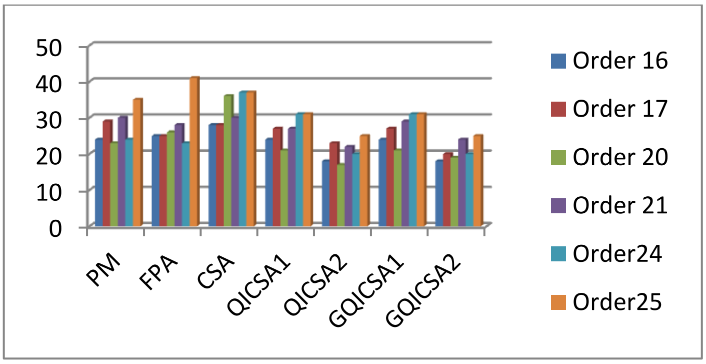

Figure 5.7: Column chart representation of required number of SPT terms to design different filters with different orders

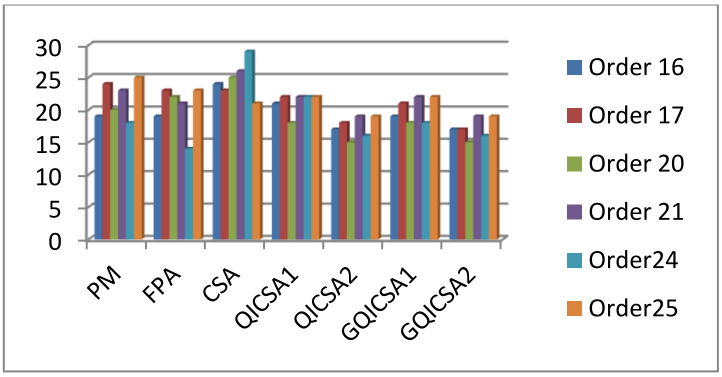

Figure 5.8: Column chart representation of required number of adders (TA) to design different filters with different orders

Figure 5.9(a)–5.9(f) plot the Best Fitness Value vs. Iterations used to obtain coefficients of the filters for different orders using GQICSA, QICSA and CSA.

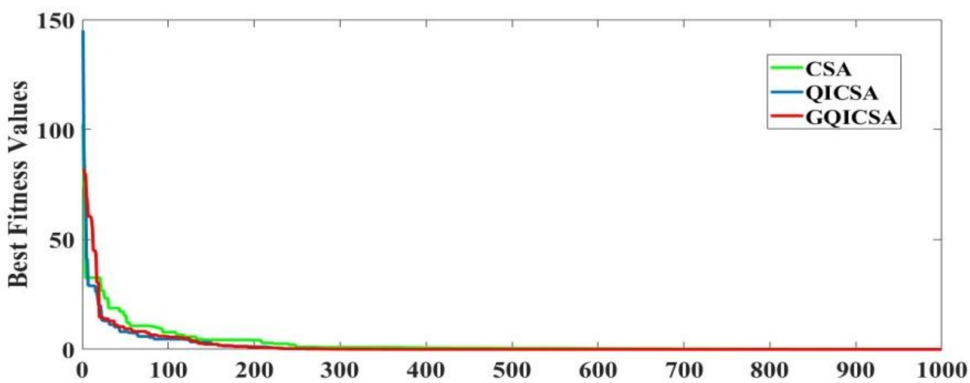

Figure 5.9(a)

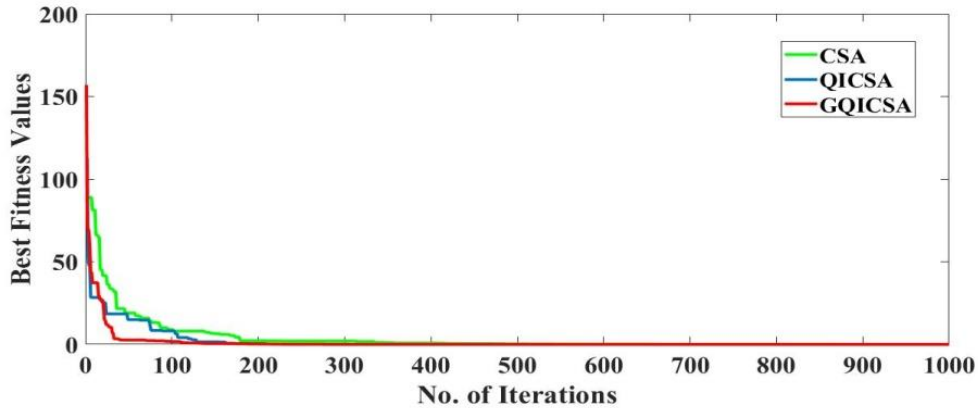

Figure 5.9(b)

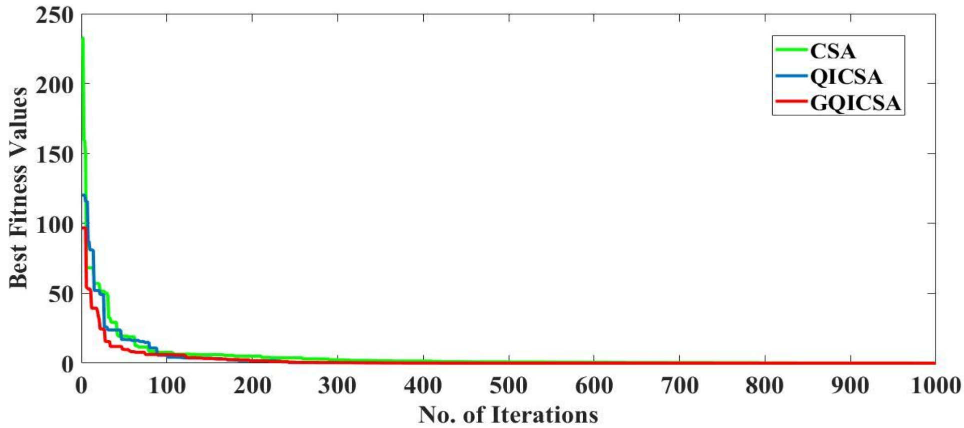

Figure 5.9(c)

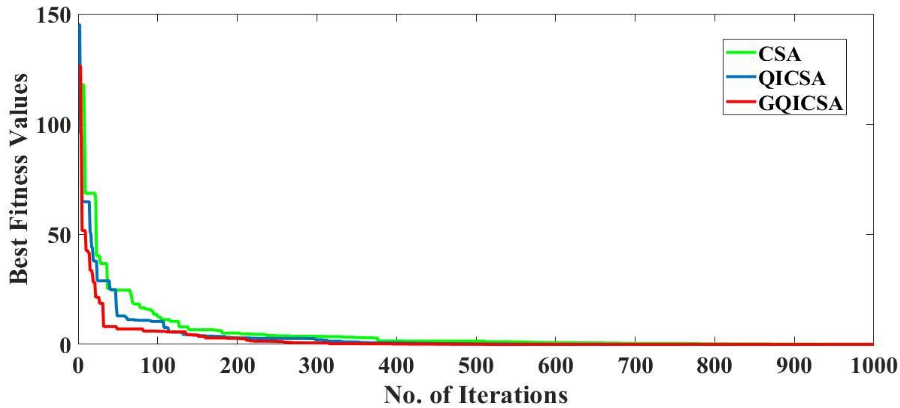

Figure 5.9(d)

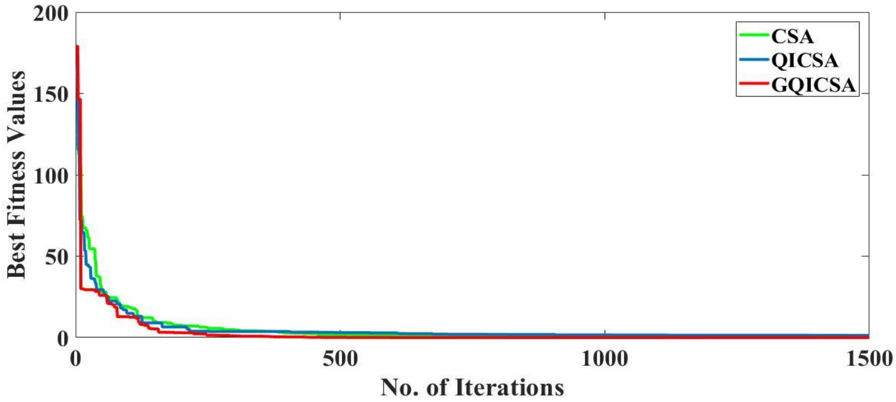

Figure 5.9(e)

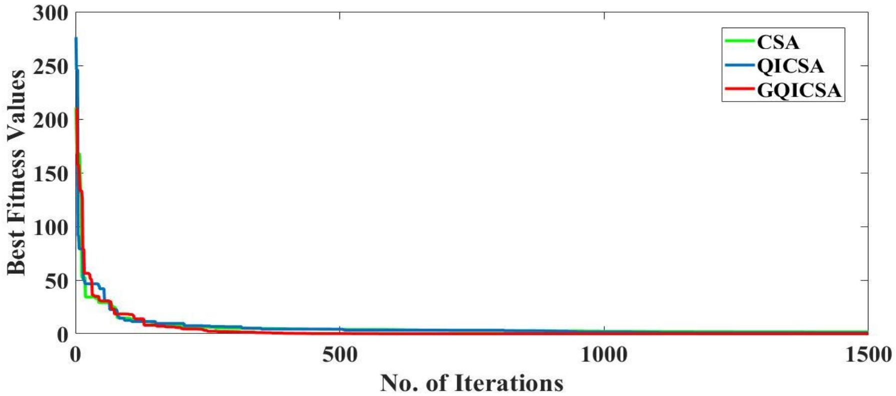

Figure 5.9(f)

Figure 5.9: Plot of Best Fitness Value vs. number of iterations for GQICSA, QICSA and CSA while used to obtain coefficients of the filters of (a) order 16, (b) order 17, (c) order 20, (d) order 21, (e) order 24, (f) order 25

From Figures of 5.9 it is apparent that GQICSA is capable of achieving the best solution in minimum number of iterations rather than QICSA and simple CSA. TABLE XIII reports the time required for execution of the algorithms to obtain optimized coefficients for different order filters and Figure 5.8 shows the column chart representation of the same. It is relevant to mention here that the process is executed by Matlab R2017b in a computer with Intel Core i3 processor and 4GB RAM.

Table 5.13: Execution time of FPA, CSA, QICSA and GQICSA to obtain optimized coefficients for different order filters

Filter	Execution Time			
Order	**FPA**	**CSA**	**QICSA**	**GQICSA**
16	45.4880	43.1593	43.1565	42.1040
17	44.8302	44.4738	42.7976	41.5025
20	45.1742	42.9535	42.7680	41.2205
21	44.7336	44.2608	42.6916	41.1250
24	68.0248	64.4953	64.4931	62.2242
25	67.5020	66.1814	64.0716	62.0028

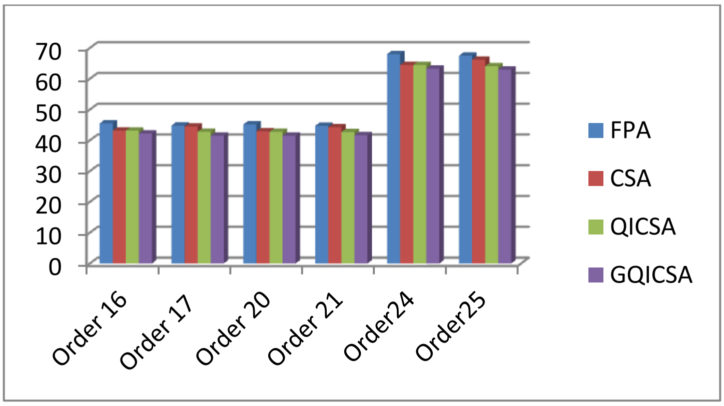

Figure 5.10: Column chart representation of the execution time of FPA, CSA, QICSA and GQICSA to obtain optimized coefficients for different order filters

In the following two tables Table 5.14 and 5.15, a comparative study with few recently proposed optimization algorithms is presented for different orders Type I and II lowpass filters.

Table 5.14: Performance Comparison of different recently proposed algorithms used to design even order filters

Algorithm	**Filter Order(N)**					
	16		**20**		**24**	
	Asb(dB)	δ_p	Asb(dB)	δ_p	Asb(dB)	δ_p
CSA (Yang & Deb, 2010)	34.8293	1.80	33.7077	1.48	22.0503	0.49
GCS (Dhabal & Venkateswaran, 2017)	35.0015	0.40	36.2120	0.45	40.5500	0.20
ACSA (Sengupta & Basak, 2016)	35.0208	0.37	37.0500	0.32	42.0150	0.16
QICSA (Layeb & Boussalia, 2012)	36.0029	0.38	40.0104	0.30	45.2146	0.17
GQICSA	**36.0035**	**0.38**	**40.7590**	**0.29**	**45.2160**	**0.17**

Table 5.15: Performance Comparison of different recently proposed algorithms used to design odd order filters

Algorithm	**Filter Order(N)**					
	17		**21**		**25**	
	Asb(dB)	$\boldsymbol{\delta_p}$	**Asb(dB)**	$\boldsymbol{\delta_p}$	**Asb(dB)**	$\boldsymbol{\delta_p}$
CSA (Yang & Deb, 2010)	32.7543	0.22	24.6848	0.34	13.8709	0.56
GCS (Dhabal & Venkateswaran, 2017)	35.7800	0.25	36.3512	0.25	40.2500	0.30
ACSA (Sengupta & Basak, 2016)	36.5250	0.36	38.2006	0.19	44.1400	0.20
QICSA (Layeb & Boussalia, 2012)	38.5122	0.37	42.6822	0.22	47.9356	0.12
GQICSA	38.6310	0.37	43.0010	0.22	48.0280	0.12

We also analysed performance of the GQICS algorithm for implementing higher order lowpass Type I and Type II filters. Table 5.16 shows a comparative study of the performances of GQICSA, QICSA and CSA to obtain higher order filters. For obtaining filters of order 32 and 40, initial population of size 30 is used and total no. of iterations is set to 3000 whereas for implementing filters of order 48, number of iterations is increased to 4000 and population size is increased by 10. For filter of order 55, population size of 100 and number of iterations of 5000 are used.

Table 5.16: Performance Comparison of GQICSA, QICSA and CSA for higher order filter design

Algorithm	**Filter Order(N)**			
	32	**40**	**48**	**55**
CSA	37.2850	40.1502	42.4412	44.2104
QICSA	46.1050	49.0020	50.5125	52.1500
GQICSA	**50.2032**	**53.2520**	**55.2200**	**58.4160**

From Table 5.16 it can be seen that the two most efficient optimization techniques of the date QICSA and CSA perform well enough to optimize coefficients for higher order filters whereas GQICSA performed excellent in fewer iterations compared to QICSA & CSA to obtain coefficients for higher order filters which can attain high stopband attenuation with negligible passband ripple.

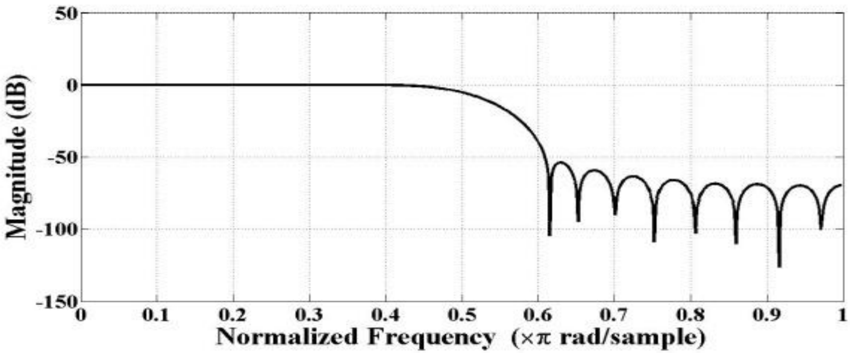

Figure 5.11(a)

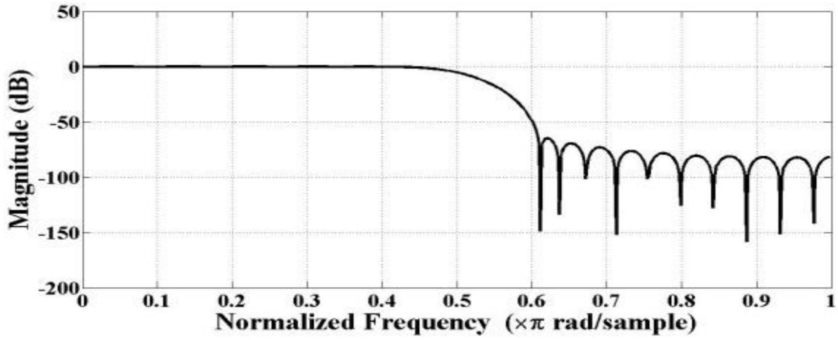

Figure 5.11(b)

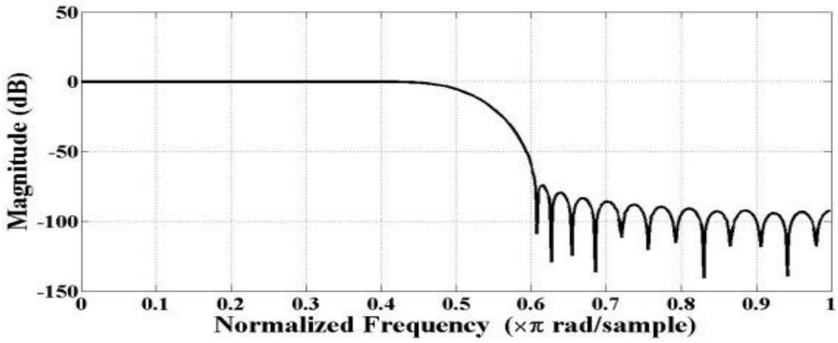

Figure 5.11(c)

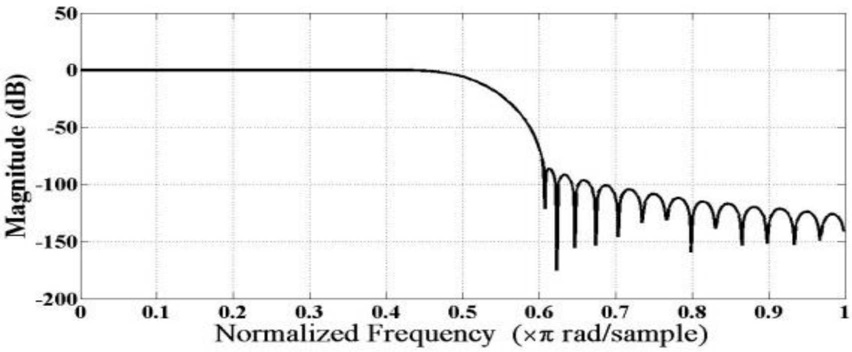

Figure 5.11(d)

Figure 5.11: Magnitude Responses of the filters of (a) order 32, (b) order 40, (c) order 48, and (d) order 55 designed using GQICSA

The Magnitude Responses of the higher order filters designed using GQICSA are shown in Figure 5.11(a)-(d). Table 5.17 shows the execution time required to obtain optimized coefficients to design different higher order filters using CSA, QICSA and GQICSA. Figure 5.12 shows the column chart representation of the same.

Table 5.17: Comparison of Execution Time to design higher order filters using GQICSA, QICSA and CSA

Filter Order	**Execution Time (Seconds)**		
	CSA	QICSA	**GQICSA**
32	69.34	60.2	**55.4**
40	71.26	65.44	**60.5**
48	76.22	70.25	**65.5**
55	78.4	74.23	**68.42**

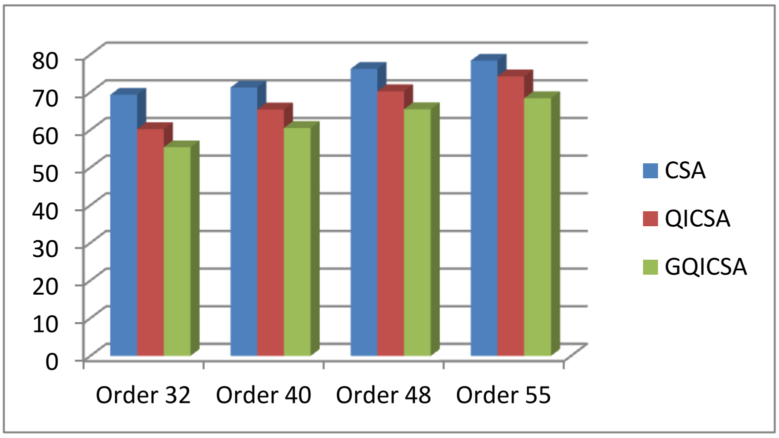

Figure 5.12: Column chart representation of the execution time of CSA, QICSA and GQICSA to obtain optimized coefficients for different higher order filters

Statistical significance test (t-test) (WEB CENTER FOR Social Research Methods, 2006) was performed for the best fitness values for different iterations obtained at the time of implementing 24th and 25th order lowpass filters and the corresponding p values are shown in Table 5.18.

Table 5.18: Statistical test (t-test)

Algorithm	***p* values for GQICSA with respect to other algorithms**	
	Filter Order(N)	
	24	**25**
GA	0.000006	0.00004
PSO	0.00007	0.00006
QPSO	0.0008	0.0007
FPA	0.0005	0.0002
CSA	0.008	0.006
QICSA	0.04	0.04

5.6. Hardware Efficient FIR Filter Design using Fast Converging Flower Pollination Algorithm

5.6.1. Flower Pollination Algorithm

Flower Pollination Algorithm was proposed by Xin She Yang and Suhash Deb in 2012 (Yang & Deb, 2012). Attractive procedure of flow pollination of flowering plants stands as the key motivation for this algorithm. Four basic rules are needed to be followed to implement the algorithm:

(i) Biotic and cross pollination are considered as global pollination, Pollen-carrying pollinator moves obeying Lévy flights.

(ii) Abiotic pollination and self-pollination are used for local pollination.

(iii) Pollinators are mainly insects and they are responsible for development of flower constancy, which is equivalent to a reproduction probability that is proportional to the similarity of two flowers involved.

(iv) Switching between global and local pollination can be controlled by a switch parameter $p \in [0,1]$, slightly biased toward local pollination.

Flower constancy can be described by the following equation.

$$x_i^{t+1} = x_i^t + \gamma L(\lambda)(c_{best} - x_i^t) \tag{5.31}$$

x_i^t denotes the solution vector x_i at iteration t, c_{best} represents the current best solution of the latest generation. Scaling factor γ is used control the step size. $L(\lambda)$is Lévy flights based step size. As the pollinators used to travel over a long distance Lévy flights can be used to mimic the characteristic proficiently. L is computed using Lévy distribution. Local pollination is carried out by the following equation:

$$x_i^{t+1} = x_i^t + \epsilon\left(x_j^t - x_k^t\right) \tag{5.32}$$

x_j^t and x_k^t are pollen from different flowers of a single plant species. If x_j^t and x_k^t come from same spices that means are selected from the same population, it is equivalent to a local random walk if ϵ is drawn from a uniform distribution in $[0,1]$.

Input: Population size (n), Total number of iterations($iter_{MAX}$), lower bound, upper bound, objective function (f), probability (p)
Output: Global best solution $BEST$
Begin
Initialize a population of n flowers within bounds and evaluate the fitness of the solutions using f
Find the best solution and store in $BEST$ and store fitness of the solution $BEST$ in $fmin$
$t = 1$
while ($t < iter_{MAX}$)
  for $i = 1{:}n$
    Compute fitness of solution i and store in f_i^t and generate a random number r
    if($r < p$)
      Draw a step vector L from a Lévy Distribution
      Perform global pollination and update solution using Equation5.31
    Else
      Draw ϵ from a uniform distribution in $[0, 1]$
      Perform local pollination and update solution using Equation 5.32
    End if
    Evaluate new solution and store its fitness in f_i^{t+1}
    if ($f_i^{t+1} < f_i^t$)
      $s_i^{t+1} = x_i^{t+1}$
    else
      $s_i^{t+1} = x_i^t$
    End if
  End for
  Find the current best solution $CBEST$ among the solutions stored in s and its fitness $cmin$
  if ($cmin < fmin$)
    $fmin = cmin$; $BEST = CBEST$
  End if
  $t = t + 1$
End while
End

Algorithm 5.3: *Flower Pollination Algorithm*

5.6.2. Fast Converging Flower Pollination Algorithm

In FPA local random walk is performed using Eqn. 5.32, whereas in Fast converging Flower Pollination Algorithm local random walk is performed using the following Eqn.

$$x_i^{t+1} = x_i^t + \epsilon(CBEST - x_k^t) \tag{5.33}$$

In case of FFPA instead of using permute1 function, $CBEST$ i.e. best nest till the latest iteration is used. This specialty of FFPA keeps the selection pressure towards the better solutions, hence assures better result in fewer iterations. Moreover, this advancement of the Algorithm does not flood the population by the high fitness solutions.

The flowchart in Figure 5.13 briefs the implementation of a FIR filter using Fast converging Flower Pollination Algorithm.

Following filter parameters order of the filter (N), word length of filter coefficients (B), passband edge frequency(ω_p), stopband edge frequency (ω_s), passband ripple (δ_p), stopband ripple (δ_s) and algorithm parameters objective function, Total number of flowers (n), Size of each flower (candidate solution), Total no. of iteration (MaxIteration), Upper bound (U_B)and Lower bound(L_B) are used as the input to the algorithm. In the flowchart t refers to the current generation. The algorithm starts with the step of generating the initial population where each candidate solution represents a set of filter coefficients. As we are concerned about the symmetric filters, size of each candidate solution of the population is $K = \left\lceil \frac{N+1}{2} \right\rceil$. Initial population is generated randomly but within a specific range demarcated by lower and upper bounds, UL and UB respectively. It can be defined by the following equation:

$$x_{j,i} = rand_j(0,1).\left(UB_j - UL_j\right) + UL_j \tag{5.34}$$

$rand_j(0,1)$ returns a uniformly distributed random number within range $(0,1)$ i.e. $0 < rand_j(0,1) < 1$. Generated number is multiplied with $\left(UB_j - UL_j\right)$ and then added to UL_j to obtain a number between UL_j & UB_j. The subscript j signifies that a new random value is generated for each element of a single candidate solution.

Quality of each solution in the initial population is evaluated using a mean square error based objective function defined in (5.8). For determining error in passband and stopband (5.9) and (5.10) have been used respectively.

For each solution scaling factor (sf) is computed. Scaling factor is the ratio of 2^B to the maximum valued coefficient. Quantization of the coefficients is performed by multiplying them by 2^B and sf. Adder cost is estimated to implement filter using each solution in the population after performing Common Sub Expression (CSE) elimination (Reddy & Sahoo, 2015). Implementation of filters requires two types of adders: (a) Structural adder and (b) Multiplier adder (Reddy & Sahoo, 2015). Number of structural adder is estimated by computing the total number of addition required to obtain filter coefficients but after CSE elimination. Required number of multiplier adders is estimated by calculating the total number of SPT terms or the nonzero bits required to represent filter coefficients but avoiding the repeating terms or bit positions.

Solution with the least value of φ is then stored in BEST. In the next step a switching parameter $p \in [0,1]$ is chosen. For values of t less than MaxIteration new solutions are generated by local distribution or global distribution and solutions of the last population are replaced by the new better solutions. For each new solution required number of adders is estimated to implement a filter. Solution with minimum φ value is stored in CURRENT_BEST. BEST is updated after each of the iterations if local best solution obtains better filter than the global best solution in other words it can be said that CURRENT_BEST is less than BEST. Algorithm terminates after maximum number of iterations occurred. Finally an optimized set of filter coefficients is received from the algorithm. Optimized filter coefficients are capable of implementing a symmetric FIR filter with minimum adder cost.

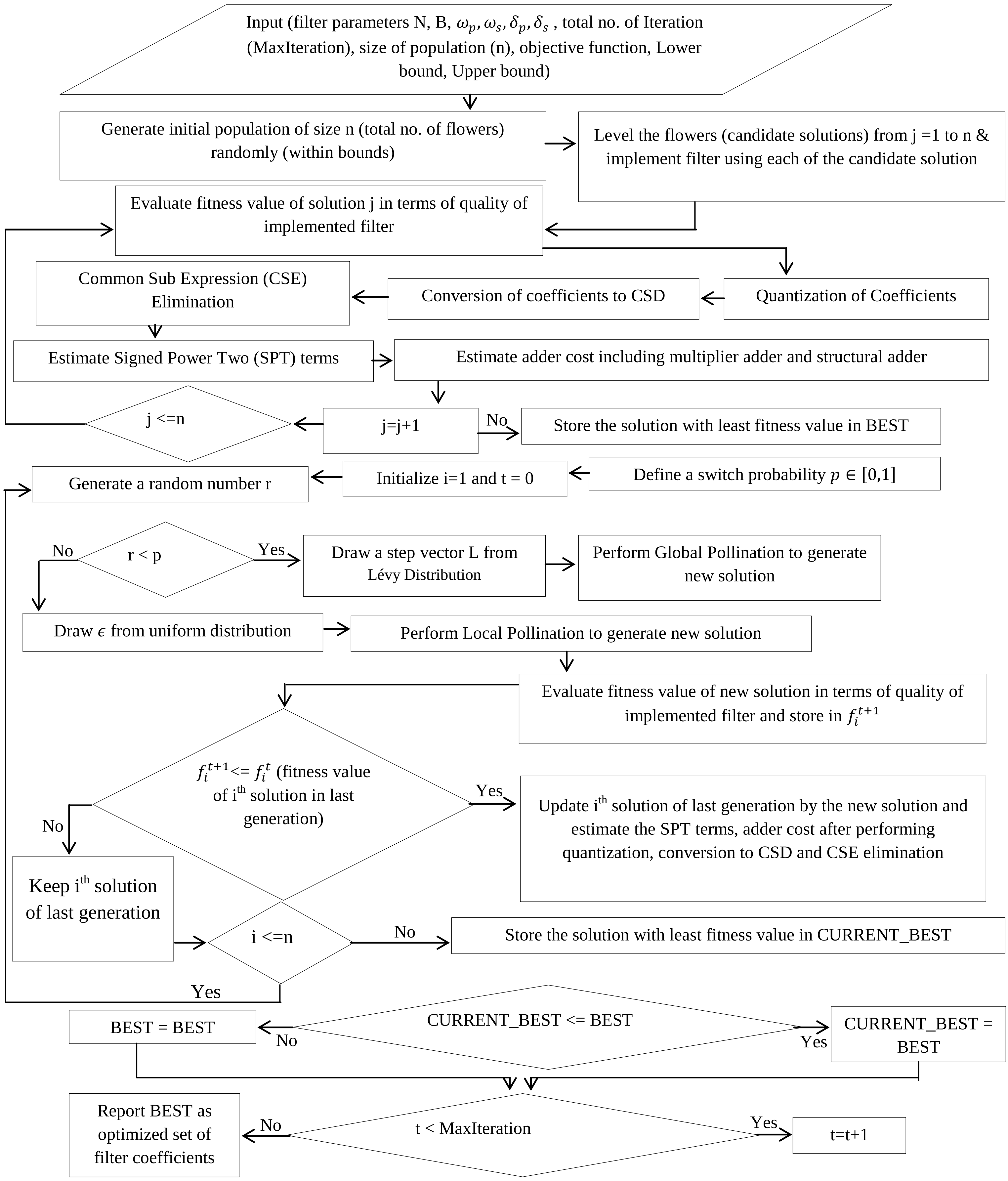

Figure 5.13: Filter Design using Fast Converging Flower Pollination Algorithm

5.6.3. Simulation Results

Present section covers up analysis of the simulation results of Fast Converging Flower Pollination Algorithm (FFPA) used to design symmetric FIR filters in transposed direct form.

In our research work 0.25π and 0.39π are used as the normalized pass band and stop band edge frequencies respectively. Stop band attenuation is taken as 30. Pass band and stop band both are having ripple of 0.4. To obtain a benchmark filter again Parks McClellan Algorithm (PMA) (Parks & McClellan, 1972) is used. As we are concerned about the symmetric filters and the filter order is specified at 20, size of each candidate solution of the initial population of FFPA must be 11. After receiving an optimized solution as the output of the algorithm, using symmetricity property of the coefficients a complete set of filter coefficients can be obtained.

Figure 5.14(a) contains magnitude response of different filters implemented using the coefficients optimized by Flower Pollination Algorithm but with different population size and specific number of total iterations of 500. Response of these filters do not match with the ideal filter characteristics which clearly indicates that number of iteration chosen in this case is not suitable for the implementation of filter. Similar process is replicated for different values of total iterations ranges from 1000 to 4000 with a gap of 500 and plots obtained in each case for different population sizes are shown in Figure 5.14(b) to 5.14(h). Studying the responses of the filters shown in Figure 5.14 it is observed that for iteration 3000 and population size 20, filter is obtaining least stopband attenuation. Hence, for further study iterations of 3000 and population size 20 are chosen.

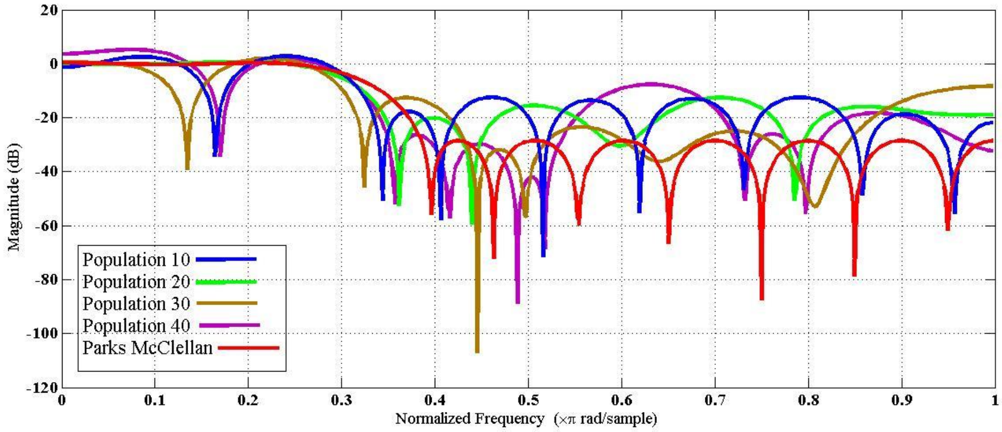

Figure 5.14 (a)

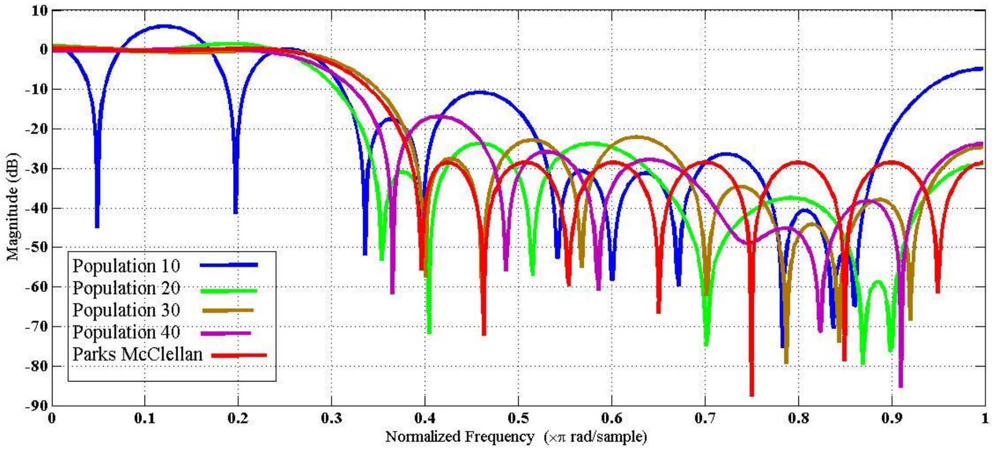

Figure 5.14 (b)

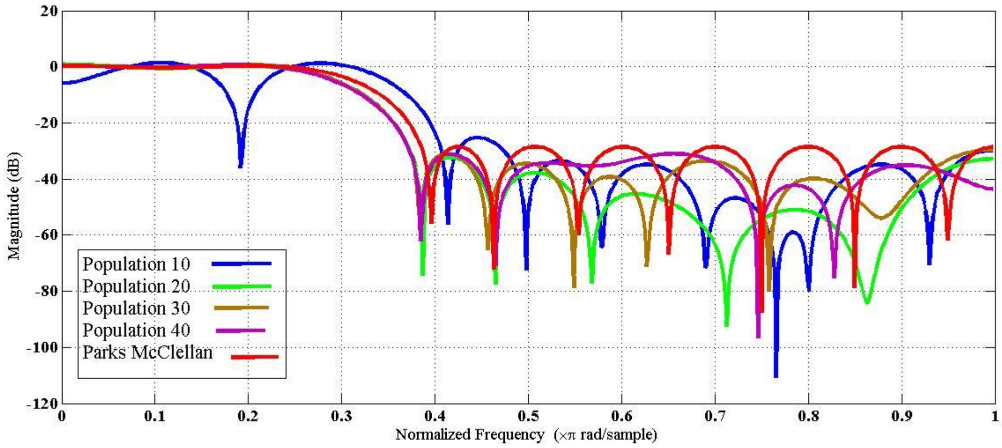

Figure 5.14 (c)

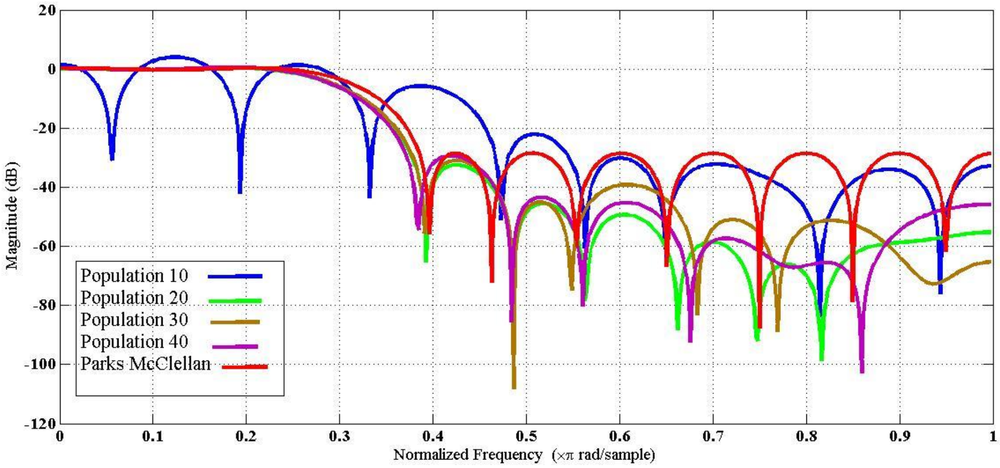

Figure 5.14 (d)

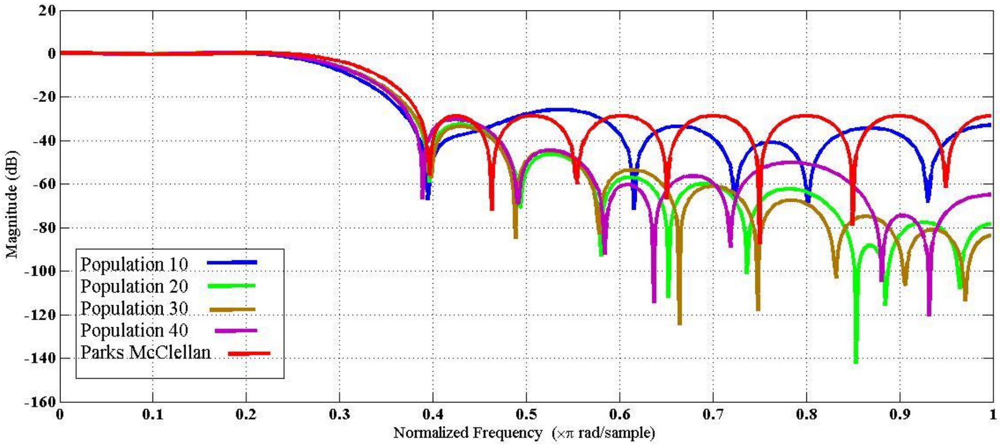

Figure 5.14 (e)

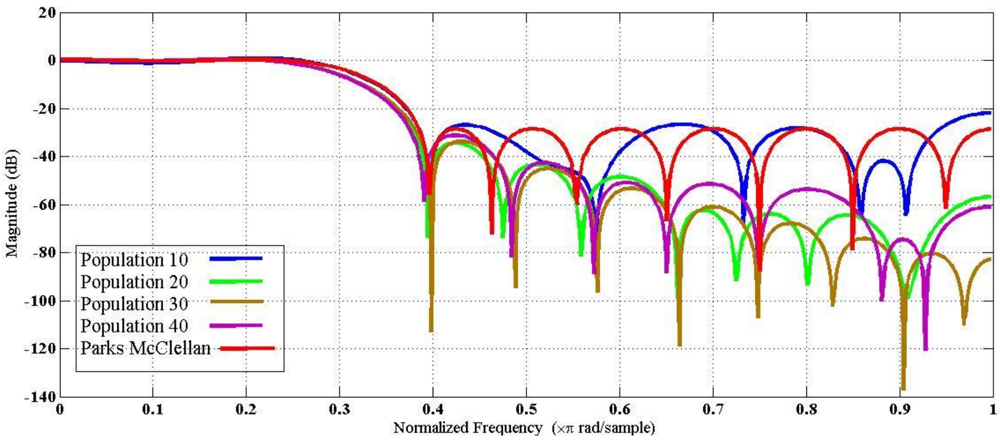

Figure 5.14 (f)

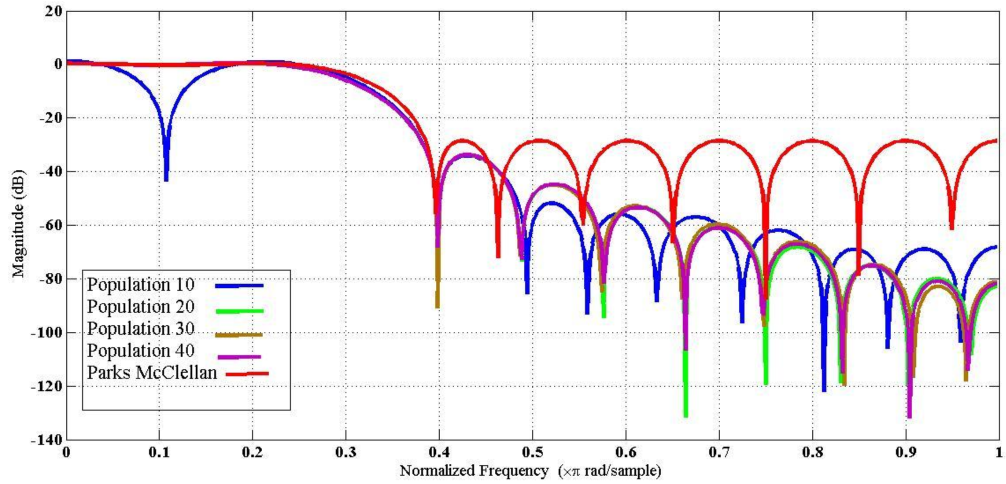

Figure 5.14 (g)

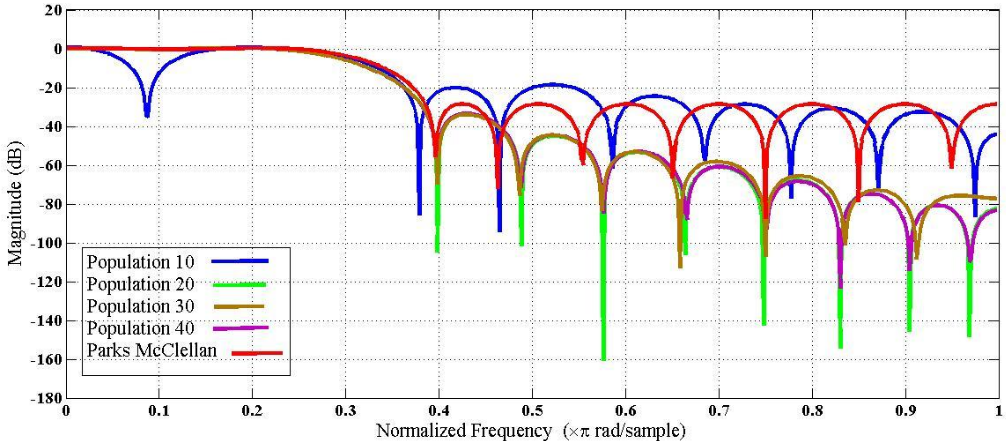

Figure 5.14 (h)

Figure 5.14: magnitude response of different filters implemented using the coefficients optimized by Flower Pollination Algorithm but with different population size and total number of iterations (a) 500, (b) 1000, (c) 1500, (d) 2000, (e) 2500, (f) 3000, (g) 3500, (h) 4000

Hardware cost of a filter is very much dependent on the total numbers of adders required to design a filter. Adder cost of a filter can either be reduced by decreasing order of filter or by reducing number of nonzero bits present in the filter coefficients. Magnitude response of four different filters FFPA1, FFPA2, FFPA3 and FFPA4 are shown in Figure 5.15 respectively. FFPA1, FFPA2 and FFPA3 filters are implemented using the optimized coefficients obtained by Fast Converging Flower Pollination Algorithm with iterations 3000, population 20 but with different word length of coefficients. From Table 5.19 it can be observed that FFPA1 uses coefficients of word length 10 and obtains stop band attenuation of 37.61 but requires 16 adders and 23 SPT terms to be designed. Hence the filter achieved a gain of 37.8378% in SPT terms and 23.8095% in adder cost compared to the benchmark filter implemented using PMA. Whereas by reducing the word length of the coefficients to 9 FFPA2 obtains stop band attenuation of 34.23 with only 12 adders and 18 SPT terms. FFPA2 achieved a gain of 51.3513% in ST terms and 42.8571% in adder cost. Further reduction in word length leads to decrement in SPT terms to 14 but there is no reduction in adder cost is observed. Stop band attenuation obtained by this filter FFPA3 is 30.10, gain in SPT terms is 62.1621 and gain in adder cost is 42.8571%. FFPA4 is another filter shown in this study where to reduce adder cost order of the filter is decreased from 20 to 18. FPA4 gives stop band attenuation of 30.15 and requires 16 adders to be implemented using the coefficients of word length 10. Hence the filter achieved a gain of 24.3243% in SPT terms and 28.5714% in adder cost.

A comparative study of attenuation in stopband and adder costs of the filters implemented using optimized coefficients obtained by Fast Converging Flower Pollination Algorithm as well as the filters designed using the optimized coefficients obtained by different traditional optimization algorithms like GA, PSO, Differential Evolution (DE) and CSA is performed in Table 5.19. Efficiency of Fast Converging Flower Pollination Algorithm is clearly proved by the comparative study as FFPA1 obtains least stop band attenuation of 37.61 among all the filters designed by the coefficients having similar word length. In terms of adder cost also FFPA1 achieves gain of 23.8095% compared to the benchmark filter which is pity high than the gain achieved by the filters designed using the coefficients obtained by other conventional algorithms. FFPA2 and FFPA3 with reduced word length of 9 and 8 respectively also satisfy the required filter characteristics. FFPA4 with reduced filter order of 18 is also capable of achieving filter response as required. Magnitude

Response and Frequency Response of these filters are shown in Figure 5.17(a) and 4(b) respectively.

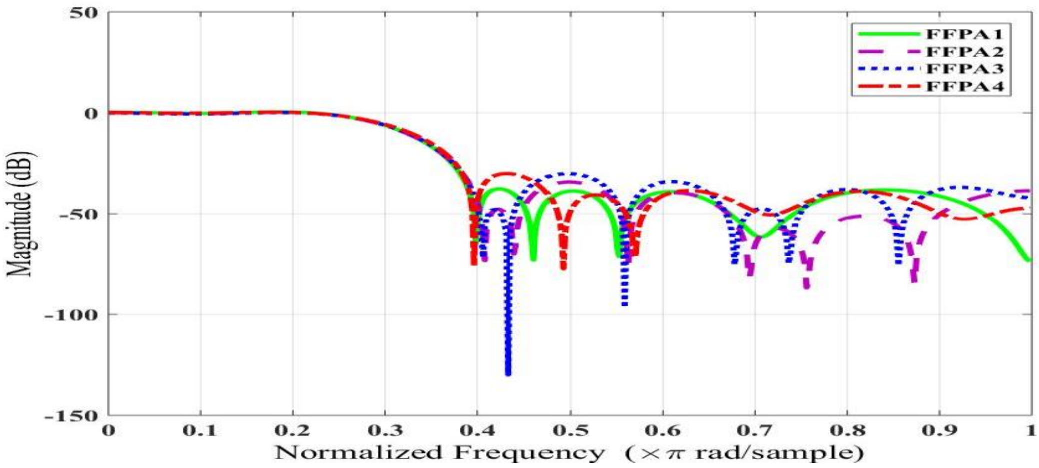

Figure 5.15: Magnitude Response of filters implemented by the optimized coefficients obtained using FFPA1, FFPA2, FFPA3 and FFPA4

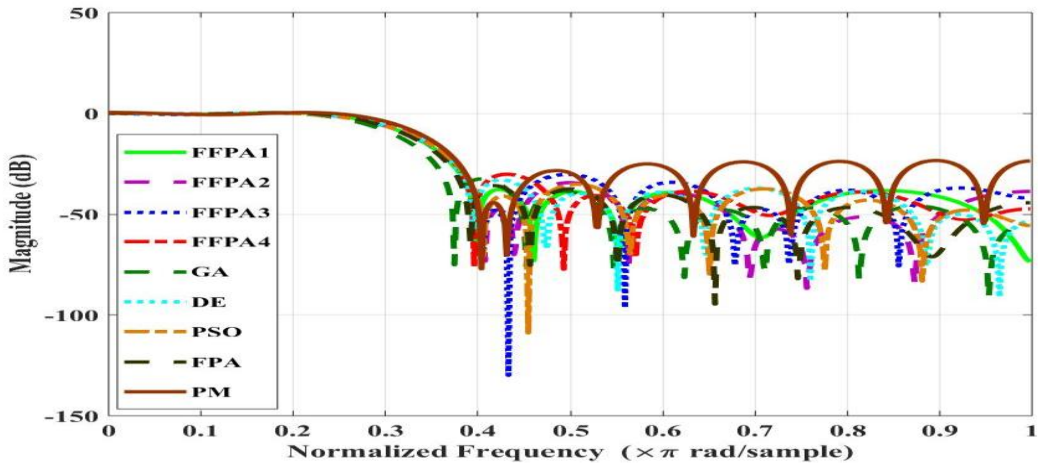

Figure 5.16: Comparison of Magnitude Response of filters implemented by the optimized coefficients obtained using FFPA1, FFPA2, FFPA3, FFPA4, GA, PSO, ACO, CSA and Parks McClellan Algorithm

Table 5.19: Comparative study of different properties of the designed filters

Filters	N	WL	Asb (dB)	SPT	SPT gain (%)	MA	SA	TA	TA gain (%)
PM	20	10	28.58	37		8	13	21	
GA	20	10	32.34	30	21.875	9	12	21	0
DE	20	10	32.86	32	15.625	7	10	17	19.0476
PSO	20	10	35.03	28	28.125	7	10	17	19.0476
CSA	20	10	35.70	28	28.125	7	10	17	19.0476
FPA1	20	10	37.61	23	37.8378	6	10	16	23.8095
FPA2	20	9	34.23	18	51.3513	6	6	12	42.8571
FPA3	20	8	30.10	14	62.1621	6	6	12	42.8571
FPA4	18	10	30.15	28	24.3243	7	10	15	28.5714

Table 5.20: Execution time of GA, PSO, ACO, CSA, FPA, and FFPA to obtain optimized coefficients for 20th order lowpass filters

Algorithm	Execution Time(Seconds)
GA	50.0044
PSO	49.2032
ACO	47.2204
CSA	42.9535
FPA	45.1742
FFPA	41.8801

Present case study of our research work includes filtration of a noisy Phonocardiogram (PCG) Signal. For proper diagnosis of diseases, any biomedical signal is needed to be noise free. In Phonocardiography high-fidelity recording of the sounds and murmurs made by the heart during a cardiac cycle caused by flow of blood through the heart are plotted using machine named phonocardiograph (Ganguly & Sharma, 2017). Usually heart produces the sound Lub & Dub, where Lub is the first sound S1 and S2 is the second sound Dub. The time between S1 and S2 is systole (Lub-------Dub), caused by the flow of blood from the heart to the lungs and body, flow of blood across the Pulmonic and Aortic valves (ausmed, 2018). This sound primarily occurs due to closing of the bicuspid and tricupsid valves. They close because of the contraction of the ventricle. The time between S2 and S1 is diastole (Dub--------Lub), caused by closing by the flow of blood from the atria to the ventricles, flow of blood across the bicuspid and tricuspid valves.

Anatomy of heart sounds can be obtained by the following (ausmed, 2018):

LUB-- DUB-------------LUB--DUB

S1 S2 S3 S4 S1 S2

S1 occurs at the onset of the ventricular contraction. It contains a series of low-frequency vibrations, and is usually the longest and loudest heart sound.

S2 occurs at the end of the ventricular contraction. Its frequency is higher than S1, and its duration is shorter.

S3 (ventricular gallop) is a low frequency sound, may be heard at the beginning of the diastole during the rapid filling of the ventricles.

S4 (atrial gallop) may be heard in late diastole during atrial contraction.

S3 & S4 are of very low intensity, and are only audible externally when amplified.

Other - opening snap, ejection sound may be heard at the time of valve diseases.

Murmurs are high frequency noise like sounds which may be heard between the two major heart sounds systole and diastole (ausmed, 2018).

For analysis a noise free PCG signal of a 40 years old human being negatively diagnosed with any heart disease is collected from Jeevan Rekha Diagnostic Pvt. Ltd., India is collected and then mixed with Gaussian white noise. Noisy PCG signal is then filtered using the implemented filters FFPA1, FFPA2, FFPA3 and FFPA4.

In Table 5.21 comparative study of the Signal to Noise Ratio (SNR) and correlation value of different filtered signals, filtered by the filters designed using the optimized coefficients obtained by FFPA1, FFPA2, FFPA3, FFPA4 and traditional algorithm for PMA are shown.

Table 5.21: Comparative study of SNR and Correlation value of filtered heart sound signal filtered using different designed filters

Filters	SNR	Correlation value
PM	11.0577	0.9663
FFPA1	11.3342	0.9763
FFPA2	11.4646	0.9764
FFPA3	11.6489	0.9767
FFPA4	12.0718	0.9894

Table 5.22 shows comparative study of average error (ε_{avg}) of filtered phonocardiogram signal filtered using different designed filters.

Table 5.22: Comparative study of average error (ε_{avg}) of filtered phonocardiogram signal filtered using different designed filters

Filters	ε_{avg}
PM	0.0324
FFPA1	0.0313
FFPA2	0.0308
FFPA3	0.0301
FFPA4	0.0297

In the Figure 5.17(a) – 5.17(g) frequency spectrums of the original PCG signal, noisy PCG signal and the filtered PCG signals are shown. Analysis of Figure 5.17(a) proves the presence of S1 and S2 in lower frequency range. Figure 5.17(b) shows the presence of noise in higher frequency range also. Figure 5.17(c) – 5.17(g) prove the efficiencies of different filters in filtering noise from higher frequency range. Figure 5.18(a) – 5.18(e) show the near view of the frequency spectrums shown in Figure 5.17(c) – 5.17(g) correspondingly. From Figure 5.18(a) – 5.18(e) it can be observed that filter implemented by PMA have reduced lesser amount of noise in the frequency ranges 7000-7500 Hz and 8000-9000 Hz compared to the filters designed using the coefficients optimized by FPA. In Figure 5.19(a) and 5.19(b) column chart representation of SNR and correlation values of the filtered signals are shown respectively. In Table 4 comparative study of average error calculated by the following Eqn. 5.35 (Cherif et al., 2014) for different filters used to filter noisy PCG signal is shown. Figure 5.20 shows the same comparison in column chart representation.

$$\varepsilon_{avg} = \frac{\sum_{i=1}^{N} |P_{oi} - P_{ri}|}{N} \quad (5.35)$$

$p_o$is the original PCG signal, p_{oi} is i^{th} sample of p_o. $p_r$is the synthesis PCG signal and p_{or} is i^{th} sample of p_r.

It is proved that filters implemented using the coefficients optimized by FPA outpace the filter designed using PMA and also the IIR filters used in (Cherif et al., 2014) for filtering noisy PCG signal in terms of average error, whereas comparison with the higher order FIR filters has been ignored as because key motivation of the present research work is to design an hardware efficient FIR filter.

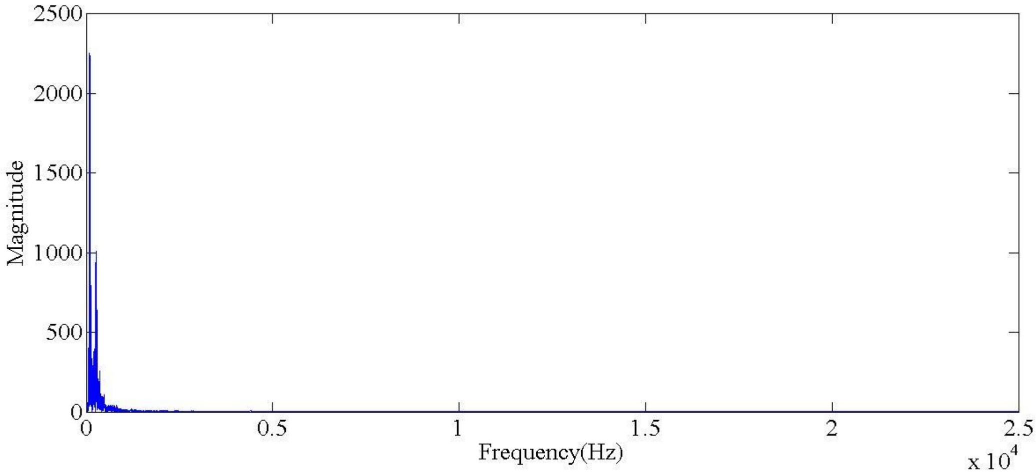

Figure 5.17(a)

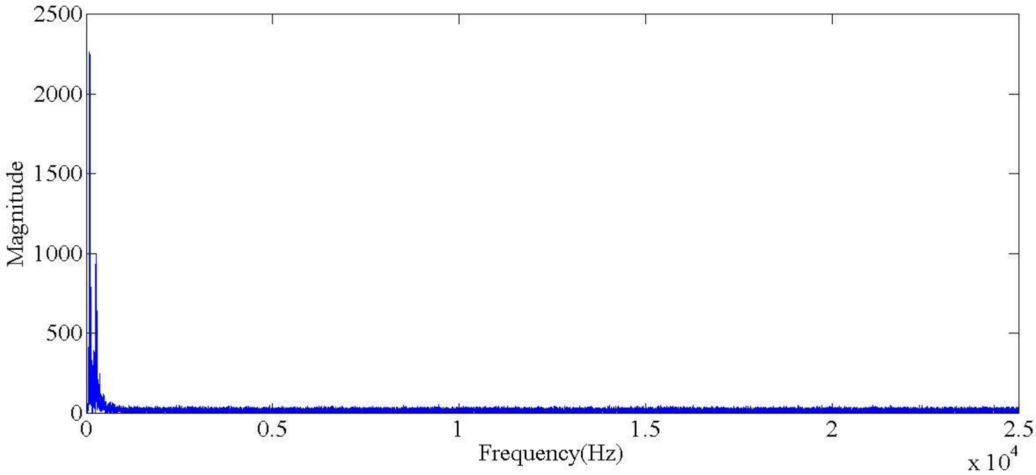

Figure 5.17(b)

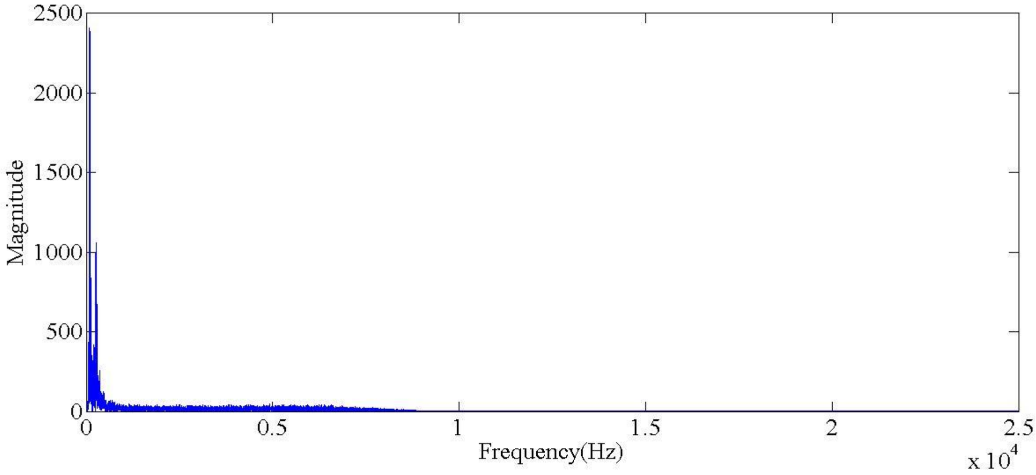

Figure 5.17(c)

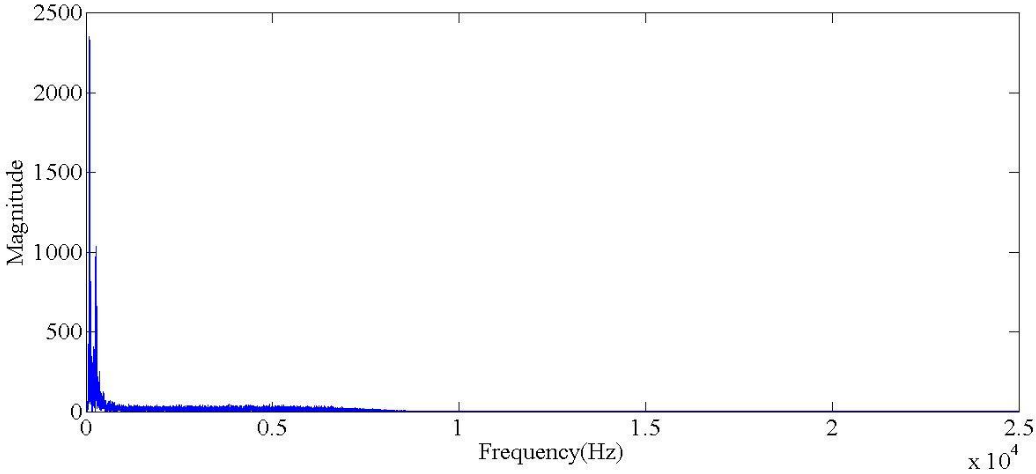

Figure 5.17(d)

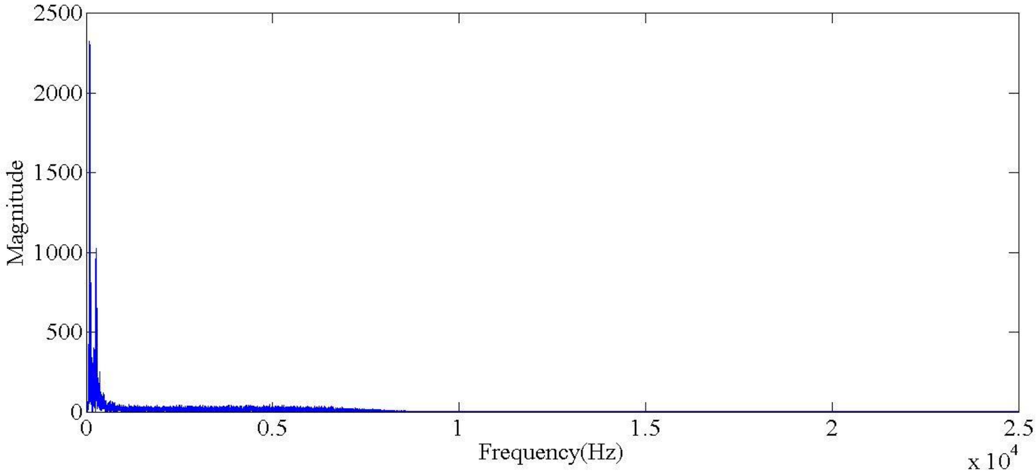

Figure 5.17(e)

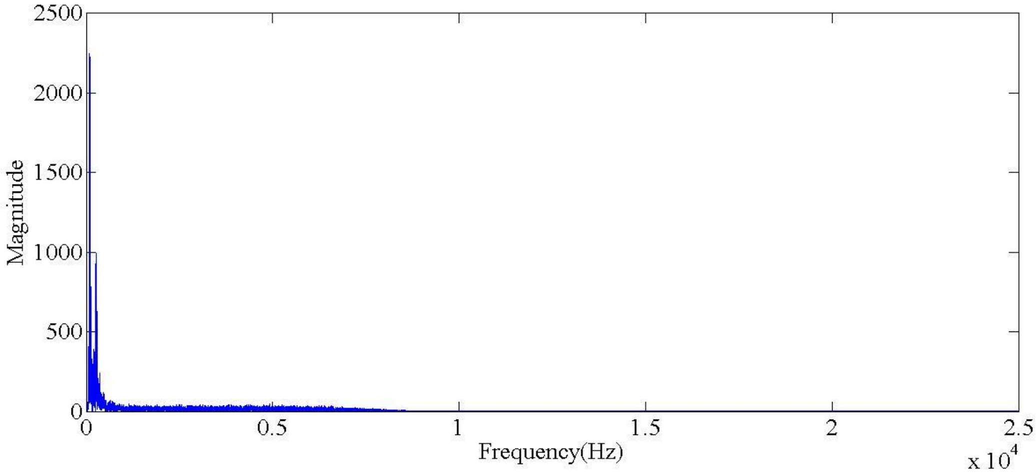

Figure 5.17(f)

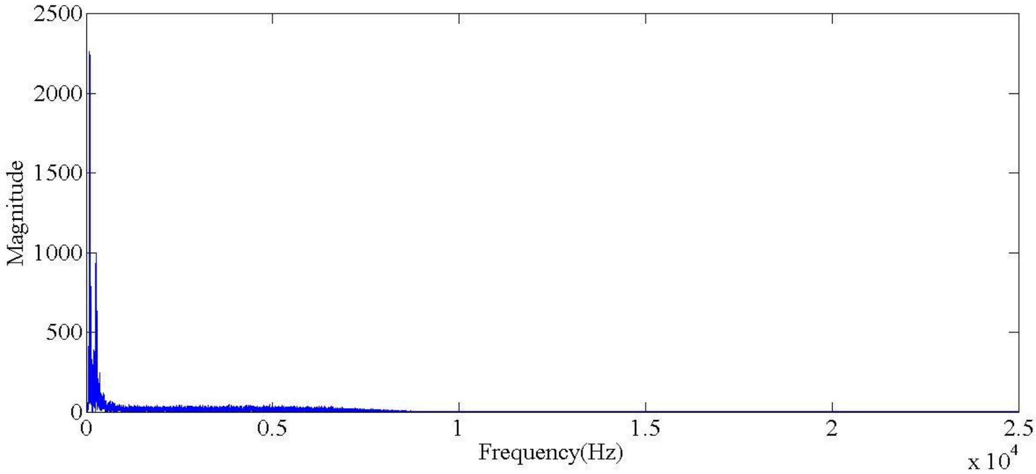

Figure 5.17(g)

Figure 5.17: Frequency spectrum of (a) Original PCG Signal (b) Noisy PCG Signal (c) PCG Signal Filtered by PMA (d) PCG Signal Filtered by FPA1, (e) PCG Signal Filtered by FPA2 (f) PCG Signal Filtered by FPA3 (g) PCG Signal Filtered by FPA4

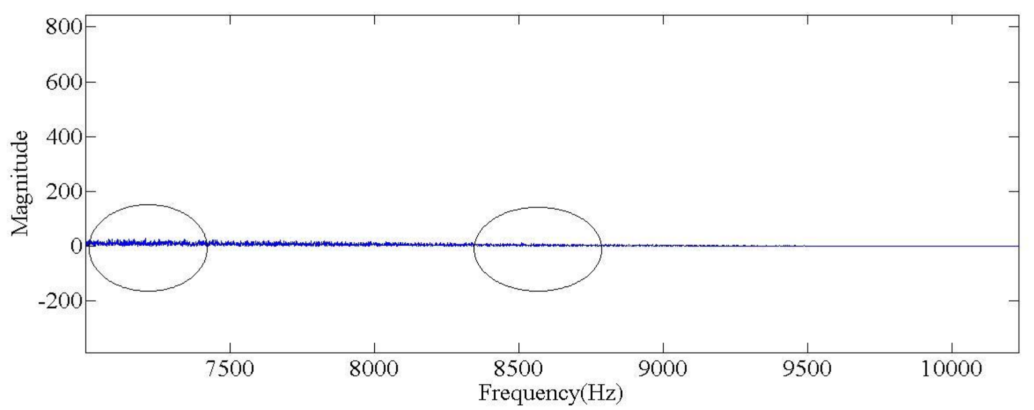

Figure 5.18(a)

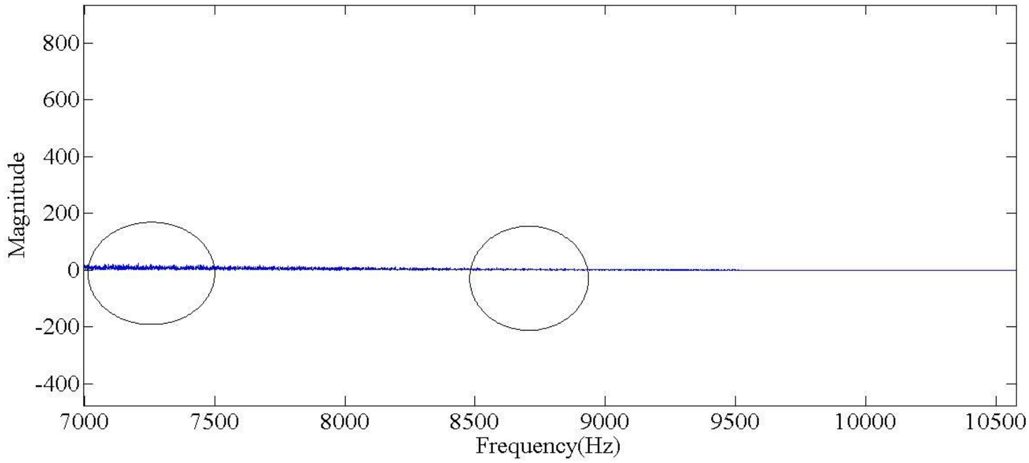

Figure 5.18(b)

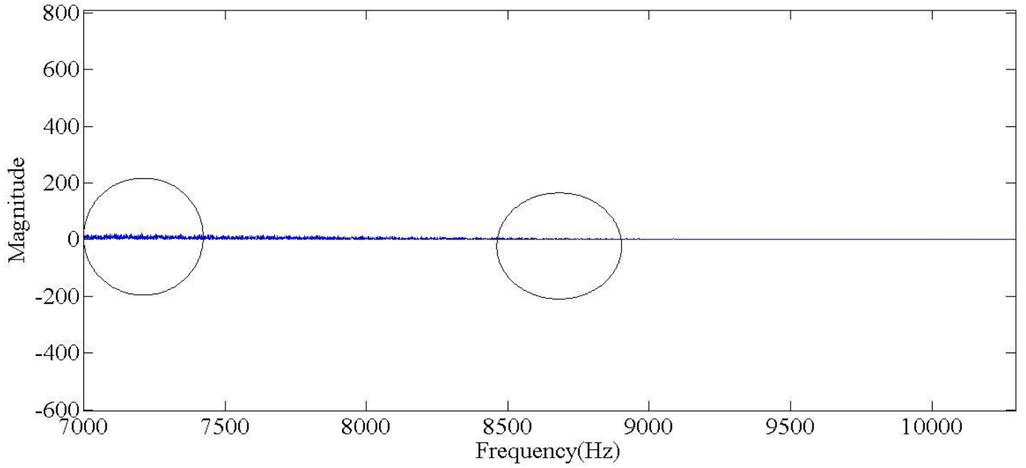

Figure 5.18(c)

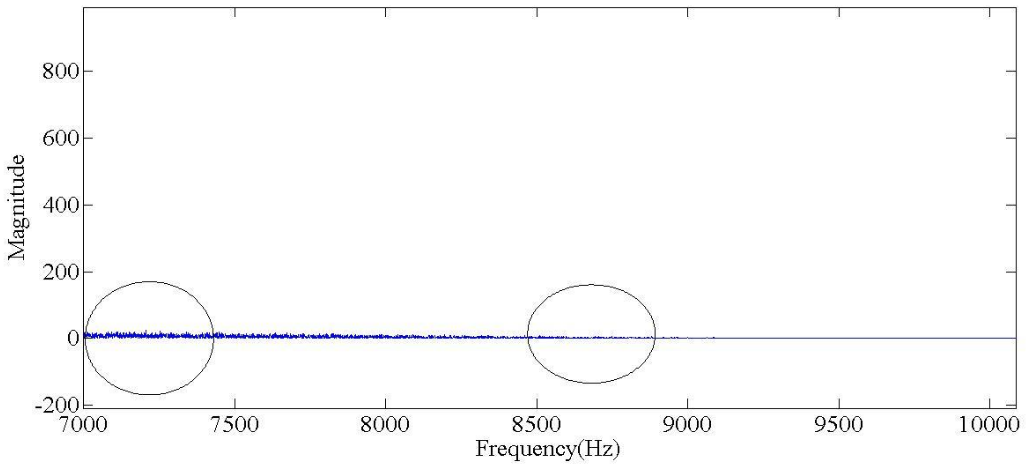

Figure 5.18(d)

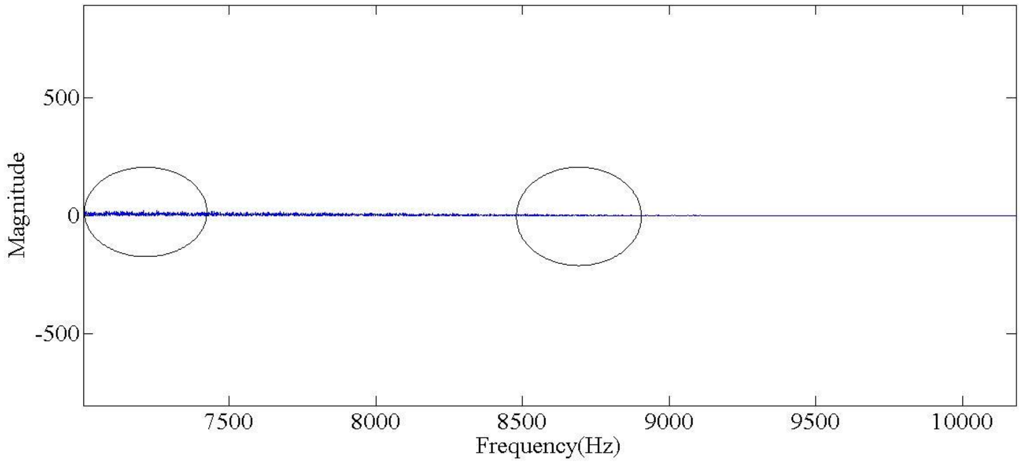

Figure 5.18(e)

Figure 5.18: Near view of frequency spectrum of (a) PCG Signal filtered by PMA, (b) PCG Signal filtered by FPA1, (c) PCG Signal filtered by FPA2, (d) PCG Signal filtered by FPA3, (e) PCG Signal filtered by FPA4

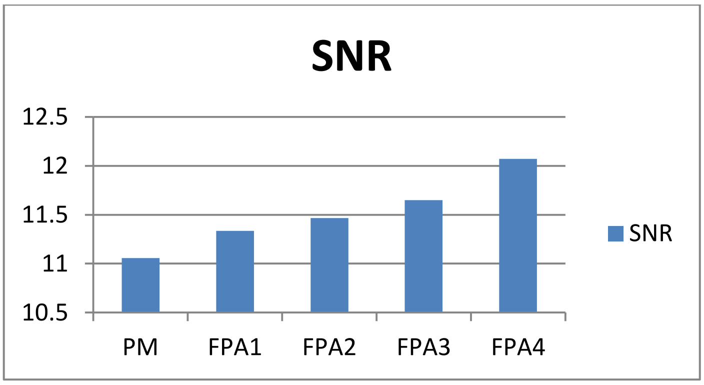

Figure 5.19(a)

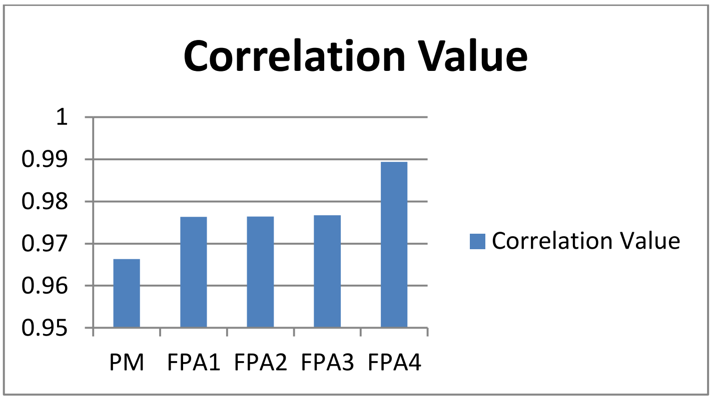

Figure 5.19(b)

Figure 5.19: Column chart representation of (a) Comparison of SNR, (b) Correlation value of filtered PCG signals

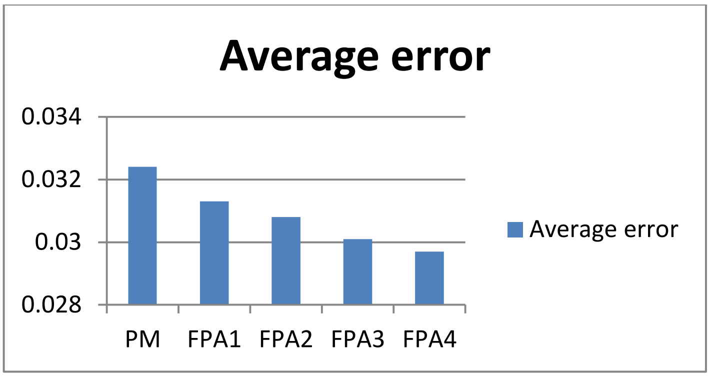

Figure 20: Column chart representation of average error of the filtered PCG signals

5.7. Conclusions

This chapter first proposes an adaptive algorithm namely Global Best Steered Quantum Inspired Cuckoo Search Algorithm (GQICSA) for designing efficient FIR filter with lower hardware cost. GQICSA is used for obtaining optimized set of filter coefficients. Adder cost of a filter is estimated after quantizing the filter coefficients followed by CSE elimination. Among the two kinds of adders, number of structural adders is assessed by computing the total number of additions required to obtain filter coefficients and the required number of multiplier adders is assessed by calculating the total number of SPT terms or the nonzero bits required to represent the filter coefficients. It can be observed from the simulation results and discussions that reduction in word length of coefficients allows the optimized set of filter coefficients to achieve the ideal frequency response and also to outperform the benchmark filter. It has also been shown in the analysis that QICSA performs better than other conventional algorithms for optimizing filter coefficients to design low (hardware) cost filters without compromising the filter responses and efficacy, whereas proposed GQICSA performs even better than QICSA. GQICSA also shows better performance in implementation of higher order filters. Efficiency of GQICSA over QICSA and conventional CSA has been proved with 16 well known benchmark functions. Statistical significance t-test also proves GQICSA as significantly better than other conventional algorithmic approaches. GQICSA can be used to obtain hardware efficient highpass, bandpass as well as bandstop filters. As future work, we would like to use more efficient optimization algorithms for implementing hardware efficient filters. Another direction would be to use the designed filters to filter some real life signals like biomedical signals, astrophysical signals etc.

In the trailing part of this chapter proposes an approach to design an efficient FIR filter with lower hardware cost using an adaptive nature inspired optimization algorithm. Fast Converging Flower Pollination Algorithm is used for obtaining optimized set of filter coefficients. Adder cost of a filter is estimated after quantizing the filter coefficients followed by CSE elimination. It is also proved in analysis that the Fast Converging Flower Pollination Algorithm performs better in fewer iterations compared to other conventional algorithms for optimizing filter coefficients that can be used to design a lower hardware costing filter without compromising the filter responses and efficacy. Performance analysis of the designed hardware efficient filters has been shown by filtering a noisy phonocardiogram signal. In future studies by modifying cut-off frequencies and filter order more efficient filter for de-noising phonocardiogram signal can be implemented with the use of Fast Converging Flower Pollination Algorithm.

Chapter 6

Conclusions and Future Research

The scientific contributions of the research reported in this thesis along with a brief road map of the possible future avenues of this work are reported in this Chapter. One of the foremost goals of the present research is to improve the quality of the digital filters used for removing noise from signals even in lower hardware cost. In recent times, the research activities in the field of digital signal processing are gaining ground under the umbrella of soft computing. Limitations of the conventional approaches for filter design motivated the researchers to use soft computing techniques to resolve the purpose. Several soft computing approaches have been proved to be effective for implementation of Finite Impulse Response (FIR) filters.

In chapter 3 of the thesis, we presented an approach of using Genetic Algorithm (GA) for obtaining least noisy signal from a set of filtered signals. Kaiser Window function with passband ripple varying from 0.01 to 0.40 and stopband ripple varying from 0.09 to 0.49 have been used for filtering noisy heart sound signal. For evaluation of fitness values of the solutions, an adaptive parameter of signals is used. Genetic selection is performed using the Roulette Wheel Selection approach. Genetic operators, crossover and mutation, are used for updating the candidate solutions with the probabilities of 100% and 25% respectively. The proposed Algorithm has been applied over the corrupted heart sound signal for a range of sampling frequency (5000-12000 Hz). Filtered heart sound signals obtained by the proposed algorithm for different sampling frequencies are then compared with the original signal for the evaluation purpose.

The proposed approach is capable of identifying subtle noise present in the signal and excluding it from the signal. Hence, it is very much useful for de-noising biomedical signals, where very little amount of noise may cause erroneous diagnosis. In future research instead of using GA, more advanced optimization techniques like Cuckoo Search Algorithm, Flower Pollination Algorithm (FPA) and many more can be used for the same purpose. Fitness function used to determine fitness values of the candidate solutions can be modified to improve the quality of the optimized filtered signal.

Another new strategy based on the Ant Weight Lifting (AWL) algorithm has been proposed in chapter 3 for filter parameter optimization. AWL is used to optimize the parameters required to design digital FIR filters using Kaiser Window function. A new innovative objective function has been introduced for optimization that performs based on the signal de-noising capability of the implemented filters. A case study has been carried out on de-noising heart sound signal using the filter implemented by the Kaiser Window with optimized parameters. In future research instead of using AWL, more advanced optimization techniques such as CSA inspired by quantum principles can be used for the same purpose. Even AWL can also be modified to improve its performance.

A comparative study of three conventional optimization algorithms, GA, Particle Swarm Optimization (PSO) and BAT Algorithm, for optimizing FIR filter coefficients has been presented in chapter 4. Responses of the implemented filters are compared with the traditional approach of filter design using Parks McClellan Algorithm (PMA) as reference. In optimization algorithms, mean square error based function is used as the objective function. It was observed that the BAT algorithm statistically outperforms GA and PSO in terms of stopband attenuation characteristics and ripple performance of the designed filter. In chapter 4, we have also proposed design of even order low pass FIR filter and odd order bandpass FIR filter using coefficients optimized by an adaptive Global Best steered

Cuckoo Search Algorithm (gbest CSA). For optimization, we also use here a mean square error based objective function. We evaluated the efficacy of the proposed technique by comparing the filter responses with responses of the filters designed using standard Cuckoo Search Algorithm (CSA) and traditional technique of filter design with PMA. Superiority of the proposed algorithm compared to the conventional CSA has been proved using seven standard benchmark functions. Another adaptive algorithm, Fast Converging Cuckoo Search Algorithm (FCSA), is used for optimizing the coefficients of lowpass, highpass and bandpass FIR filters. Efficiency of the proposed algorithm has been demonstrated in all the cases compared to the traditional CSA and PMA.

In chapter 5, a new algorithm, namely *Global Best Steered Quantum Inspired Cuckoo Search Algorithm* (GQICSA), has been proposed for obtaining optimized set of coefficients to implement FIR Filter. Adder cost of a filter is estimated after quantizing the filter coefficients followed by Common Sub-expression Elimination (CSE). We found from the simulation results that reduction in word length of coefficients does not negatively affect the filters to achieve the ideal frequency response. Moreover, filters developed using GQICSA outperform the benchmark filters designed by PMA in terms of stopband attenuation. Analysis of the results revealed that GQICSA not only improves over various conventional algorithms including CSA, it also surpasses Quantum Inspired CSA (QICSA), a modified version of CSA updated using quantum principles, for optimizing filter coefficients to design lower hardware costing filters without compromising the filter responses and efficiency. GQICSA also provides significant improvements compared to CSA and QICSA in terms of stopband attenuation and execution time for higher order filter design. Efficiency of GQICSA over QICSA and conventional CSA has also been exhibited with 16 benchmark functions.

For the same purpose for implementing hardware efficient FIR filters, a new algorithm namely Fast Converging Flower Pollination Algorithm has also been used in chapter 5. It has been shown by the simulation results that reduction neither in word length of coefficients nor in filter order causes the filter implemented using optimized set of coefficients obtained by the proposed algorithm to be incapable of achieving the ideal frequency response. Implemented Filters have been proved to be effective to filter noisy Phonocardiogram signal.

In most of the above mentioned cases for filter coefficient optimization, mean square error function is used as the objective function. In future research studies, instead of using the same objective function, an innovative one can be used aiming to enhance the filter quality. Moreover, the proposed algorithms can be modified to use as multi-objective optimization algorithms for high quality filter design with other conflicting objectives. Amazing physiognomies of nature can be used to design innovative optimization algorithms which can be used to implement much efficient filters. In further research work proposed algorithms such as gbest CSA & GQICSA both can be used to design filter banks, Infinite Impulse Response (IIR) filters. Moreover, a set of filtered signals can also be obtained using these algorithms by means of single or multiple objectives, and then from the set of filtered signals an optimized signal can be found again using these optimization algorithms.

Bibliography

- Ababneh J. I., & Bataineh M. H. (2008). *Linear phase FIR filter design using particle swarm optimization and genetic algorithms.* Digital Signal Processing, 657-668.

- Aggarwal A., Rawat T. K., Kumar M., & Upadhyay D. K. (2015). *Optimal design of FIR high pass filter based on L1 error approximation using real coded genetic algorithm.* An International Journal Engineering Science and Technology, 18: 594-602.

- Aggarwal D., Reddy K. S. & Sahoo S. K. (2013). *FIR filter design approach for reduced hardware with order optimization and coefficient quantization*, in Proc: Int. Conf. on Intelligent Sys. and Signal Processing, 1-2 March, 2013, Gujrat, India.

- Ahmed S. U. (2008). *Design of Digital Filters Using Genetic Algorithms.* University of Victoria, Department of Electrical and Computer Engineering. Victoria: Sabbir U. Ahmad, 2008, University of Victoria.

- Aickelin U. & Clark A. (2011). *Heuristic Optimization*, Journal of Operational Research Society 62: 251-252.

- Aktan M., Yurdakul A. & Dundar G. (2008). *An algorithm for the design of low-power hardware-efficient FIR filters.* Transactions on Circuits and Systems I: Regular Papers, 55: 1536-1545.

- Antoniou A. (1993). *Digital Filters: Analysis, Design,, and Applications* (2 ed.). New York: McGraw-Hill.

- antweb, 2018. Retrieved from <http://www.antweb.org/antblog/2012/10/how-can-ants-carry-so-much-weight-nathalie-france.html> (last accessed on 22nd April, 2018).

- An-xin Z., Ping L. & Jian-jun L. (2011). *The design of FIR filter based on APA Genetic Algorithms*, International Conference on Mechatronic Science, Electric Engineering and Computer, 19-22 August, 2011, Jilin, China.

- ausmed, 2018. Retreived from <https://www.ausmed.com/articles/heart-murmur-sounds/> (last accessed on 22nd April 2018).

- a-z-animals, 2018. Retrieved from <http://a-z-animals.com/animals/ant/> (last accessed on 22nd April, 2018).

- Baghel M., Agrawal S. & Silakari S. (2012). *Survey of Metaheuristic Algorithms for Combinatorial Optimization*, International Journal of Computer Applications. 58: 21–31.

- Bartheleme P., Liebovitch L. S. & Glendon R.(2008). *A L´evy flight for light*, Nature, 453: 495-498.
- Beckman D., Chari A. N., Devabhaktuni S. & Preskill J. (1996). *Efficient Networks for Quantum Factoring*, Physical Review A., 54(2): 1034-1063.
- Benhala B. (2014). *Ant Colony Optimization for Optimal Low-Pass Butterworth Filter Design*. WSEAS TRANSACTIONS on CIRCUITS and SYSTEMS, 313-318.
- Benioff P. (1980). *The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines.* Journal of Statistical Physics. 22 (5): 563–591.
- Bianchi L., Marco D., Luca M. G, Walter J. G. (2009). *A Survey on metaheuristics for stochastic combinatorial optimization.* Natural Computing: An International Journal, 8(2): 239-287.
- Blackman, R. B. and Tukey, J. W. (1959). *Particular Pairs of Windows*, In The Measurement of Power Spectra, From the Point of View of Communications Engineering. New York: Dover, pp. 98-99, 1959.
- Bone S., Castro M.. *A Brief History of Quantum Computing*, retrieved from https://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol4/spb3/#7 (last accessed on 22nd April 22 2018).
- Brassard G., Høyer P., Mosca M., & Tapp A. (2002). *Quantum amplitude amplification and estimation. in Quantum Computation and Information*. In S. J. Lomonaco, & H. E. Brandt (Eds.), AMS Contemporary Mathematics Series. 305: 53-74.
- Brown C., Liebovitch L. S. & Glendon R.(2007). *L´evy flights in Dobe Ju/'hoansi foraging patterns*. Human Ecol, 35: 129-138.
- Brownlee J. (2011). *Clever Algorithms: Nature-Inspired Programming Recipes.* Melbourne.
- Cen L. (2007). *A hybrid genetic algorithm for the design of FIR filters with SPoTcoefficients.* Article Signal. Processing, 528-540.
- Chakraborty N., Mukherjee A. & Dhabal S. (2016). *Design of IIR filter Using Gbest Guided Cuckoo Search Algorithm.* International Conference on Computer, Electrical & Communication Engineering. Kolkata: IEEE.
- Chakravarti P. K., & Mehra R. (2014, August). *Gibb's Phenomenon Analysis on FIR Filter using Window Techniques.* International Journal of Advanced Engineering Research and Technology, 86-90.

- Chandra A. & Chattopadhyay S. (2013). *Canonic signed digit based design of multiplier-less fir filter using self-organizing random immigrants genetic algorithm.* International Journal of Artificial Intelligence & Applications, 21-34.

- Chandra A. & Chattopadhyay S. (2013). *Supremacy of trigonometric mutation strategy for efficient design of multiplier-less low-pass FIR filter using differential evolution algorithm.* 4th Sustainable Energy and Intelligent Systems Conference, (pp. 272-277), Chennai: IEEE.

- Chandra A. & Chattopadhyay S. (2016, March). *Design of hardware efficient FIR filter: A review of the state-of-the-art approaches.* Engineering Science and Technology, an International Journal, 19, 212-226.

- Chandra A. (2016). *Design of hardware efficient multiplier less low pass FIR filter using evolutionary algorithms.* PhD Thesis, Jadavpur University, Department of Electronics and Tele-Communication Engineering.

- Chandra A., & Chattopadhyay S. (2011). *A novel self adaptive differential algorithm for efficient design of multiplier-less low-pass FIR filters.* Second International Conference on Sustainable Energy and Intelligent System (pp. 733-738). IEEE.

- Chandra A., & Chattopadhyay S. (2012, October). *Role of Mutation Strategies of Differential Evolution Algorithm in Designing Hardware Efficient Multiplier-less Low-pass FIR Filter. JOURNAL OF MULTIMEDIA, 7*: 353-363.

- Chen C. L., & Wilson J. (1999). *A trellis search algorithm for the design of FIR filters with signed-power-of-two coefficients.* Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 46: 29-39.

- De Jong K. (1975). *Analysis of the behaviour of a class of genetic adaptive systems, Ph.D. Thesis.* Ann Arbor, MI,USA: University of Michigan.

- Desale S., Rasool A., Andhale S., & Rane P. (2015). *Heuristic and Meta-Heuristic Algorithms and Their Relevance to the Real World: A Survey*. International Journal Of Computer Engineering In Research Trends, 296-304.

- Deshpande D. K. (2002). *Some studies on the design of equiripple finite impulse response digital filters.* Karnatak University, Department of Physics.

- Deutsch, D. (1985). *Quantum Theory, the Church-Turing Principle and the Universal Quantum Computer*, Proceedings of the Royal Society of London A. **400** (1818): 97–117.

- Dey S., Bhattacharya S. & Maulik U. (2017). *Quantum-inspired multi-objective simulated annealing for bilevel image thresholding.* In S. Bhattachrya, U. Maulik, & P. Dutta, *Quantum Inspired Computational Intelligence Research and Applications* (pp. 207-232). Cambridhe: Elsevier Inc. Todd Green.

- Dey S., Bhattcharya S. & Mallick U. (2017). *Quantum-inspired multi-objective simulated annealing for bilevel image thresholding*", in: S. Bhattacharya, U. Maulik, P. Dutta (Eds.), Quantum Inspired Computational Intelligence - Reseach and Applications, Elsevier Inc., Cambridge, US, pp. 207-232.

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

- Dhabal S. & Venkateswaran P. (2017). *An efficient gbest-guided Cuckoo Search algorithm for higher order two channel filter bank design.* Swarm and Evolutionary Computation, 33, 68-84.

- Djelloul H., Layeb A. & Chikhi S. (2015). *Quantum Inspired Cuckoo Search Algorithm for graph colouring problem,* International Journal Bio-Inspired Computation, 7 : 183-194.

- Dorigo M. & Caro G. D. (n.d.). *The Ant Colony Optimization Meta-Heuristic.*

- Dwivedi A. K. & Patel R. N. (2017). *Digital filter design using quantum-inspired multiobjective cat swarm optimization algorithm*, in: S. Bhattacharya, U. Maulik, P. Dutta (Eds.), Quantum Inspired Computational Intelligence - Reseach and Applications, Elsevier Inc., Cambridge, US, pp. 327-360.

- Eiselt H. A. & Sandblom C.-L. (2000). *Heuristic Algorithms.* In H. A. Eiselt, & C.-L. Sandblom, *Integer Programming and Network Model* (pp. 229-258). Heidelberg: Springer, Berlin, Heidelberg.

- Electric Power Systems Analysis and Nature Inspired Optimization Algorithms, 2015a. Retrived from <http://al-roomi.org/benchmarks/unconstrained/n-dimensions/188-schwefel-s-function-no-1-2-double-sum-or-rotated-hyper-ellipsoid-function> (last accessed on 22nd April 2018).

- Electric Power Systems Analysis and Nature Inspired Optimization Algorithms, 2015b. Retrieved from http://al-roomi.org/benchmarks/unconstrained/n-dimensions/194-step-function-no-3 (last accessed on 22nd April 2018).

- Electric Power Systems Analysis and Nature Inspired Optimization Algorithms, 2015c. Retrieved from <http://al-roomi.org/benchmarks/unconstrained/n-dimensions/176-generalized-schwefel-s-problem-2-26> (last accessed on 22nd April 2018).
- Electric Power Systems Analysis and Nature Inspired Optimization Algorithms, 2015d. Retrieved from <http://al-roomi.org/benchmarks/unconstrained/2-dimensions/7-shekel-s-foxholes-function> (accessed on 22nd April 2018).
- Engelbrecht A. (2005). *Fundamentals of computational Swarm intelligence.* Hoboken: Wiley.
- entomologytoday, 2018. Retrieved from <http://entomologytoday.org/2014/02/11/ants-can-lift-up-to-5000-times-their-own-body-weight-new-study-suggests/> (last accessed on 22nd April, 2018).
- Feng Z. G. & Teo K. L. (2008). *A discrete filled function method for the design of FIR filters with signed-power-two coefficients.* Transactions on Signal Processing, 56, 134-139.
- Feynman R. (1982). *Simulating physics with computers, International Journal of Theoretical Physics*. **21** (6): 467- 488.
- Fister I., Yang X.-S., Fong S., Zhuang Y. (2015). *Bat algorithm: Recent Advances,* in Proc:15th International Symposium on Computational Intelligence and Informatics, 19-21 Nov, 2015, Budapest, Hungary.
- ftexploring, 2018. Retrieved from <http://www.ftexploring.com/think/superbugs_p2.html> (last accessed on 22nd April, 2018).
- Ganguly A., Sharma M., *Detection of pathological heart murmurs by feature extraction of phonocardiogram signals*, Journal of Applied and Advanced Research, vol. 2.
- Gentili P., Piazza F. & Uncini A. (1995). *Efficient Genetic Algorithm Design for Power-of-Two FIR Filters.* International Conference on Acoustics, Speech, and Signal Processing. 2, pp. 1268-1271. Detroit, MI: IEEE.
- Gibbs J. W. (1898). *Fourier's Series.* Nature, 59(1522): 200.
- Goddard W. & Melville S. (2004). *Research Methodology: An Introduction.* Juta and Company Ltd.
- Gold B. & Jordan K. (1969). *A direct search procedure for designing finite duration impulse response filters.* Transaction on Audio and Electroacoustics, 17: 33-36.

- Han K. H. & Kim J. H. (2002). *Quantum-inspired evolutionary algorithm for a class of combinatorial optimization*, IEEE Trans on Evolutionary Computation, 6: 580-593.
- Harris, Fredric J. (Jan 1978). *On the use of Windows for Harmonic Analysis with the Discrete Fourier Transform.* Proceedings of the IEEE. **66** (1): 51–83.
- Hartley R. (1996). *Subexpression sharing in filters using canoninc signed digit multipliers.* Transactions on Circuits and Systems II: Analog and Digital Signal Processing, 43: 677-688.
- Hassan R. F. & Abbood A. S. (2013). *Design of Finite Impulse Response Filters based on Genetic Algorithm.* Diyala Journal of Engineering Sciences, 6: 28-39.
- Hassan R. F. & Abood A. S. (2013). *Design of Finite Impulse Response Filters based on Genetic Algorithm,* Diyala Journal of Engg Science, 6: 28-39.
- Herrmann O. (1970). *Design of Nonrecursive Digital Filters with Linear Phase.* Electroic Letters, 6: 328-329.
- Hewitt E. & Hewitt R. E. (1979). *The Gibbs-Willbraham phenomenon: An episode in Fourier analysis.* Archive for History of Exact Sciences, 21(2): 129-160. Retrieved 16 September 2011. Available on-line at: National Chiao Tung University: Open Course Ware: Hewitt & Hewitt 1979.
- Hofstetter E., Oppenheim A., & Siegel J. (1971). *A New Technique for the Design of Nonrecursive Digital Filters.* 5th Annual Princton Conference Information Sciences and Systems, (pp. 64-72). Princeton.
- Holland J. H. (1975). *Adaption in natural and artificial systems.* Ann Arbor, MI, USA: University of Michigan Press.
- Holland J. H. (1992). *Adaption in natural and artificial systems:An Introductory Analysis with Applications to Biology,* Control and Artificial Intelligence, Cambridge, MA, USA: MIT Press.
- Jackson L. B. (1996). *Digital Filters and Signal Processing* (3 ed.). Kluwer.
- Jamil M. & Yang X-S (2013). *A literature survey of benchmark functions for global optimization problems*, International Journal Mathematical Modeling and Numerical Optimization, 4: 150–194.
- Ji D. (2015). The application of Artificial Bee Colony (ABS) Algorithm in FIR Filter Design, in Proc: 12th Int. Conf. on Natural Computing, Fuzzy Systems and Knowledge Discovery, 13-15 August, 2015, Changsha, China.

- Judea P. (1984). Heuristics: intelligent search strategies for computer problem solving, United States: Addison-Wesley Pub. Co., Inc., Reading, MA. P.3. Retrieved June 13, 2017.
- Kaiser J. F. & Schafer R. w. (1980). On the use of the I_0-sinh window for spectrum analysis, IEEE Transactions on Acoustics, Speech and Signal Processing, 28: 105-107.
- Kaiser, J. F. (1966). Digital Filters. In Kuo, F. F. and Kaiser, J. F. (Eds.), *System Analysis by Digital Computer*, chap. 7. New York, Wiley.
- Kar R., Mandal D., Mondal S., & Ghoshal S. P. (2012). *Craziness based Particle Swarm Optimization Algorithm for FIR Band Stop Filter Design*. Swarm and Evolutionary Computation, 58-64.
- Karaboga N. & Cetinkaya B. (2006). *Design of Digital FIR Filters using Differential Evolution Algorithm*. Circuits Systems Signal Processing, *25*: 649-660.
- Karaboga N. & Cetinkaya B. (2014). *Design of minimum phase digital IIR filters by using genetic algorithm*, in: Proc. 6th Nordic Signal Processing Symposium, 11 June 2014, Finland.
- Karakonstantis G. & Roy K. (2007). *An optimal algorithm for low power multiplierless FIR filter design using Chebyshev criterion*. International Conference on Acoustics, Speech and Signal Processing. *2*, pp. 49-52. IEEE.
- Kaye P., Laflamme R. & Mosca M. (2007). *An introduction to quantum computing*, Oxford University Press, 2007, ISBN 0-19-857049-X.
- Kennedy J. & Eberhart R. (1995). Particle swarm optimization. *International Conference on Neural Networks* (pp. 1942-48). Piscataway: IEEE.
- Kennedy J., Eberhart R., & Shi Y. (2001). *Swarm Intelligence.* London: Academic Press.
- Kenny V., Nathal M. & Saldana S. (2014, May 25). *Heuristic Algorithms*. Retrieved March 20, 2018, from Northwestern University Process Optimization Open Textbook on Process Optimization:
- Kirandeep & Singh B. (2015, June). *Differential Evolution Technique for the Design of High-Pass FIR Digital Filter.* International Journal of Scientific Research Engineering & Technology, 4.
- Kodek D. (1980). *Design of optimal finite wordlength FIR digital filters using integer programming techniques.* Transactions on Acoustics, Speech and Signal Processing, 28: 304-308.

- Kodek D. M. (1980). *Design of optimal Finite Wordlength FIR Digital Filters using Integer Programming Techniques*, IEEE Transactions on Acoustics, Speech and Signal Processing, Assp-28: 304-308.
- Kokash N. (2005). *An introduction to heuristic algorithms*, Dept. Informatics Telecommun, 1–8.
- Kokash N. (2005). *An introduction to heuristic algorithms.*
- Konar D., Bhattacharya S., Panigrahi B. K., Ghose M. K. (2017). *An Efficient Pure Color Image Denoising Using Quantum Parallel Bidirectional Self-Organizing Neural Network Architecture*, in: S. Bhattacharya, U. Maulik, P. Dutta (Eds.), Quantum Inspired Computational Intelligence - Reseach and Applications, Elsevier Inc., Cambridge, US, pp. 149-206.
- Kumar A. (n.d.). *Genetic Algorithms*. Retrieved July 30, 2018, from www.geeksforgeeks.org: https://www.geeksforgeeks.org/genetic-algorithms/
- Kumar M. & Rawat R. K. (2015). *Optimal Design of FIR Fractional order differentiator using Cuckoo Search Algorithm*, Article in Expert Systems with Applications, 42, 10.1016/j.eswa.2014.12.020.
- Kumar M., & Rawat, T. K. (2014). *Optimal design of FIR fractional order differentiator using cuckoo search algorithm.* Expert Systems with Applications, 42, 3433-3449.
- Kumar R. & Joytishree. (2012). *Blending Roulette Wheel Selection & Rank Selection in Genetic Algorithms.* International Journal of Machine Learning and Computing, 2, 365-370.
- Laha S. (2015). *A quantum-inspired cuckoo search algorithm for the travelling salesman problem*, in Proc: Int. Conf. on Comput., Communi. and Security, 4-5 Dec, 2015, Pamplemousses, Mauritius.
- Layeb A. & Boussalia S. R. (2012). *A Novel Quantum Inspired Cuckoo Search Algorithm for Bin Packing Problem*, International Journal Information Technology and Computer Science, 5: 58-67.
- Lee A. (2000). *Design of 1-D and 2-D Filters with Finite-wordlength Coefficients Using Genetic Algorithms.* College of Graduate Studies and Research, University of Windsor, Department of Electrical Engineering. Windsor: Alfred Lee 2000.
- Li D. & Lim Y. C. (1993). *A polynomial-time algorithm for designing digital filters with power-of-two coefficients.* International Symposium on Circuits and Systems. 1, pp. 84-87. Chicago: IEEE.

- Li D. (1995). *Minimum number of adders for implementing a multiplier and its application to the design of multiplierless digital filters.* Transaction on Circuits and Systems II: Analog and Digital Signal Processing, 42: 453-460.
- Li Y., Feng S., Zhang X. & Jiao L. (2014). *SAR image segmentation based on quantum-inspired multiobjective evolutionary clustering algorithm*, Inf. Process, Lett. 114: 287-293.
- Lim Y. C., Yang R., Li D. & Song J. (1999). *Signed power-of-two term allocation scheme for the design of digital filters.* Transactions on Circuits and Systems II:Analog and Digital Signal Processing, 46: 577-584.
- Lim Y., & Parker S. (1983). *FIR filter design over a discrete powers-of-two coefficient space.* Transactions on Acoustics, Speech and Signal Processing, 31, 583-591.
- Liu G., Li Y. & He G. (2010). *Design of Digital FIR Filters Using Differential Evolution Algorithm Based on Reserved Genes.* Congress on Evolutionary Computation. Barcelona: IEEE.
- Long H., Xu W. & Wang X. (2010). *Using Selection to Improve Quantum-behaved Particle Swarm Optimization, Control and Decision*, 25: 1499-1506.
- Lu K. & Li H. (2015). *Quantum-behaved Flower Pollination Algorithm*, in Proc: Int. Symp. on Distributed Computing and Appl. for Business Engg. and Sci., 18-24 August, 2015, Guiyang, China.
- Luitel B. & Venayagamoorthy G. K. (2008). *Differential Evolution Particle Swarm Optimization for Digital Filter Design.* Congress on Evolutionary Computation (pp. 3954-3961). HongKong: IEEE.
- Macura W. K. (2018). *Ant Colony Algorithm*. (E. W. Weisstein, Producer) Retrieved 24th July, 2018, from A Wolfram Web Resource : http://mathworld.wolfram.com/AntColonyAlgorithm.html
- Mandal S., Ghoshal S. P., Mukherjee P., Sengupta D., Kar R. & Mandal D. (2012). *Design of Optimal Linear Phase FIR High Pass Filter using Improved Particle Swarm Optimization.* International Journal on Signal and Image Processing, 5-9.
- Mandal S., Kar R., Mandal D. & Ghosal S. P. (2011). *Swarm Intelligence based Optimal Linear Phase FIR High Pass Filter Design using Particle Swarm Optimization with Constriction Factor and Inertia Weight Approach.* International Journal of Electrical and Electronics Engineering, 296-301.

- McClellan J. H. & Parks T. W. (1973). *A Unified Approach to tf»e Design of Optimum FIR Linear Phase Digital Filters.* Transaction on Circuit Theory, CT-20, 697-701.

- Melanie M. (1999). *An introduction to genetic algorithm* (Fifth ed.). London, England: The MIT Press Cambridge, Massachusetts.

- Mitra S. K. (2013). *Digital Signal Processing - A Computer Based Approach* (Third Edition ed.). Mc Graw Hill.

- Mondal S., Ghoshal S. P., Kar R. & Mandal D. (2012). *Novel Particle Swarm Optimization for Low Pass FIR Filter Design.* WSEAS TRANSACTIONS on SIGNAL PROCESSING, 111-120.

- Montanaro A. (2016). *Quantum algorithms: an overview*. npj Quantum Information, 2.

- Naik M., Nath M. R., Wunnava A., Sahany S. & Panda R. (2015). *A new Adaptive Cuckoo Search Algorithm*, International Conference on Recent Trends in Inform. Systems, 9-11 July, 2015, Kolkata, India.

- Najjarzadeh M. & Ayatollahi A. (2008). *FIR Digital Filters Design: Particle Swarm Optimization Utilizing LMS and Minimax Strategies.* International Symposium on Signal Processing and Information Technology (pp. 129-132). Sarajevo: IEEE.

- Nielsen M. A. & Chuang I. L. (2000). *Quantum Computation and Quantum Information.* Cambridge University Press.

- Obitko M. (1998). *Selection*. Retrieved July 30, 2018, from http://www.obitko.com: http://www.obitko.com/tutorials/genetic-algorithms/selection.php

- Oldroyd D. R. (1986). *Charls Darwin's Theory of evolution: A review of our present understanding*, Biology and Philosaphy, 1: 133-168.

- Oner M. (1998). *A Genetic Algorithm for Optimization of Linear Phase FIR Filter Coefficients*, in Proc: 32nd Asilomar Conf. on Signals, Systems & Computers, 1-4 November, 1998 CA, USA.

- Parks T. W., & McClellan J. H. (1972). *Chebyshev Approximation for Nonrecursive Digital filters with Linear Phase.* Transaction on Circuit Theory, CT-19, 189-194.

- Pavlyukevich I. (2007, October 1). *Lévy flights, non-local search and simulated annealing.* Journal of Computational Physics, 226(2): 1830-1844.

- Payne R. B. (2005). Sorenson M. D. & Klitz K., *The Cuckoos*, UK: Oxford University Press.

- Pearl J. (1984). *Heuristics: intelligent search strategies for computer problem solving.* United States: Addison-Wesley Pub. Co., Inc., Reading, MA, Retrieved June 13, 2017.

- Proakis J. G. & Manolakis D. G. (2015). *Digital Signal Processing - Principle,* Algorithms and Applications (18th ed.). PEARSON.

- Rabiner L. R. (1972). *The Design of Finite Impulse Response Digital Filters using Linear Programming Techniques.* The Bell System Technical Journal, 51: 1177-1198.

- Rabiner L. R. (1972, October). *Linear Program Design of Finite Impulse Response Digital Filters.* Transaction on Audio Electroacoustics, 20: 280-288.

- Rabiner L. R., Gold B. & McGonegal C. A. (1970). *An approach to the approximation problem for nonrecursive digital filters.* Transaction on Audio and Electroacoustics, 18: 83-106.

- Rabiner L. R., McClellan J. H. & Parks T. W. (1975). *FIR Digital Filter Design Techniques using Weighted Chebyshev Approximation.* Proceedings of IEEE, 63, 595-610.

- Reddy K. S. & Sahoo S. K. (2015). *An approach for FIR filter coefficient optimization using differential evolution algorithm.* International Journal of Electronics and Communications, 69, 101-108.

- Reddy S. K. (2015). *High Performance VLSI Architecture for Digital FIR Filter Design.* BIRLA INSTITUTE OF TECHNOLOGY & SCIENCE, Electrical & Electronics Engineering, Pilani.

- Saha S. K., Dutta R., Chowdhury R., Kar R., Mandal D. & Ghosal S. P. (2013). *Efficient and Accurate Optimal Linear Phase FIR Filter Design Using Opposition-Based Harmony Search Algorithm*, The Scientific World Journal, http://dx.doi.org/10.1155/2013/320489.
- Saha S. K., Kar R., Mandal D. & Ghosal S. P. (2013). *A Novel Firefly Algoritthm for Optimal Linear Phase FIR Filter Design*, International Journal Swarm Intelligence Research, 4: 29-48.
- Saha S. K., Kar R., Mandal D. & Ghoshal S. P. (2013). *A novel firefly algoritthm for optimallinear phase FIR filter design.* International Journal of Swarm Intelligence, 29-48.

- Salivahanan S., Vallavaraj, A., & Gnanapriya, C. (2007). *Digital Signal Processing.* Tata McGraw Hill.

- Samanta, S., Choudhury A., Dey N., Ashour A. S. & Balas V. E. (2017). *Quantum-inspired evolutionary algorithm for scaling factor optimization during manifold medical information* *embedding.* In S. Bhattacharya, U. Maulik, & P. Dutta, *Quantum Inspired Computational Intelligence Research and Applications* (pp. 285-326). Cambridge: Elsevier Inc. Todd Green.

- Samueli H. (1989). *An improved search algorithm for the design of multiplierless FIR filters with powers-of-two coefficients.* Transactions on Circuits and Systems, 36, 1044-1047.

- Sasahara T. & Suyama K. (2015). *An ACO Approach for Design of CSD Coefficient FIR Filters.* Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA) (pp. 463-468). Hong Kong: IEEE.

- scribol, 2018. Retrieved from <http://scribol.com/environment/animals-environment/the-ant-supercolony-that-stretches-6000km/> (last accessed on 22nd April, 2018).

- Sengupta S. & Basak S. (2016). *Computationally efficient low-pass FIR filter design using Cuckoo Search with adaptive Levy step size.* International Conference on Global Trends in Signal Processing, Information Computing and Communication. Jalgaon: IEEE.

- Sharma S. (2009). *Theory and Design of Finite Impulse Response (FIR) Filters in Digital Signal Processing (with Matlab Programs)* (5th ed.). India: KATSON.

- Shi D. & Yu Y. J. (2011). *Design of linear phase FIR filters with high probability of achieving minimum number of adders.* Transactions on Circuits and Systems I: Regular Papers, 58, 126-136.

- Singh R. & Arya S. K. (2012). *Optimization of IIR digital filters using Particle Swam Optimization*, in: Proc: Int. Conf. on Commun., Inform. and Comp. Techn., 19-20 October 2012, Mumbai, India.
- Singh T., Josan H. S. (2014). *Design of Low Pass Digital FIR Filter Using Cuckoo Search Algorithm*, International Journal of Engineering Research and Applications, 4: 72–77.
- Sivanandam S. N. & Deepa S. N., *Principles of Soft Computing*, 2nd Edition, Willy, India.
- Smith S. W. (2011). *Digital Signal Processing - A Practical Guide fir Engineers and Scientists* (1st ed.). Elsevier.

- Solank S. (2012). *Design FIR Filter with Signed Power of Two Terms using MATLAB*, International Journal of Engineering & Technology 1:1-6.
- stanford 2018a. *Rectangular Window*, retrieved from <https://ccrma.stanford.edu/~jos/sasp/Rectangular_Window.html> (last accessed on 22nd April, 2018).

- stanford 2018b. *Hann or Hanning or Raised Cosine,* retrieved from <https://ccrma.stanford.edu/~jos/sasp/Hann_Hanning_Raised_Cosine.html> (last accessed on 22nd April, 2018).

- stanford 2018c. *Hamming Window,* retrieved from <https://ccrma.stanford.edu/~jos/sasp/Hamming_Window.html> (last accessed on 22nd April, 2018).

- stanford 2018d. *Blackman Window,* retrieved from <https://ccrma.stanford.edu/~jos/st/Use_Blackman_Window.html> (last accessed on 22nd April, 2018).

- stanford 2018e. *Barlett Triangular Window,* retrieved from <https://ccrma.stanford.edu/~jos/sasp/Bartlett_Triangular_Window.htm> (last accessed on 22nd April, 2018).

- Steiglitz K., Parks T. W. & Kaiser J. F. (1992). *METEOR: a constraint-based FIR filter design program.* Transactions on Signal Processing, 40, 1901-1909.

- Storn R. & Price K. (1997). *Differential evolution: a simple and efficient heuristic for global optimization over continuous spaces.* Journal of Global Optimization, 11, 341-59.

- Storn R. (1996). *On the usage of differential evolution for function optimization.* Biennial conference of the North American fuzzy information processing socity, (pp. 519-23). Berkely.

- Sun J., Feng B. & Xu W. (2004). *A global search strategy of quantum behaved particle swarm optimization*, in Proc: Conf. on Cybernetics and Intelligent Systems. Piscataway, 1-3 December, 2004, Singapore.
- Surjanovic S. & Bingham D. (2017). *Virtual Library of Simulation Experiments: Test Functions and Datasets*, Simon Fraser University, Derek Bingham, 2013, Last Updated August 2017 (accessed on 22 April 2018).
- Thapar S., Kaur P. & Aggarwal N. (2012). *A Low Pass FIR Filter Design Using Genetic Algorithm Based Artificial Neural Network.* International Journal of Computer Technology and Electronics Engineering, 99-103.

- Tsutsumi S. & Suyama K. (2014, 1066-1071). *Design of FIR Filters with Discrete Coefficients Using Ant Colony Optimization.* Electronics and Communications in Japan.

- Tzeng S. T. (2004). *Design of 2-D FIR Digitall Filters with Symmetric Properties by Genetic Algorithm Approach.* Asia-Pacific Conference on Circuits and Systems (pp. 417-420). Tainan: IEEE.

- Tzeng S.-T. (2006). *General McClellan Transformation for Designing 2-D Complex FIR Digital Filters by Genetic Algorithm Approach.* The 6th Asian Control Conference, (pp. 609-612). Bali.

- Vasudhara, Mandal D. & Ghoshal S. P. (2013). *Digital FIR Filter Design Using Hybrid Random Particle Swarm Optimization with Differential Evolution.* International Journal of Computational Intelligence Systems, 911-927.

- Ventura D. & Martinez T. (1997). *An artificial nuron with quantum mechanical properties.* International Conference on Artificial Neural Networks and Genetic Algorithms (pp. 482-485). Norwich: Springer.

- WEB CENTER FOR Social Research Methods, 2006. Retrieved from <https://socialresearchmethods.net/kb/stat_t.php> (last accessed on 22nd April 22 2018).

- Williams T., Ahmadi M. & Miller W. C. (2005). *High throughput rate 2-D fir filter designusing singular value decomposition and genetic algorithms.* Fifth Int. Conf. on Inform., Commun. and Signal Processing (pp. 574-578). Bangkok: IEEE.

- Williams T., Ahmadi M. & W.C. Miller W.C. (2005). *High Throughput Rate 2-D Fir Filter Design Using Singular Value Decomposition and Genetic Algorithms*, in Proc: Fifth Int. Conf. on Inform., Commun. and Signal Processing, 6-9 December, 2005, Bangkok, Thailand.

- Wolf R. d. (2011). *Quantum Computing: Lecture Notes.* Amsterdam.

- Xu F., Chang C. H. & Jong C. C. (2007). *Design of low-complexity FIR filters based on signed-powers-of-two coefficients with reusable common subexpression.* Transactions on Compute Aided Design of Integrated Circuits and Systems, 26, 1898-1907.

- Yang X.-S. (2013). Flower Pollination Algorithm for Global Optimization, https://arxiv.org/pdf/1312.5673.pdf (last accessed on 22nd April 2018).

- Yang X.-S. (2014), *Nature Inspired Optimization Algorithms* (1 ed.), Elsevier.

- Yang X.-S. (2014). *Genetic Algorithm.* In X.-S. Yang, *Nature-Inspired Optimization Algorithm* (1 ed., pp. 77-97). London: Elsevier Insights.

- Yang X.-S., Deb S. & Fong S. (2011). Accelerated particle swarm optimization and support vector machine for business optimization and applications, In: Networked digital technologies. Commun. in com.r and inform. Sci., 136: 53-66, Germany: Springer.
- Yang X-S. & Deb S. (2009). *Cuckoo Search via Lévy flights.* World Congress on Nature & Biologically Inspired Computing (pp. 210-214). Coimbatore: IEEE.
- Yang X-S. & Deb S. (2013, June). *Multiobjective cuckoo search for design optimization.* Computers & Operations Research, 40(6): 1616-1624.
- Yang X-S. (2008). *Nature Inspired Meta-heuristics Algorithm.* UK: Luniver Press.
- Yang X-S. (2008). *Nature-Inspired Metaheuristic Algorithms.* Luniver Press.
- Yang X-S. (2009). *Firefly algorithms for multimodal optimization.* In O. Watanabe, & T. Zeugmann (Ed.), *Fifth symposium on stochastic algorithms, foundations and applications. 5792*, pp. 169-78. Lecture nots in computer science.
- Yang X-S. (2010a). *Engineering Optimization: An Introduction with Metaheuristic Applications.* John Wiley & Sons.
- Yang X-S. (2010b), Introduction, Nature-Inspried Metaheuristic Algorithms, Second ed., Luniver, UK, pp. 1-5.
- Yang X-S. (2010c). *Firefly algorithm, stochastic test functions and design optimization.* International Journal of Bio Inspired Computation, 2(2): 78-84.
- Yang X-S. (2010d). A New Metaheuristic Bat-Inspired Algorithm, https://arxiv.org/pdf/1004.4170.pdf (accessed on 20 April 2018).
- Yang X-S. (2011). *Metaheuristic Optimization: Algorithm Analysis and Open Problems*, in: P. M. Pardalos, S. Rebennack (Eds.), Experimental Algorithms. SEA 2011, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, 2011, 6630: pp. 21-32.
- Yang X-S. (2014). *Nature Inspired Optimization Algorithms* (1 ed.). London: Elsevier Insghits.
- Yang X-S., & Deb S. (2010). *Engineering Optimization by Cuckoo Search.* International Journal of Mathematical Modelingl Numerical Optimization, 1(4): 330-343.
- Yang X-S., Cui Z., Xiao R., Gandomi A. & Karamanoglu M. (2013). *Swarm intelligence and bio-inspired computation: theory and applications.* London: Elsevier.
- Yanofsky N. (2007). *An Introduction to Quantum Computing.* New York.

- Yao C. Y. & Chien C. J. (2002). *A partial milp algorithm for the design of linear phase FIR filters with spt oefficients.* Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 46: 2302-2310.

- Yarizadeh-Beneh M. (2016). *Improving the refueling cycle of a WWER-1000 using cuckoo search method and thermal-neutronic coupling of PARCS v2.7, COBRA-EN and WIMSD-5B codes*, Nuclear Engineering and Design, 310: 247-257.
- Yli Kaakinen J. & Saramaki T. (2001). *A systematic algorithm for the design of multiplierless FIR filters.* International Symposium on Circuits and Systems. 2, pp. 185-188. Sydney: IEEE.

- Yu Y. J., & Lim Y. C. (2007). *Design of linear phase FIR filters in subexpression space using mixed integer linear programming.* Transactions on Circuits and Systems I: Regular Papers, 54: 2330-2338.

- Yu Y. J., Shi D. & Lim Y. C. (2009). *Design of extrapolated impulse response FIR filters with residual compensation in subexpression space.* Transactions on Circuits and Systems I: Regular Papers, 56: 2621-2633.

- Zhao Q. & Tadokoro Y. (1988). *A simple design of FIR filters with powers-of-two coefficients.* Transactions on Circuits and Systems, 35: 566-570.