1-1-2019

# Digital Filter Design Using Improved Artificial Bee Colony Algorithms

Rija Raju
*University of Windsor*

# Digital Filter Design Using Improved Artificial Bee Colony Algorithms

By

**Rija Raju**

A Dissertation
Submitted to the Faculty of Graduate Studies
through the Department of Electrical and Computer Engineering
in Partial Fulfillment of the Requirements for
the Degree of Doctor of Philosophy
at the University of Windsor

Windsor, Ontario, Canada

2019

**Digital Filter Design Using Improved Artificial Bee Colony Algorithms**

by

**Rija Raju**

APPROVED BY:

---

W.-K. Ling, External Examiner
Guangdong University of Technology

---

Z. Kobti
School of Computer Science

---

H. Wu
Department of Electrical and Computer Engineering

---

N. Kar
Department of Electrical and Computer Engineering

---

H. K. Kwan, Advisor
Department of Electrical and Computer Engineering

December 17, 2019

# DECLARATION OF CO-AUTHORSHIP/ PREVIOUS PUBLICATION

## *I. Co-Authorship*

I hereby declare that this dissertation incorporates material that is result of joint research, as follows:

This dissertation also incorporates the outcome of a joint research undertaken under the supervision of and in collaboration with my advisor Dr. H. K. Kwan. The collaborative work is covered mainly in Chapter 3, Chapter 4, Chapter 5 and Chapter 6 of the dissertation. In the dissertation, the contribution of ideas, the experimental designs, the data analysis and interpretation, and the writing were performed by the author, and the contribution of my advisor include the provision of ideas, the formulation of design problems, the analysis of the design results, and the writing and editing help. Dr. A. Jiang provided the MATLAB codes for the partial l1 optimization in [33] and programming advice.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my dissertation and have obtained written permission from each of the co-author(s) to include the above material(s) in my dissertation.

I certify that, with the above qualification, this dissertation, and the research to which it refers, is the product of my own work.

## *II. Previous Publication*

This dissertation includes 4 original papers that have been previously published in peer reviewed conference proceedings, as follows:

| Dissertation chapter | Publication title/full citation | Publication status |
|---|---|---|
| Chapter 3 | H. K. Kwan and R. Raju, "Minimax design of linear phase FIR differentiators using artificial bee colony algorithm," in *Proc. of 8th International Conference on Wireless Communications and Signal Processing (WCSP 2016)*, Yangzhou, China, Oct. 13-15, 2016, pp. 1-4. | Published |
| Chapter 4 | R. Raju, H. K. Kwan and A. Jiang, "Sparse FIR filter design using artificial bee colony algorithm," in *Proc. of IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS 2018),* Windsor, Ontario, Canada, Aug. 2018, pp. 956-959. | Published |
| Chapter 5 | R. Raju and H. K. Kwan, "FIR filter design using multiobjective artificial bee colony algorithm," in *Proc. of 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE 2017)*, Windsor, Ontario, Canada, Apr. 30-May 3, 2017, pp. 1-4. | Published |
| Chapter 6 | R. Raju and H. K. Kwan, "IIR filter design using multiobjective artificial bee colony algorithm," in *Proc. of 2018 IEEE 31th Canadian Conference on Electrical and Computer Engineering (CCECE 2018),* Quebec City, Quebec, Ontario, Canada, May 13-16, 2018, pp. 1-4. | Published |

I certify that I have obtained a written permission from the copyright owners to include the above published materials in my dissertation. I certify that the above material describes work completed during my registration as graduate student at the University of Windsor.

I declare that, to the best of my knowledge, my dissertation does not infringe upon anyone's copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my dissertation, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my dissertation and have included copies of such copyright clearances to my appendix.

I declare that this is a true copy of my dissertation, including any final revisions, as approved by my dissertation committee and the Graduate Studies office, and that this dissertation has not been submitted for a higher degree to any other University or Institution.

## ABSTRACT

Digital filters are often used in digital signal processing applications. The design objective of a digital filter is to find the optimal set of filter coefficients, which satisfies the desired specifications of magnitude and group delay responses. Evolutionary algorithms are population-based metaheuristic algorithms inspired by the biological behaviors of species. Compared to gradient-based optimization algorithms such as steepest descent and Newton's like methods, these bio-inspired algorithms have the advantages of not getting stuck at local optima and being independent of the starting point in the solution space. The limitations of evolutionary algorithms include the presence of control parameters, problem specific tuning procedure, premature convergence and slower convergence rate. The artificial bee colony (ABC) algorithm is a swarm-based search metaheuristic algorithm inspired by the foraging behaviors of honey bee colonies, with the benefit of a relatively fewer control parameters. In its original form, the ABC algorithm has certain limitations such as low convergence rate, and insufficient balance between exploration and exploitation in the search equations. In this dissertation, an ABC-AMR algorithm is proposed by incorporating an adaptive modification rate (AMR) into the original ABC algorithm to increase convergence rate by adjusting the balance between exploration and exploitation in the search equations through an adaptive determination of the number of parameters to be updated in every iteration. A constrained ABC-AMR algorithm is also developed for solving constrained optimization problems.

There are many real-world problems requiring simultaneous optimizations of more than one conflicting objectives. Multiobjective (MO) optimization produces a set of feasible

solutions called the Pareto front instead of a single optimum solution. For multiobjective optimization, if a decision maker's preferences can be incorporated during the optimization process, the search process can be confined to the region of interest instead of searching the entire region. In this dissertation, two algorithms are developed for such incorporation. The first one is a reference-point-based MOABC algorithm in which a decision maker's preferences are included in the optimization process as the reference point. The second one is a physical-programming-based MOABC algorithm in which physical programming is used for setting the region of interest of a decision maker.

In this dissertation, the four developed algorithms are applied to solve digital filter design problems. The ABC-AMR algorithm is used to design Types 3 and 4 linear phase FIR differentiators, and the results are compared to those obtained by the original ABC algorithm, three improved ABC algorithms, and the Parks-McClellan algorithm. The constrained ABC-AMR algorithm is applied to the design of sparse Type 1 linear phase FIR filters of filter orders 60, 70 and 80, and the results are compared to three state-of-the-art design methods. The reference-point-based multiobjective ABC algorithm is used to design of asymmetric lowpass, highpass, bandpass and bandstop FIR filters, and the results are compared to those obtained by the preference-based multiobjective differential evolution algorithm. The physical-programming-based multiobjective ABC algorithm is used to design IIR lowpass, highpass and bandpass filters, and the results are compared to three state-of-the-art design methods. Based on the obtained design results, the four design algorithms are shown to be competitive as compared to the state-of-the-art design methods.

*To my*

*loving daughter Jenna*

*and beloved husband Rojan*

## ACKNOWLEDGEMENTS

First and foremost, I would like to thank God Almighty for giving me the strength, knowledge, ability and opportunity to undertake this research study and to persevere and complete it satisfactorily. Without his blessings, this achievement would not have been possible.

I would like to express my sincere gratitude to my advisor Prof. Hon Keung Kwan, for his patience, motivation, valuable advice, and support during my graduate studies at the University of Windsor; for suggesting the research topics, the ABC algorithm, and for providing guidance, feedbacks, rewriting and editing help on my dissertation. I could not complete my research work as reported in this dissertation without his help. I am also grateful to my Doctoral committee members: Dr. Ziad Kobti, Dr. Huapeng Wu, and Dr. Narayan Kar, for their valuable suggestions. I thank the external examiner Dr. Wing-Kuen Ling for his suggestions and help in improving the dissertation. I would also like to thank the departmental graduate secretary, Ms. Andria Ballo for all her help during my studies at the University of Windsor. I would like to thank my fellow graduate students, especially Dr. Miao Zhang, of the ISPLab for their help and support.

I would like to express my gratitude and appreciation to my parents for their blessings, support and prayers with me in whatever I pursue.

Lastly, my deepest gratitude goes to my husband Rojan, for his unconditional support and optimism. Without his help this dissertation work would have never come to a successful completion.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## LIST OF ACRONYMS

| | |
|---|---|
| 1-D | One-Dimensional |
| 2-D | Two-Dimensional |
| ABC | Artificial Bee Colony Algorithm |
| ABC-AMR | Artificial Bee Colony Algorithm with Adaptive Modification Rate |
| CSA | Cuckoo Search Algorithm |
| dB | Decibel |
| DE | Differential Evolution |
| DM | Decision Maker |
| EA | Evolutionary Algorithm |
| FIR | Finite Impulse Response |
| GA | Genetic Algorithm |
| GABC | Gbest guided ABC |
| GME | Generalized Multiple Exchange |
| HSA | Harmony Search Algorithm |
| IEMO | Interactive Evolutionary Multiobjective Optimization |
| IIR | Infinite Impulse Response |
| IRLS | Iterative Reweighted Least Squares |
| LS-MOEA | Local Search Operator Enhanced Multiobjective Evolutionary Algorithm |
| MOABC | Multiobjective Artificial Bee Colony |
| MODE | Multiobjective Differential Evolution |
| MOEA | Multiobjective Evolutionary Algorithm |

| | |
|---|---|
| MOO | Multiobjective Optimization |
| NP | Nondeterministic Polynomial time |
| NSGA | Nondominated Sorting Genetic Algorithm |
| OL | Orthogonal Learning |
| PAES | Pareto Archived Evolution Strategy |
| PM | Parks-McClellan |
| PP | Physical Programming |
| PSO | Particle Swarm Optimization |
| SDP | Semidefinite Programming |
| SOCP | Second Order Cone Programming |
| SPEA | Strength Pareto Evolutionary Algorithm |
| SP | Spherical Pruning |
| TLBO | Teaching Learning-Based Optimization |

# CHAPTER 1

# INTRODUCTION

## *1.1 Introduction to Digital Filter Design*

Electronic filters are circuits capable of passing certain frequency signals to extract useful information. The electronic filters may be analog or digital depending on the components used. The analog filters operate on continuous time analog signals, whereas digital filter performs mathematical operations on digital signals. Unlike analog filters which requires active and passive physical components, the digital filters can be implemented on computers.

Digital filters can be mathematically expressed by the constant coefficient difference equation:

$$y(n) = \sum_{k=0}^{M} b(k)x(n-k) - \sum_{k=1}^{N} a(k)y(n-k) \tag{1.1}$$

where $b(k)$ and $a(k)$ are the forward tap coefficients and feedback tap coefficients respectively. The transfer function of the digital filter can be expressed as,

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^{M} b(k)z^{-k}}{1 + \sum_{k=1}^{N} a(k)z^{-k}} \tag{1.2}$$

The digital filters can be classified into two categories finite impulse response (FIR) and infinite impulse response (IIR) digital filters depending on the length of their impulse responses and location of poles.

### *1.1.1 Finite Impulse Response Filters*

A finite impulse response filter is based on the feed forward difference equation, which means that the output of the system does not depend on the past or future values of output but depends only on the present value of the input. FIR digital filters include asymmetric FIR filters and symmetric FIR digital filters.

The asymmetric FIR filters are a class of causal filters with the difference equation and transfer function is as expressed below,

$$y(n) = \sum_{k=0}^{N} b(k)x(n-k) \tag{1.3}$$

and

$$H(z) = \frac{Y(z)}{X(z)} = \sum\nolimits_{k=0}^{N} b(k)z^{-k} \tag{1.4}$$

The frequency response of filter can be found by substituting $z = e^{j\omega T}$, where $\omega$ is the frequency of the input signal,

$$H(\omega) = \sum_{n=0}^{N} b_n\, e^{-j\omega nT}$$

$$= \sum_{n=0}^{N} b_n \cos(\omega nT) - j\sum_{n=0}^{N} b_n \sin(\omega nT)$$

$$= |\,H(\omega)|e^{j\theta(w)} \tag{1.5}$$

In equation 1.5, the magnitude response $|H(w)|$is equal to,

$$|H(w)| = \left\{\left[\sum\nolimits_{n=0}^{N} b_n \cos nwT\right]^2 + \left[\sum\nolimits_{n=0}^{N} b_n \sin nwT\right]^2\right\}^{\frac{1}{2}} \tag{1.6}$$

and the phase response $\theta(w)$ is equal to,

$$\theta(w) = -\,tan^{-1}\left[\frac{\sum_{n=0}^{N} b_n\, sin\, nwT}{\sum_{n=0}^{N} b_n\, cos\, nwT}\right] \tag{1.7}$$

From equation 1.7, the group delay $\tau(w)$ can be expressed as,

$$\tau(w) = \frac{\partial\theta(w)}{\partial wT} = \frac{1}{1+c^2}\frac{\partial c}{\partial wT} \tag{1.8}$$

where,

$$c = \frac{\sum_{n=0}^{N} b_n\, sin\, nwT}{\sum_{n=0}^{N} b_n\, cos\, nwT} \tag{1.9}$$

The symmetric FIR filters have constant group delay, and the filter coefficients are either symmetric or anti symmetric with respect to mid-point. A filter of order $N$ or length $M = N+1$ is said to be linear phase if it satisfies the following equation,

$$h(n) = \pm h(M-1-n) \tag{1.10}$$

where $n = 0,1,2, \dots \dots \dots \dots \dots, M-1$.

Depending on the type of symmetry, there are four types of linear phase FIR filters; Type 1, Type 2, Type 3 and Type 4.

In Type 1 filters, the filter order, $N(=M-1)$ is even and the coefficients are symmetrically distributed,

$$h(n) = h(M-1-n) \tag{1.11}$$

where $n = 0,1,2, \dots \dots \dots \dots \dots, M-1$ and the frequency response $H(\omega)$ is given by,

$$H(\omega) = e^{-\frac{j\omega T(M-1)}{2}}\left(h\left(\frac{M-1}{2}\right) + \sum_{n=0}^{(M-3)/2} 2h(n)\cos\left(\frac{M-1}{2} - n\right)\omega T\right) \tag{1.12}$$

In Type 2 filters, the filter order, $N(= M - 1)$ is odd and the coefficients are symmetrically distributed,

$$h(n) = h(M - 1 - n) \tag{1.13}$$

where $n = 0,1,2, \dots \dots \dots, M - 1$ and the frequency response $H(\omega)$ is given by,

$$H(\omega) = e^{-\frac{j\omega T(M-1)}{2}} \sum_{n=0}^{\frac{M}{2}-1} 2h(n) \cos\left(\frac{M-1}{2} - n\right) \omega T \tag{1.14}$$

In Type 3 filters, the filter order, $N(= M - 1)$ is even and the coefficients are anti-symmetrically distributed,

$$h(n) = -h(M - 1 - n) \tag{1.15}$$

where $n = 0,1,2, \dots \dots \dots, M - 1$ and the frequency response $H(\omega)$ is given by,

$$H(\omega) = je^{-\frac{j\omega T(M-1)}{2}} \left( \sum_{n=0}^{(M-3)/2} 2h(n) \sin\left(\frac{M-1}{2} - n\right) \omega T \right) \tag{1.16}$$

In Type 4 filters, the filter order, $N(= M - 1)$ is odd and the coefficients are anti - symmetrically distributed,

$$h(n) = -h(M - 1 - n) \tag{1.17}$$

where $n = 0,1,2, \dots \dots \dots, M - 1$ and the frequency response $H(\omega)$ is given by,

$$H(\omega) = je^{-\frac{j\omega T(M-1)}{2}} \left( \sum_{n=0}^{M/2-1} 2h(n) \sin\left(\frac{M-1}{2} - n\right) \omega T \right) \tag{1.18}$$

### *1.1.2 Infinite Impulse Response Filters*

IIR filters include the following; direct-form general IIR filter; direct-form allpass IIR filter; cascade-form general IIR filter and cascade-form allpass IIR filter.

Direct-form general IIR filter consisting of $M$th order numerator and $N$th order denominator transfer function can be expressed as,

$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^{M} b(k) z^{-k}}{1 + \sum_{k=1}^{N} a(k) z^{-k}} = \sum_{n=0}^{\infty} c(n) z^{-n} \tag{1.19}$$

where $B(z)$ and $A(z)$ are polynomials written in ascending powers of $z^{-1}$, $M$ can be smaller or larger than $N$. The coefficients $\boldsymbol{c}(n)$ for $n \geq 0$ represent the impulse response values of the digital filter. The corresponding coefficient vector $\boldsymbol{c}$ consisting of $M + N + 1$ distinct coefficients can be expressed as,

$$\boldsymbol{c} = [b_0 \; b_1 \; b_2 \; \dots \; b_{M-1} b_M \; a_0 \; a_1 \; a_2 \; \dots \; a_{N-1} a_N \,]^T \tag{1.20}$$

Direct-form allpass IIR filter can characterized by a unity magnitude response throughout the frequency band and its group delay response is a function of its coefficient values. It can be used to equalize the group delay of another digital filter or a system connected in cascade. The direct-form transfer function of an $N$th-order allpass IIR filter ($N$ can be even or odd) can be expressed as,

$$H_{AP}(z) = \frac{\sum_{n=0}^{N} a_{N-n} z^{-n}}{\sum_{n=0}^{N} a_n z^{-n}} = z^{-N} \frac{\sum_{n=0}^{N} a_n z^{n}}{\sum_{n=0}^{N} a_n z^{-n}} \tag{1.21}$$

The coefficient vector $\boldsymbol{c}$ consisting of $N + 1$distinct coefficients can be expressed as,

$$\boldsymbol{c} = [a_0 \; a_1 \; a_2 \; \dots \; a_{N-1} a_N \,]^T \tag{1.22}$$

The frequency response of a direct-form allpass IIR filter can be evaluated by substituting $z = e^{j\omega T}$ into its digital transfer function equation 1.21, the magnitude response is given as,

$$|H_{AP}(\omega)|=1 \tag{1.23}$$

Cascade-form general IIR filters can be obtained by combining two or more direct-form structures. Assuming both the numerator and denominator transfer function are of same order such that $M = N$, the cascade-form transfer function of an even $N$th order IIR filter can be expressed as,

$$H(z) = b_0 \prod_{n=1}^{\frac{N}{2}} \frac{B_n(z)}{A_n(z)} \tag{1.24}$$

$$= b_0 \prod_{n=1}^{\frac{N}{2}} \frac{(1 + b_{1n}z^{-1} + b_{2n}z^{-2})}{(1 + a_{1n}z^{-1} + a_{2n}z^{-2})} = \sum_{k=0}^{\infty} c(n)z^{-n}$$

where $b_{1n}, b_{2n}, a_{1n}, a_{2n}$ with $n = 1\ to \frac{N}{2}$ are real valued coefficients, and $b_0$ is a scaling constant. The coefficients $\boldsymbol{c}(n)$ for $n \geq 0$ represents the impulse response values of IIR filter. The corresponding coefficient vector $\boldsymbol{c}$ consisting of $2N + 1$ distinct coefficients can be expressed as,

$$\boldsymbol{c} = \left[b_{11}\ b_{21}\ a_{11}\ a_{21}\ \dots\ b_{1,\frac{N}{2}} b_{2,\frac{N}{2}}\ a_{1,\frac{N}{2}}\ a_{2,\frac{N}{2}} b_0\right]^T \tag{1.25}$$

Cascade-form allpass IIR filter of an even $N$ th-order can be expressed as,

$$H_{AP}(z) = z^{-N} \prod_{n=1}^{\frac{N}{2}} \frac{(1 + a_{1n}z^{-1} + a_{2n}z^{-2})}{(1 + a_{1n}z^{-1} + a_{2n}z^{-2})} \tag{1.26}$$

The corresponding coefficient vector $\boldsymbol{c}$ consisting of $N$ distinct coefficients can be expressed as,

$$\boldsymbol{c} = \left[a_{11}\ a_{21}\ a_{12}\ a_{22}\ \dots\ a_{1n}\ a_{2n} \dots\ a_{1,\frac{N}{2}}\ a_{2,\frac{N}{2}}\right]^T \tag{1.27}$$

The frequency response of a cascade-form allpass IIR filter can be evaluated by substituting $z = e^{j\omega T}$ into its digital transfer function equation 1.26 and magnitude response is given by,

$$|H_{AP}(\omega)|=1 \qquad (1.28)$$

Two typical classes of design optimization methods for digital filters are and evolutionary optimization [1] and mathematical optimization [2]. A number of useful reference books on digital filter design methods are listed under [3]-[7] and a number of general reference books on digital signal processing are listed in [8]-[10]. A collection of papers on IIR and FIR filter design methods are listed in [11]-[122]. In general, FIR digital filters can be subdivided into linear phase FIR digital filters and nonlinear phase FIR digital filters. The design of linear phase FIR digital filters is described in [11]-[19]; the design of differentiators and integrators are described in [20]-[23]; the design of sparse linear phase FIR digital filters is described in [24]-[46]; and the design of nonlinear phase (or general or asymmetric) FIR digital filters are described in [47]-[60]. An IIR digital filter can be designed to approximate given magnitude response in both passband(s) and stopband(s) and linear phase response in passband(s). The design of IIR digital filters are described in [61]-[85]and adaptive digital filters in [86]-[93]. The design of variable IIR digital filters are described in [94]-110] and variable FIR digital filters is described in [111]. The designs of 2-dimensional FIR digital filters are described in [112]-[114] and IIR digital filters are described in [115]-[122].

## *1.2 Limitations of Evolutionary Algorithm in Digital Filter Design*

Classical methods such as steepest descent and Newton like methods have several shortcomings such as sensitivity to initial points, difficulty in analytical calculation of Hessian matrix, and optimal step size requirement to minimize the objective function value progressively, making it impractical to solve problems with many variables. Evolutionary algorithms have benefits over the classical optimization methods and can be applied efficiently to solve nondifferentiable, multimodal, non-convex, non-separable problems.

Nevertheless, when evolutionary algorithms are used for digital filter design various challenges has been faced, Figure 1.1 shows various limitations of evolutionary algorithm in digital filter design.

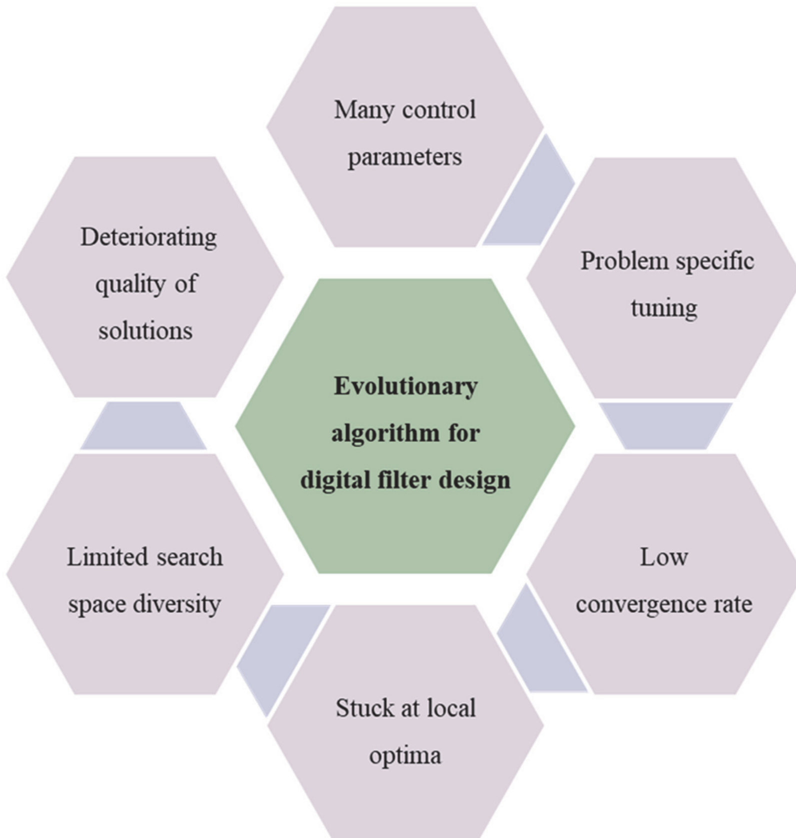


**Figure 1.1** Limitations of Evolutionary Algorithms in Digital Filter Design

### *1.2.1 Many Control Parameters and Problem Specific Tuning*

Conventional algorithms such genetic algorithm (GA), particle swarm optimization (PSO), and differential evolution (DE) contain many control parameters and each of these parameters must be tuned to their optimal value for best performance. Modifying each of these parameters for filter design application require a tedious task of trial and error run.

Also, same set of parameters that works well for one problem does not guarantee a global optimum for another problem with same algorithm, so problem specific tuning of parameters is required in every task.

### *1.2.2 Low Convergence Rate*

Evolutionary algorithms are inspired by the biological process of natural selection and mutation, it is a slow process and needs long computation time to reach global optimum.

### *1.2.3 Stuck at Local Optima*

Even though it is easy to customize the evolutionary algorithms for any application, it is important to choose the best suited algorithm for a given problem. The wrong configuration can lead to premature convergence to a local optimum solution and will not yield global optimum.

### *1.2.4 Limited Search Space Diversity*

In general, for reducing the longer computation time, instead of initializing with a random population, optimization process is seeded with a good candidate solution that is previously known or created. This process is found to reduce the diversity of search space, especially in higher dimension problems.

### *1.2.5 Deteriorating Quality of Solutions with Increase in Dimensionality*

When a limited search space is applied to non-convex, non-differentiable, multimodal, composite functions the quality of solution deteriorates with increase in dimensionality. In filter design applications, peak error value of designed filter cannot be reduced to an optimal value with the increase in filter order.

## *1.3 Motivation*

Classical optimization methods and conventional evolutionary algorithms have certain limitations when applied to digital filter design. In order to overcome these limitations, this dissertation focuses on an improved ABC algorithm for the design of optimal FIR and IIR digital filters.

ABC algorithm is a swarm-based search algorithm inspired by the social cognitive behavior of honey bees. Basic ABC algorithm works better than most of the conventional evolutionary algorithm in terms of peak error values, and it is easy to tune the algorithm towards any specific problem. However, it faces some difficulties such as lower convergence rate, getting stuck at local optimum and difficulty in minimizing the peak error values of the higher order digital filters. The above said problems are a result of insufficient balance between the exploration and exploitation in the search equation. Exploration refers to investigating unknown regions in the solution space to discover global optimum and exploitation refers to applying knowledge about previous good solution to find a better solution. The former occurs at initial stages of optimization while latter at later stages of optimization. Though these two techniques contradict each other, a proper balance between them is necessary for obtaining optimal results. Many variants of ABC algorithm have been developed to address the concerning issues, most of them improves the exploitation by directing the search towards the best solution, but this will limit the diversity in the search space. As filter design problem is analyzed, in order to lower peak error value and satisfy design constraints of higher order filters, new solutions must be introduced into the solution space. So, in this dissertation, a novel improvement known as adaptive modification rate is introduced to the original ABC algorithm, which mutates the parameters in the solution space adaptively.

Research in the field of evolutionary computation is generally limited to single-objective optimization but most of the real-life problems involve optimization of more than one competing objectives. Instead of finding a single optimum solution, these types of problems with the conflicting objectives can be solved using multiobjective optimization (MOO). MOO generates a set of optimal solutions in the objective space, known as Pareto front. At the end of optimization process, the decision maker (DM) choose a single solution from the Pareto front according to his/her preference. In MOO, there are certain limitations encountered as the number of objective increases such as, difficulty in visualization of objective space, prominence of nondominated solutions which slows down the convergence rate, an exponential increase in population size to meet population diversity. If preference of decision maker can be incorporated into optimization process, a preferred and smaller set of Pareto optimal solutions near the region of interest can be found. It requires the decision maker to suggest a reference direction, a reference point or clues to guide the search toward the region of interest. Different approaches can be used to incorporate the decision maker's preferences into the optimization process. In *a posteriori* methods, preferences can be used at the end after Pareto front has been completely determined whereas in *a priori* methods, preferences are given at the beginning of search process which requires the decision maker to have some high-level information about the objectives initially. Interactive methods involve the preferences to be set up interactively during the optimization process.

In this dissertation, the preferences are incorporated into multiobjective optimization *a priori* by physical programming approach and reference point-based approach.

### *1.4 Main Contributions*

Digital filter design is an approximation problem, in which a designer tries to find a set of filter coefficients which provides the best approximation of a desired filter. Even though, it is impossible to produce exact magnitude or phase response of desired filter the classical methods such as Butterworth and Chebyshev methods can be applied for the design of optimal basic filters. Design of filters with arbitrary magnitude and phase response can only be formulated as complex approximation problem and can be solved using evolutionary algorithms. Using ABC algorithm, improved ABC algorithms and ABC-AMR algorithm, various digital filters are designed in both single-objective space and multiobjective space. The main contributions are listed below:

- The dissertation provides an in-depth analysis of ABC algorithm based digital filter design, its advantages, limitations and modifications to be applied for improving its performance in filter design. Initially, an investigation has been performed into the modifications available in the literature. Various digital filters are designed using basic ABC algorithm, its variants and their error values have been compared.
- A new and improved ABC known as ABC-AMR is proposed for digital filter design. Various digital filters are designed to evaluate the performance of the proposed method. Design results from linear phase Type 3 and Type 4 differentiators has been published in - H. K. Kwan and R. Raju, "Minimax design of linear phase FIR

differentiators using artificial bee colony algorithm," in *Proc. of 8th International Conference on Wireless Communications and Signal Processing (WCSP 2016)*, Yangzhou, China, Oct. 13-15, 2016, pp. 1-4. Simulation results indicate that the proposed method can be used successfully to design various digital filters.

- In order to minimize hardware requirement in filter design problems, another class of digital filters known as sparse filters are designed using constrained ABC-AMR algorithm and iterative shrinkage technique. The work has been published in - R. Raju, H. K. Kwan and A. Jiang, "Sparse FIR filter design using artificial bee colony algorithm," in *Proc. of IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS 2018),* Windsor, Ontario, Canada, Aug. 2018, pp. 956-959. Using constrained ABC-AMR algorithm, an increase in sparsity of digital filter can be achieved. Sparse digital filters can be used in applications where computational cost and hardware complexities are critical, because located sparse or zero-valued coefficients do not require multiplications.
- For designing asymmetric FIR filters in a multiobjective space, a user can provide a reference point and the search can be directed towards preferred regions in the Pareto front by minimizing the normalized Euclidean distance towards the reference point. Using the reference-point-based multiobjective ABC, asymmetric FIR filters are designed, and the work has been published in - R. Raju and H. K. Kwan, "FIR filter design using multiobjective artificial bee colony algorithm," in *Proc. of 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE 2017)*, Windsor, Ontario, Canada, Apr. 30-May 3, 2017, pp. 1-4. Comparing the obtained design results with those obtained by the multiobjective differential evolution, lower error values can be obtained.
- While dealing with multiobjective space, most of the algorithms try to improve solutions in Pareto front which are outside the region of interest. Thus, a decision maker's preferences are introduced into the optimization process using a physical programming approach. In this approach, a decision maker can set preferences using

different degrees of desirability such as highly desirable (HD), desirable (D), tolerable (T), undesirable (U) and highly undesirable (HU). Solutions in undesirable (U) and highly undesirable (HU) are not considered in the optimization process. Using a physical programming method, IIR filters are designed and the work has been published in - R. Raju and H. K. Kwan, “IIR filter design using multiobjective artificial bee colony algorithm,” in *Proc. of 2018 IEEE 31th Canadian Conference on Electrical and Computer Engineering (CCECE 2018),* Quebec City, Quebec, Ontario, Canada, May 13-16, 2018, pp. 1-4. The proposed design method can achieve similar or better results when compared to state-of-the-art design methods.

### *1.5 Organization*

The dissertation is organized into six chapters: In Chapter 2, variants of ABC algorithm, their advantages and limitations when applied to filter design applications is briefly described. The multiobjective evolutionary algorithms and its shortcomings are also described in the same chapter. Chapter 3 proposes an improvement applied to the original ABC algorithm, called the ABC-AMR algorithm, which is then used in the design of Type 3, Type 4 linear phase differentiators and the results are compared to those obtained by other variants of the ABC algorithm. In Chapter 4, the constrained ABC-AMR algorithm is applied to design sparse filters, and the results are compared with those obtained by other design methods in the literature. In Chapter 5, a reference-point approach is used to incorporate a decision maker’s preferences into optimization process and a multiobjective error function is formulated for the design of asymmetric FIR filters. In Chapter 6, single-objective ABC algorithm is extended to multiobjective space and the preferences of a decision maker are incorporated into optimization process using a physical programming approach. This reduces computational complexities by directing a search towards the region of interest. Using a physical-programming-based multiobjective ABC, IIR filters are designed. Finally, Chapter 7 concludes with the main findings of this dissertation and makes suggestions for future research.

# CHAPTER 2

# LITERATURE SURVEY

Evolutionary algorithms (EA) are population-based metaheuristic search methods that imitate the processes of Darwinian Evolution. Given a set of potential solutions, evolutionary algorithms apply the principle of survival of the fittest to discover optimal solutions in a search space. New individuals in each generation are created by selecting the parent individuals from the existing population according to their level of fitness and by applying principle of natural genetics. This process will improve the quality of individuals in each generation and finally evolve to an optimal solution.

Evolutionary algorithms are inspired by natural process such as reproduction, selection, recombination and mutation. Every individual in the population represents a single possible solution of the optimization problem. EA starts with a set of randomly initialized population. Fitness value of the solutions is calculated by evaluating the objective function for every individual. The individuals with higher fitness value represent the better-quality solutions and some of these individuals are chosen to seed the next generation by applying recombination or mutation. If optimization criteria or maximum number of generations are not met, new generation will be started to produce a new set of individuals. Recombination is an operator in which two or more selected individuals are combined to produce one or more offsprings. Each offspring is then mutated, and its fitness value is calculated. If a new offspring is better than its parents, it is inserted into the current population producing an individual in a new generation. This new generation becomes the current population and the iterative process repeats until it reaches the optimum solution.

The evolutionary algorithm can be applied to all types of problems in diverse fields, such as economics, arts, engineering, biology, marketing, operations research etc. There are many population-based stochastic optimization algorithms based on the principle of evolution and some of the popular algorithms are as follows: genetic algorithms (GA) by

Holland and Goldberg [123]-[124]; particle swarm optimization (PSO) by Kennedy and Eberhart [125]; ant colony optimization (ACO) by Dorigo and Stutzle [126]; differential evolution (DE) by Storn and Price [127]; simulated annealing by Kirkpatrick *et al*. [128].

## *2.1 Artificial Bee Colony Algorithm (ABC)*

The ABC algorithm is metaheuristic optimization algorithm defined by Dervis Karaboga in 2005 [129], based on collective intelligent biological behavior of honey bee colonies.

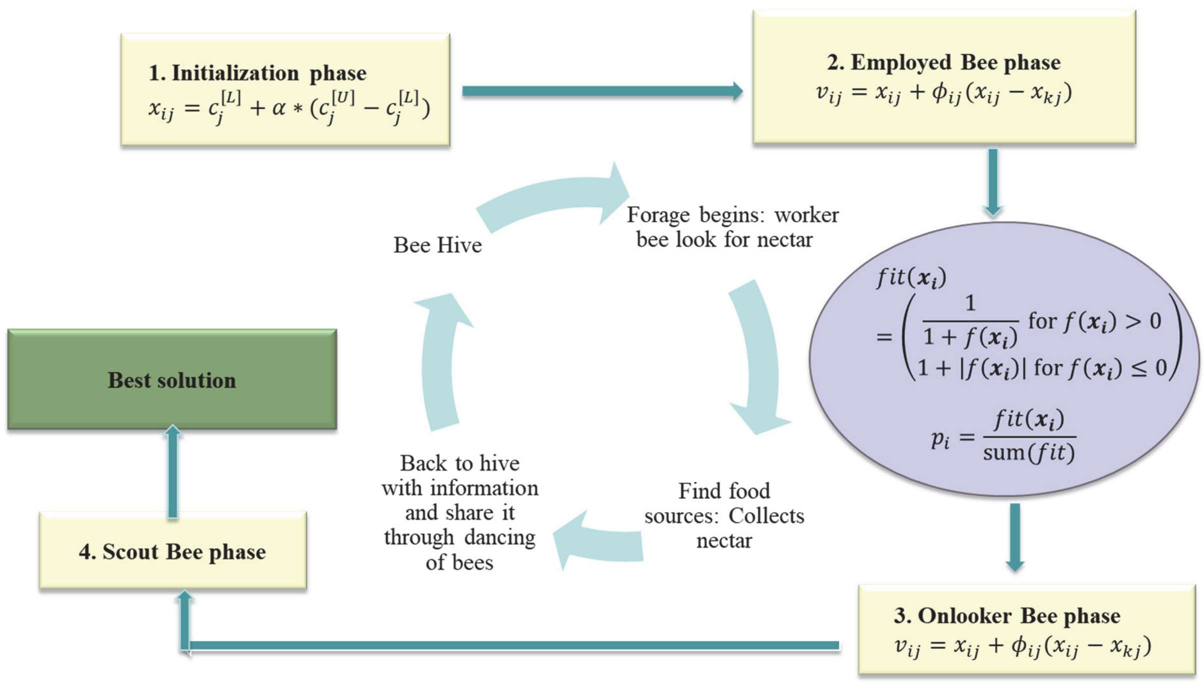


**Figure 2.1** Schematic Representation of Foraging Behavior of Honey Bees

Honey bees is an eusocial flying insect characterized by a high level of organization of society and division of labor. Every honey bee colony consists of a single queen, hundreds of male drones, thousands of female worker bees and numerous developing eggs, larvae and pupae. The queen is the only member of the colony who can lay fertilized eggs, capable of producing 2000 eggs per day. Workers are female honey bees that are unable to produce any fertilized eggs. They forage nectar and pollen, defend against attack and perform necessary tasks for the survival of the hive. Drones are male honey bees; whose purpose is to mate with the queen, soon after mating they dies. Since bees can’t talk, they perform dances to communicate important messages. “Waggle dance” is performed by a worker bee back at the hive to tell other bees about where to find food sources. The dance shows the direction of flowers relative to the sun, and bees automatically adjust their dances according to changing position of the sun. Speed of the dance indicates how far nectar is from the hive.

Inspired by the foraging behavior of honey bee colonies, Karaboga proposed the ABC algorithm [129]-[133] to solve multimodal, multidimensional problems. The foraging behavior of honey bees and a schematic diagram for the ABC algorithm is shown in Figure 2.1. Unlike other optimization algorithms, the ABC algorithm does not need any parameter tuning. The ABC algorithm finds the best solution in a search space like worker bees in bee hives searching for food sources with the highest amount of nectar. In contrast to other heuristic search algorithms, the ABC algorithm showed superior performance and has several advantages: strong robustness, fast convergence, high flexibility and fewer control parameters. Search strategy of the ABC algorithm is like the standard DE algorithm; however, it has a decision making mechanism that decides which areas within the search space is required to be surveyed in detail. This strategy discovers new high quality nectar sources within a search space while preserving existing good quality solutions.

The ABC contains three groups: scouts, onlooker bees, and employed bees [129]-[133]. A bee carrying out the random search is a scout. A bee going to the food source which has been visited previously is called an employee bee. A bee waiting in the dance area is called an onlooker bee. The number of food sources is equal to the number of employed or onlooker bees. A solution which cannot be improved after several predetermined trials becomes a scout bee and is abandoned. The best food source indicates a promising solution to an optimization problem and a fitness function is used to evaluate the quality of the solution obtained.

The main phases in the original ABC algorithm is as described below:

### 2.1.1 Initialization Phase

In initialization phase of the ABC algorithm, initial food locations are generated as a uniform random distribution, total number of food locations $SN$ is equal to the number of employed bees or onlooker bees,

$$\boldsymbol{x} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \dots, \boldsymbol{x}_{SN-1}, \boldsymbol{x}_{SN}\} \tag{2.1}$$

where $\boldsymbol{x_i}$ for $i = 1$ to $SN$ is a $1 \times D$ vector generated as,

$$x_{ij} = c_j^{[L]} + \alpha\left(c_j^{[U]} - c_j^{[L]}\right) \tag{2.2}$$

where α is a random number in $[0,1]$, $j \in \{1,2, \dots \dots \dots . D\}$, $c_j^{[U]}$ and $c_j^{[L]}$ are the upper and lower limits of $j$th dimension. Each food location is associated with an employed bee which exploits current location to find a better food location in its neighborhood.

### 2.1.2 Employed Bee Phase

In employed bee phase, bees search iteratively for food within a population. An employed bee first searches for foods in the adjacent region of its current food source, a new food location $\boldsymbol{v}_i$ is calculated by,

$$v_{ij} = x_{ij} + \phi_{ij}\left(x_{ij} - x_{kj}\right) \tag{2.3}$$

where $j$ is a randomly selected parameter index; $\boldsymbol{x}_k$ is a randomly selected food source; $\phi_{ij}$ is a random number within the range [-1,1]. If the fitness value of new food source is better than current one, then current food source is replaced by new food source. The fitness value is calculated using equation,

$$fit_i(\boldsymbol{x}_i) = \begin{pmatrix} \dfrac{1}{1 + f_i(\boldsymbol{x}_i)} \text{ for } f_i(\boldsymbol{x}_i) > 0 \\ 1 + |f_i(\boldsymbol{x}_i)| \text{ for } f_i(\boldsymbol{x}_i) \leq 0 \end{pmatrix} \tag{2.4}$$

where $f_i(\boldsymbol{x}_i)$ is the objective function value at $\boldsymbol{x}_i$.

### 2.1.3 Onlooker Bee Phase

When employed bees complete their food search, they pass the information to onlooker bees which in turn choose their food sources depending on the probability value calculated using,

$$p_{i=}\frac{fit_i(\boldsymbol{x}_i)}{\sum_{m=1}^{SN} fit_i(\boldsymbol{x}_i)} \tag{2.5}$$

where $fit_i(\boldsymbol{x}_i)$ is the fitness value associated with food location and is given by equation 2.4. Solutions with a higher fitness value has a greater probability of being chosen by onlooker bees.

### 2.1.4 Scout Bee Phase

In scout bee phase, a food source that is not improved after several trials, will be changed to a scout bee. Scout bees will randomly search for a new solution according to equation 2.2.

Numerical comparison of the ABC algorithm with other swarm-based algorithms [131]-[133], indicate that former can produce better results with benefit of fewer control parameters. A review of the ABC algorithm can be found in [134]. As the ABC algorithm is free from parameter tuning it is used widely in a variety of practical applications.

Like other evolutionary algorithms, the ABC algorithm also faces shortcomings such as getting trapped in local minima, and slower convergence speed. The above two problems are a result of insufficient balance between exploration and exploitation capability of search equation in the ABC algorithm. The solution generation equation which produces new food source based on the information of the previous solution, is good at exploration but poor in exploitation. Accelerating convergence speed and avoiding local optima are two most important goals in the ABC research.

There are many updates applied to the ABC algorithm in recent years to improve its speed, convergence rate and diversity in population. Main challenge in improving the performance of optimization problems is to find the right balance between exploration and exploitation. To address this concerning issue, numerous ABC variants have been developed. These improvements can be divided into two types, primarily new solution search equations have been introduced and secondarily, the original ABC is hybridized with other techniques. Some of the most popular modifications of the ABC algorithm are described below.

The ABC algorithm in its original version lacks a mechanism to deal with constrained optimization problems. Hence, a number of modifications have been applied to the original ABC algorithm to improve its performance for specific constrained engineering application problems. For constraint handling, the ABC algorithm can be combined with the Deb's rule and a probabilistic selection scheme to determine the optimum solution in the feasible region of a search space depending on a violation index value [135]. The first modification is made in the solution generation equation in employed and onlooker bee phase by changing more than one parameter in each iteration. In the second modification, greedy selection in the ABC algorithm is replaced by the Deb's selection mechanism which assigns probability value to solutions based on their fitness value. The probability value for each solution is generated according to the following equation,

$$p_i = \begin{cases} 0.5 + \left( \dfrac{fit_i(\boldsymbol{x}_i)}{\sum_{j=1}^{sn} fit_j(\boldsymbol{x}_j)} \right) \times 0.5 & \text{if solution is feasible} \\ \left( 1 - \dfrac{violation_i}{\sum_{j=1}^{sn} violation_j} \right) \times 0.5 & \text{otherwise} \end{cases} \tag{2.6}$$

where $violation_i$, is the penalty value of the solution $\boldsymbol{x}_i$ and $fit_i(\boldsymbol{x}_i)$ is the fitness value of the solution $\boldsymbol{x}_i$. Probability values of infeasible solutions are between 0 and 0.5 while those of feasible ones are between 0.5 and 1. By a selection mechanism like roulette wheel, solutions are selected probabilistically proportional to their fitness values of feasible solutions and inversely proportional to their violation values of infeasible solutions.

Zhu and Kwong [136] proposed the gbest-guided ABC algorithm by biasing the search towards the best solution found so far. The employed and onlooker bee phase equation 2.2 can be modified as,

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) + \varphi_{ij}(x_{gbest,j} - x_{ij}) \quad i \neq k \tag{2.7}$$

where $x_{gbest,j}$ is a randomly selected parameter index of global best solution, $\boldsymbol{x}_k$ is randomly selected food source, $\phi_{ij}$ is a random number within the range [-1,1], $\varphi_{ij} \in [0, C]$ where C is a constant. The value of C is determined through the trial and error method by applying it on different benchmark functions.

Diwold *et al.* [137] proposed a variation to the gbest-guided ABC algorithm, where a random neighbor selection is controlled through the following equation. Let $d(\boldsymbol{x}_i, \boldsymbol{x}_k)$ be the Euclidean distance between two solutions $\boldsymbol{x}_i$ and $\boldsymbol{x}_k$, the solution $\boldsymbol{x}_k$ will be chosen with probability defined by,

$$p_k = \frac{\frac{1}{d(\boldsymbol{x}_i, \boldsymbol{x}_k)}}{\sum_{l=1, l\neq i}^{SN} \frac{1}{d(\boldsymbol{x}_i, \boldsymbol{x}_l)}} \tag{2.8}$$

The closer solution has greater probability of being selected. The idea behind this modification is that it is more probable to find a better solution by mutating two good solutions close to each other in a solution space.

The best-so-far selection ABC algorithm [138] exploits the best solution found so far to improve convergence speed of the original ABC algorithm. The employed bee phase is unaltered as in equation 2.2, while onlooker bee phase is changed as follows,

$$v_{ij} = x_{ij} + fit_{gbest}\left(\phi_{ij}(x_{ij} - x_{gbest,j})\right) \quad i \neq k, d = 1,2 \ldots \ldots D \tag{2.9}$$

where $j$ is a randomly selected dimension, and $fit_{gbest}$ is the fitness value of the best solution found so far. Position update is applied to all dimension in onlooker bee phase, thus routing candidate solution towards the best solution so far. In scout bee phase, instead of choosing a new random solution, perturbation is added to current solution randomly as in equation below,

$$x_{ij} = x_{ij} + x_{ij} * \phi_{ij}\left(w_{max} - \frac{iter}{iter_{max}}(w_{max} - w_{min})\right) \quad (2.10)$$

where $j \in [1, D]$, $w_{max}, w_{min}$ are control parameters to determine the strength of perturbation and is fixed as 1 and 0.2 respectively, $iter$ is current iteration number and $iter_{max}$ is maximum number of iterations. As per the above equation, as number of iterations increases, algorithm is more exploitative than explorative.

Alatas [139] proposed two new chaotic ABC algorithm by using seven different chaotic maps as random number generators to improve convergence characteristics and to prevent the ABC algorithm from getting stuck at local solutions. In the chaotic ABC 1, instead of using uniform random distribution for population initialization it uses a chaotic map to generate solutions. In the chaotic ABC 2, if a solution cannot be improved after $\frac{limit}{2}$ trials, algorithm starts chaotic search for $\frac{limit}{2}$ trials around current solution by modifying the dimension and accepts new solution if it improves the current one. By combing modifications of the chaotic ABC 1 and the chaotic ABC 2, another variant of the ABC algorithm called the chaotic ABC 3 is proposed.

In an improved ABC algorithm [140], the population is initialized using chaotic random generator-based on the logistic map. After generating $SN$ solutions randomly, a new set of $SN$ solutions are generated by opposition-based population initialization, in which each variable is mirrored at the center of search range. From $2 * SN$ solutions the best $SN$ solutions are kept. Also, it modifies the search mechanism in onlooker and employed bee phases by incorporating differential evolution-based search. It uses following two equations inspired by the DE/best/1 and the DE/rand/1 scheme respectively,

In the ABC/best/1, equation 2.3 is modified as,

$$v_{ij} = x_{gbest,j} + \phi_{ij}(x_{ij} - x_{r1,j}) \quad (2.11)$$

In the ABC/rand/1, equation 2.3 is modified as,

$$v_{ij} = x_{r1,j} + \phi_{ij}(x_{ij} - x_{r2,j}) \tag{2.12}$$

where $r1$ and $r2$ are two random indices different from $i, x_{gbest,j}$ is the $j$ dimension randomly chosen from the best solution found so far. In the above equations, equation 2.11 biases a search towards the best solution in the current population which can improve convergence speed but may lead to a premature convergence, whereas equation 2.12 utilizes explorative property to prevent premature convergence. The above two equations are combined to find a new hybridized ABC search mechanism in which, a selective probability $p$ is used to control the frequency of introducing the ABC/best/1 and the ABC/rand/1. The value of selective probability $p$ is set as $0.25$.

Inspired by the differential evolution, Gao. *et al.* [140]-[141] proposed the global best ABC algorithm in which a solution search in employed and onlooker looker bee phases is directed towards the best solution of the previous iteration. Initial population is generated using a chaotic system and opposition-based learning which possesses ergodicity, randomness and irregularity to generate initial populations. Sinusoidal iterator is selected, and its equation is defined as follows,

$$ch_{k+1} = \sin(\pi ch_k) \quad ch_k \epsilon\ (0,1)\,, k = 0,1,2 \ldots \ldots \ldots \ldots \ldots \ldots . K \tag{2.13}$$

Based on variants from the differential evolution, the employed and onlooker bee phases are modified in the ABC/best/1 as follows,

$$v_{ij} = x_{best,j} + \phi_{ij}(x_{r1,j} - x_{r1,j}) \tag{2.14}$$

and in the ABC/best/2 as follows,

$$v_{ij} = x_{best,j} + \phi_{ij}(x_{r1,j} - x_{r2,j}) + \phi_{ij}(x_{r3,j} - x_{r4,j}) \tag{2.15}$$

where $r_1, r_2, r_3$ and $r_4$ are mutually exclusive integers randomly chosen from $\{1,2, \ldots \ldots SN\}$ and different from base index $i$, $\boldsymbol{x}_{best}$ is the individual vector with best fitness in current population and $j \in \{1,2, \ldots \ldots \ldots \ldots . D\}$ is randomly chosen indices and $\phi_{ij}$ is a random number in range $[-1,1]$. In equation 2.3, the coefficient $\phi_{ij}$ is a uniform random number in [−1, 1] and $x_{kj}$ is a random individual in population, and therefore, a solution search dominated by equation 2.3 is random enough for exploration. However, according to equation 2.14 or equation 2.15, ABC/best can drive a new candidate solution around the best solution of the previous iteration. Therefore, modified solution search equation described by above equation can increase exploitation of the ABC algorithm.

In the Rosenbrock ABC algorithm [142], optimization is carried out in two phases, during exploration phase, the ABC algorithm locates regions of attraction and during exploitation phase, it uses the adaptive Rosenbrock's rotational direction method to carry out a local search near the best solution.

In the first modification, fitness function is modified as follows,

$$fit_i = 2 - SP + \frac{2(SP-1)(r_i-1)}{SN-1} \tag{2.16}$$

where $SP \in [1.0,2.0]$ is a parameter called selection pressure, $r_i$ is rank of solution of $\boldsymbol{x}_i$ in the population. In the second modification, for every $n_c$ cycles of the ABC, a local search technique, the Rosenbrock's rotational direction method, is initiated with the global best solution as a starting point. An adaptive step size $\delta$, is defined as a fraction of average distance between selected solutions and the best solution achieved so far and is determined by following equation,

$$\delta_j = 0.1\frac{\sum_{i=1}^{m}\left(x'_{ij} - x_{best,j}\right)}{m} \tag{2.17}$$

where $\delta_j$ is step size of $j$th dimension, $m$ is the first 10% of solutions used to calculate step size, $\boldsymbol{x}'_i$ is the $i$th solution after ranking, $\boldsymbol{x}_{best}$ is the current best solution. A better solution obtained from the Rosenbrock's rotational directional search is used to replace a middle positioned solution in the population space.

In the incremental ABC [143] with local search method, a new population is added to the initial population after every $g$ iterations, biased by the members of a current population. The employed and onlooker bee phases are modified as follows,

$$v_{ij} = x_{ij} + \phi_{ij}\left(x_{gbest,j} - x_{ij}\right) \tag{2.18}$$

The scout bee phase is initialized by biasing towards the global best solution as,

$$x_{ij} = x_{gbest,j} + R_{factor}\left(x_{gbest,j} - x_{new,j}\right) \tag{2.19}$$

where $x_{ij}$ is a new solution replacing an abandoned solution and $R_{factor}$ is the bias towards the global best solution. When population size is increased, equation 2.3 of the employed bee phase is replaced as follows,

$$v_{new,j} = x_{new,j} + \phi_{ij}\left(x_{gbest,j} - x_{new,j}\right) \tag{2.20}$$

where $x_{new,j}$ is generated using equation 2.3, $v_{newj}$ is the $j$th coordinate biased by the global best solution, and $\phi_{ij}$ is a random parameter.

In the orthogonal learning-based ABC algorithm [144], a new search equation is used in the employed and onlooker bee phases as,

$$v_{ij} = x_{r1,j} + \phi_{ij}\left(x_{r1,j} - x_{r2,j}\right) \tag{2.21}$$

where $r_1$, and $r_2$ are mutually exclusive integers randomly chosen from $SN$ food locations and different from base index $i$, and $\phi_{ij}$ is the random number in the range $[-1,1]$. In equation 2.21, the vectors for generating candidate solutions are selected from the population randomly. Consequently, it has no bias in any search direction. As it is guided of only one term $\phi_{ij}\left(x_{r1,j} - x_{r2,j}\right)$, it can easily avoid oscillation phenomenon and maximize the search ability of the algorithm. A new candidate solution is generated around a randomly selected solution $\boldsymbol{x}_{r1}$, hence it can bring more information to the search equation and produces a more promising candidate solution. The orthogonal learning (OL) strategy-based algorithm (OCABC), combines the ABC algorithm with OL, the transmission vector $\boldsymbol{T}_i$ is generated as follows,

$$\boldsymbol{T}_i = \boldsymbol{x}_k + rand(0,1)(\boldsymbol{x}_{best} - \boldsymbol{x}_k) \tag{2.22}$$

$\boldsymbol{T}_i$ and $\boldsymbol{x}_i$ are mixed by making use of the orthogonal learning strategy to obtain a new solution $\boldsymbol{v}_s$, and a greedy selection is applied to select the best solution for the next generation.

In the Gaussian-based ABC algorithm [145], the food search equations in the employed and onlooker bee phases are updated as follows,

$$\begin{aligned} x_{ij}^{new} &= x_{ij}^{old} + \Delta.2(r_1 - 0.5)\beta\alpha \qquad if\ r_2 > p \\ x_{ij}^{new} &= x_{ij}^{old} + \Delta.2(r_1 - 0.5)2\alpha \qquad if\ r_2 \leq p \end{aligned} \tag{2.23}$$

where $r_1, r_2 \in [0,1]$ are random numbers from uniform distribution,

$$\begin{gathered} \Delta = x_{ij}^{old} - x_{r_1 j}^{old} \\ \beta = |s| \\ \alpha = 0.5 - 0.25\frac{iter}{maxiter} \end{gathered} \tag{2.24}$$

where $s$ is a random number extracted from a gaussian (normal) distribution and $iter$ and $maxiter$ indicate current iteration and maximum iteration number respectively; $p$ is responsible for a balance between gaussian and uniform distribution, and smaller values of $p$ seem preferable, indicating a superiority of the gaussian distribution. This method improves the performance of the ABC algorithm through a better balance between exploration and exploitation of the search space.

In the ABC algorithm-based on information learning [146], at each generation, the whole population is divided into several subpopulations by clustering partition and size of each subpopulation is dynamically adjusted based on the last search experience. Furthermore, two search mechanisms are designed to facilitate an exchange of information in each subpopulation and between different subpopulations. In the employed bee phase, the search equation is updated with the $lbest$ individual as,

$$t_{ij} = \frac{F_{r_{lbest}}.x_{lbest,j} + F_{r_{k1}}.x_{k1,j}}{F_{r_{lbest}} + F_{r_{k1}}} \tag{2.25}$$
$$v_{ij} = t_{ij} + \emptyset_{ij}(t_{ij} - x_{k2,j})$$

In the onlooker bee phase, the search equation is updated with the $gbest$ individual as,

$$t_{ij} = \frac{F_{r_{gbest}}.x_{gbest,j} + F_{r_{k1}}.x_{k1,j}}{F_{r_{gbest}} + F_{r_{k1}}} \tag{2.26}$$
$$v_{ij} = t_{ij} + \emptyset_{ij}\left(t_{ij} - x_{k2,j}\right)$$

where $j \epsilon \{1, \dots .D\}$ and $\emptyset_{ij}$ is a random number in $[-1,1]$; $k1$ and $k2$ are randomly selected indices in $[1, SN]$ such that $k1 \neq k2 \neq i$; $lbest$ and $gbest$ are the indices of the best individuals found by corresponding subpopulation and whole population respectively; $\boldsymbol{T_i}$ is a transmission vector; $Fr_i$ is fitness ranking of the $i$th individual in the current whole population from worst to best. Introduction of information about $lbest$ and $gbest$ in the search equations can guide the search towards promising regions and speed up convergence. Thus, the two search mechanisms have stronger exploitation than the original ABC. On the other hand, the information of a randomly selected individual $\boldsymbol{x}_{k1}$ in the neighborhood is inserted into the transmission vector which can maintain population diversity and escape from trapping into local optimum.

Since the introduction of the ABC algorithm, many ABC variants have been proposed for numerical optimization problems. Apart from introduction of new solution search equations [136]-[146], another common theme has been the hybridization of ABC algorithms with procedures taken from other techniques. By combing local search algorithms, the Nelder-mead algorithm (NMA) and the random walk with direction technique, Fister *et al*. [147] proposed the memetic ABC algorithm. The ABC algorithm is also hybridized with the genetic algorithm [148]-[149] and the particle swarm optimization [150]-[153]. Inspired by foraging behavior of bacteria, Zhong *et al.* [154], introduced a local search technique in solution update equations of employed and onlooker bee phase. The differential evolution [155] and a chaotic operator [156] have been introduced into the search equation of the original ABC algorithm to improve its convergence speed. For improving movement of a scout bee, mechanisms based on nonlinear interpolated path and gaussian movement is proposed by Sharma and Pant [157]. To improve solution search equation, Rajasekhar *et al.* [158] introduced levy probability distributions. The improved artificial bee colony algorithm [159] with a new search cycle operator is used to solve higher dimensional multimodal problems. Tsai *et al*. [160] proposed the interactive ABC algorithm in which a gravitational force is used to drive the bee movement in onlooker phase. The hybrid simplex ABC algorithm was proposed by Kang *et al*. [161]by integrating the Nelder-mead simplex method into the ABC algorithm. A hybrid swarm intelligent approach based on the genetic algorithm and the ABC algorithm was proposed by Zhao *et al*. [162], which combines the parallel computation merit of the genetic algorithm with the self-improving ability of the ABC Algorithm by exchanging information between bee colony and genetic algorithm population. An improved quantum EA was proposed by Duan *et al*. [163] which uses the ABC algorithm to improve the local search ability and escape from trapping into the local optima. The ABC algorithm have also been hybridized with the Hooke-jeeves pattern search [164] and the Powell's method [165] to improve its exploitation capability.

It has been clear that these modified ABC algorithms can improve performance of the ABC algorithm to some extent. However, it is impossible for a method to outperform all other algorithms on every problem. For example, some approaches utilize information about the best solution to speed up convergence on unimodal functions but gets trapped into local optima on multimodal functions.

## *2.2 Challenges Faced by Variants of ABC Algorithm in Digital Filter Design*

The modifications to the original ABC algorithm, in general, improves exploitation capability of the search equation but various challenges are faced when they are applied to digital filter design. Some of them are listed in Figure 2.2.

### *2.2.1 Strong Impact of Local Search and Directed Search*

A local search method can improve the performance of continuous optimization in many population-based metaheuristics. Generally, a local search is applied with an initial point as the global best solution and finds whether a local search can replace the best solution obtained so far. Trade-off between local search and global search is essential for successful optimization, because if the effect of local search is too strong then the ABC algorithm-based optimization is insignificant. Also, in many variants of the ABC algorithm, exploitation of search equation is improved by directing the search towards the best solution obtained so far, which increases the chance of getting stuck at a local optimum and decreases the quality of solutions for higher dimensional problems.

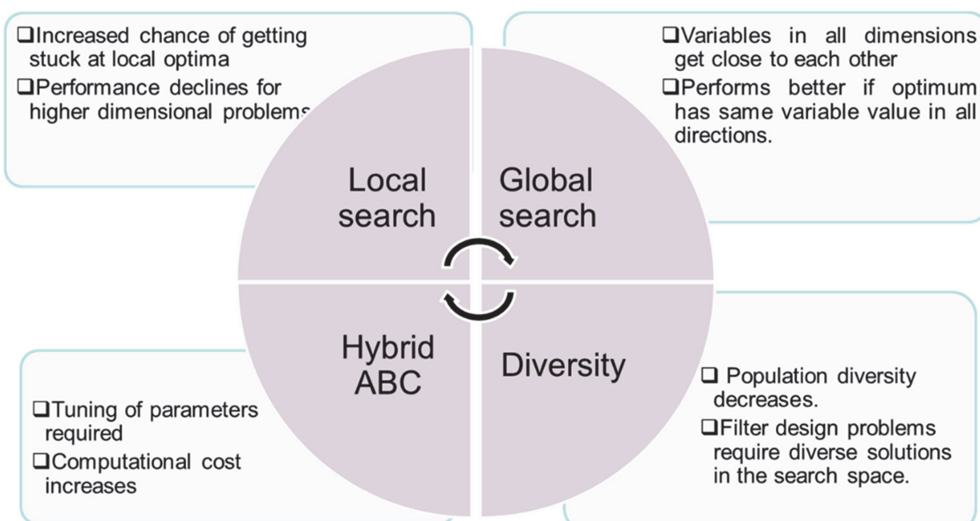


**Figure 2.2** Challenges Faced by Variants of ABC Algorithm in Digital Filter Design

### 2.2.2 *Issues in Obtaining a Global Optimum Solution*

Directing the search towards the best solutions in the solution space makes variables in all dimensions comes closer. These types of algorithms can be used to solve problems which has same optimum value in all dimensions like $f_{opt}(\boldsymbol{x}) = [0,0, \dots \dots \dots \dots 0]$. In some variants of the ABC algorithm, their solution update equations are applied to all dimensions [138], instead of a single dimension in every iteration. In such cases the variables in each dimension will have optimum value in different directions, guiding the search towards the best solution will restrict the search in the multi directions, which increases the error value and, deteriorates the optimization performance.

### 2.2.3 *Hybrid ABC*

In addition to incorporating the ABC algorithm with local search heuristics [147], the search equation can be updated by incorporating other evolutionary algorithm such as the GA [148]-[149], the PSO [150]-[151] into the ABC algorithm. This will improve the performance in some benchmark functions, but it requires tuning of the control parameters of each of these evolutionary algorithms, which in turn increases the computational cost.

### 2.2.4 *Diversity of Search Space*

Directing a search towards the best solution will increase exploitation but limits the diversity of a search space. When the search is biased towards the neighborhood of best solution, it limits exploration of unknown regions in a search space and restricts the diversity of a population.

## 2.3 *Multiobjective Optimization*

Many multiobjective algorithms have been developed in the past decade to deal with optimization problems involving more than one objective. In general, a multiobjective optimization problem can be stated [166] as,

$$\text{Minimize or Maximize} \quad f_m(\boldsymbol{x}), \qquad m = 1,2, \dots, M \tag{2.27}$$

$$\text{subject to} \begin{cases} g_j(\boldsymbol{x}) \geq 0, & j = 1,2, \ldots., J; \\ h_k(\boldsymbol{x}) = 0 & k = 1,2, \ldots, K; \\ x_i^{(L)} \leq x_i \leq x_i^{(U)} & i = 1,2 \ldots, n; \end{cases}$$

where $f_m(\boldsymbol{x}) = \left(f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), f_3(\boldsymbol{x}), \ldots., f_M(\boldsymbol{x})\right)^T$ represents the $M$ objective functions that need to be minimized or maximized, $g_j(\boldsymbol{x})$ represents $J$ inequality constraints and $h_k(\boldsymbol{x})$ represents $K$ equality constraints, ($x_i^{(L)}$, $x_i^{(U)}$) represents lower bound and upper bound of variables. The solution $\boldsymbol{x}$ is a vector of $n$ decision variables $\boldsymbol{x} = [x_1, x_2, x_3, \ldots \ldots., x_n]^T$.

### *2.3.1 Concepts and Definitions*

Some of the basic concepts and definitions that are generally used in multiobjective optimization are disused below:

A solution $\boldsymbol{x}^{(1)}$ is said to dominate the other solution $\boldsymbol{x}^{(2)}$, if both conditions 1 and 2 are true:

1. The solution $\boldsymbol{x}^{(1)}$ is no worse than $\boldsymbol{x}^{(2)}$ in all objectives, or $f_j\left(\boldsymbol{x}^{(1)}\right) \ngtr f_j\left(\boldsymbol{x}^{(2)}\right) \;\; \forall j \in \{1,2 \ldots, M\}$
2. The solution $\boldsymbol{x}^{(1)}$ is strictly better than $\boldsymbol{x}^{(2)}$ in at least one objective or $f_j\left(\boldsymbol{x}^{(1)}\right) < f_j\left(\boldsymbol{x}^{(2)}\right)$ for at least one $j \in \{1,2 \ldots, M\}$.

where $f_j\left(\boldsymbol{x}^{(i)}\right)$ is the $j$th objective function value of solution $\boldsymbol{x}^{(i)}$ to be minimized.

If for every member $\boldsymbol{x}$ in a set $P$, there exists no solution $\boldsymbol{y}$ dominating any member of the set $P$, then solutions belonging to the set $P$ constitute a locally Pareto optimal set. The set of all Pareto optimal solutions is the Pareto set (PS) and the image of PS in the objective space is called Pareto front (PF).

A solution $\boldsymbol{x}^{(1)}$ strongly dominates a solution $\boldsymbol{x}^{(2)}$ (or $\boldsymbol{x}^{(1)} < \boldsymbol{x}^{(2)}$), if solution $\boldsymbol{x}^{(1)}$ is strictly better then solution $\boldsymbol{x}^{(2)}$ in all $M$ objectives. Among a set of Pareto front solution $P$, the weakly nondominated set of solutions $P'$ are those that are not strongly dominated by other member of the set $P$.

Multiobjective optimization generally falls into two categories: classical optimization methods and evolutionary-based optimization methods. Classical methods use a deterministic procedure, which starts from a random initial solution, new solutions are generated by searching in new directions based on some transition rule. Unidirectional search is performed in each search direction until stopping criteria is reached. The search direction is determined using direct methods or gradient-based methods; direct method uses objective function values and constraint values directly to determine the search direction whereas gradient-based methods use first and/or second-order derivatives of objective function/constraint values to control search strategy. The classical methods face a plethora of difficulties, some of them are: dependence on initial solution; trapped at local optimum; inability to handle problems in discrete search space; and inefficiency on parallel machine. While dealing with non-linear, complex real-world optimization problems, classical methods are often stuck at local optimum. In addition, classical methods are problem specific, as seen with conjugate gradient methods which can effectively solve quadratic objective functions but incapable of solving problems with multi optimal solutions.

### *2.3.2 Multiobjective Evolutionary Algorithms*

Evolutionary algorithms can alleviate above difficulties of classical methods, and hence can be applied to multiobjective optimization problems. Due to metaheuristic-based nature, evolutionary algorithms can approximate Pareto front of multiobjective problems. These are known as the multiobjective evolutionary algorithm (MOEA). Some of the popular MOEAs are described below.

In the pareto archived evolution strategy (PAES), Knowles and Corne [167]-[168] suggested a simple MOEA using a single parent single offspring evolution strategy. The initial offspring population $Q_0$ is generated from the parent population $P_0$ randomly. In each generation, an offspring is compared with respect to a parent and nondominated solutions are obtained. If an offspring dominates a parent, the former is accepted as the next parent and iteration continues. On the other hand, if a parent dominates an offspring, it is discarded, and a new mutated solution is generated. However, if an offspring and a parent do not dominate each other, the choice between them is made by comparing with an archive of best solutions found so far. The offspring is compared with the archive to check if it dominates any member of the archive. If it does, the offspring is accepted as the new parent and all dominated solutions are eliminated from the archive. If the offspring does not dominate any member of the archive, both the parent and the offspring are checked for their nearness with the solutions of the archive. If the offspring resides in a least crowded region of the objective space among the members of archive, it is accepted as a parent and a copy is added to archive. The maximum size of the archive is specified by the user initially. Crowding region is maintained by dividing the entire search space deterministically into $d^n$ subspaces, where $d$ is the depth parameter and $n$ is the number of decision variables, and by updating subspaces dynamically. Even though the PAES algorithm has an advantage of diversity maintenance without any sharing parameter, it is highly sensitive to the depth parameter and the number of decision variables.

In the strength Pareto evolutionary algorithm2 (SPEA2) [169]-[170], elitism is introduced to improve convergence properties by including the elite individuals in a gene pool. Also, in the SPEA2, since the archive size is predefined, when the number of nondominated solutions is less than the predefined archive size, all empty spaces in the archive are filled up by dominated individuals. If the archive size exceeds the predefined limit, a truncation method is used to eliminate excess nondominated solutions while preserving the boundary solutions. The truncate function iteratively removes one solution at a time until the predefined limit is reached. The initial population $P_0$ is generated randomly and its nondominated members are filled into the initial archive $Q_0$. Then the offspring population is generated from the archived solutions. The fitness value is assigned to an offspring after considering both population and archive. Each solution $\boldsymbol{p}$, from the population $P$ and the archive $Q$, is assigned a strength value $s_p$ representing the number of individuals dominated by $\boldsymbol{q}$,

$$s_p = |\boldsymbol{q}|\boldsymbol{q} \in P \cup Q \wedge \boldsymbol{p} \leq \boldsymbol{q}| \tag{2.28}$$

Raw fitness value of an individual is determined by the sum of strengths of its dominator solutions in both the archive and the population,

$$R_p = \sum_{\boldsymbol{q} \in P+Q, \boldsymbol{p} \le \boldsymbol{q}} s_p \tag{2.29}$$

High value of $R_p$ means that $\boldsymbol{p}$ is dominated by many individuals, and low value of $R_p$ corresponds to $\boldsymbol{p}$ being a nondominated solution. When most solutions nondominate with each other, density information is used to discriminate between individuals having identical raw fitness. The density estimation uses the $k^{th}$ nearest neighbor approach, where density at any point is a decreasing function of distance to the $k^{th}$ nearest points. The SPEA2 preserves boundary solutions by keeping the number of solutions in an external archive constant over time.

The nondominated sorting genetic algorithm I (NSGA-I) have many disadvantages including high computational complexities and a lack of elitism. The above problems can be eliminated using the NSGA II algorithm, which selects the best population by combining the properties of both the parent and the offspring population. The NSGA II [171] is the most prominent elitist multiobjective evolutionary algorithm. The general principle of the NSGA II is as follows: Initial offspring population $Q_0$ is generated by applying genetic operator to the parent population $P_0$. Nondominated sorting is applied after combining an offspring population and a parent population. In naive forecasting approach, the first nondominated front in a population is obtained by comparing each solution obtained to every other solution in the population. This process is repeated after excluding the solutions of previous front to obtain the second nondominated front and so on. This process continues until all population members are classified. New parent population $P_{t+1}$ is filled with individuals from the best nondominated front. Instead of discarding solutions that cannot be accommodated in the new parent population, a niching strategy is used to choose individuals from the last front which reside in the least crowded regions. Crowding distance, a measure to quantify the average distance of one solution to its two nearest neighbors in the same front in the NSGA maintains a diversity among the population members. Even though, the NSGA II is considered as one of the powerful MOEAs, giving diverse solutions and a better convergence near the true Pareto optimal front, it has certain disadvantages when cardinality of the first front exceeds the population size, then some of the Pareto optimal solutions will be replaced by the non Pareto optimal solutions based only on the crowding distance.

The ABC algorithm has been extended to multiobjective domain for solving various practical design problems. The grid-based multiobjective ABC algorithm [172] is inspired by the original ABC algorithm, in which fitness value of solutions generated is calculated using the equation as below,

$$fit(\boldsymbol{x_i}) = \frac{dom(\boldsymbol{x}_i)}{FoodNumber} \tag{2.30}$$

where $dom(\boldsymbol{x}_i)$ is the number of food sources dominated by the food source $\boldsymbol{x}_i$. Best solutions are archived to an external repository, and its size is maintained using the $\varepsilon$ dominance approach. In this approach, solution space is divided into boxes of size $\varepsilon$ and only one nondominated solution is selected from each box. If box contains more than one solution, a member close to left corner is retained. On reaching termination criteria, the external archive is updated as the Pareto optimal set.

The hybrid multiobjective ABC algorithm [173] has been applied for optimizing the copper strip production by simultaneously optimizing the cost of raw materials and the amount of raw materials thrown into furnace. The optimal Pareto front is generated using four important multiobjective approaches such as, fast nondominated sorting approach, crowded distance estimation, summation of normalized objective values and diversified selection. The objective function value of all the members in colony is normalized as follows,

$$f'_m(\boldsymbol{x}_m) = \frac{f(\boldsymbol{x}_m) - f_{min}}{f_{max} - f_{min}} \tag{2.31}$$

where $f'_m(\boldsymbol{x}_m)$ is the *m*th normalized objective value and are summed to obtain a single value. New solutions, generated in the employed and onlooker bee phases, are selected based on their nondomination ranks and crowding distances.

For design optimization of laminated composite components, the vector evaluated ABC algorithm [174] has been applied to simultaneously optimize the weight of layers, and the implementation cost. For $M$ objective functions, the entire population is divided into $M$ swarms each of size $n$ and each swarm is evaluated according to one objective function while information from other swarms determine its motion in the solution space. Each of the swarm updates towards the best solution of its respective objective function and finally converges to a global optimum solution. The best solution of one swarm is used to calculate the new position in another swarm. The position update equation for the $i$th individual of the $j$th swarm is as follows,

$$D_{i+1}^{[j]} = \alpha(r - S) + (1 - \beta)^{[j]} D_i + \gamma^{[k]} Dbest_i \tag{2.32}$$

where $\alpha$ is the randomness in amplitude of the bee, $\beta$ is the convergence rate, $\gamma$ is the learning rate, $r$ is the random number in the interval $[0,1]$, $S$ is the step size, $Dbest_i$ is the overall best value of fitness function and $k$ is the randomly selected neighbor swarm.

The nondominated sorting-based multiobjective ABC algorithm [175] maintains the best and diverse solutions in the Pareto front. The size of the external archive is maintained using nondominated sorting and crowding distance techniques. In the employed bee phase, if newly generated solution does not dominate old solution $X_{ij}$, an augmented population is generated as follows,

$$U_{ik} = X_{ik} + rand_{ik}(0,1) * \left(1 - \frac{iter}{MIC}\right) \tag{2.33}$$

where $k$ is a randomly chosen dimension, $iter$ is the current iteration number and $MIC$ is the maximum number of iterations.

## *2.4 Limitations of Multiobjective Evolutionary Algorithms*

Even though the state-of-the-art multiobjective evolutionary algorithms (MOEA) can effectively solve some multiobjective practical design problems, they face some drawbacks as described below:

### *2.4.1 Exponential Increase in Population Size*

As the number of objective increases, a large fraction of population becomes nondominated and in order to maintain diversity [176], the population size must increase exponentially. If the population size is not adequate, the solutions will be distant in the objective space. New solutions generated using the recombination of their parents will also be far away from each other, and thus reaching an optimal solution is difficult.

### *2.4.2 Difficult to Select a Single Optimum Solution*

As the number of objective increases, more points are required to represent the Pareto front. It is difficult for the decision maker to choose a single optimum solution from the Pareto front containing many optimal solutions.

### *2.4.3 Visualization is Difficult*

It is difficult to visualize the Pareto front of optimization problems with more than three objectives.

## *2.5 Preference-Based Multiobjective Evolutionary Algorithms*

Even though the above multiobjective evolutionary algorithms can result in an optimal set, from a practical point, a user needs only one solution. From multiple trade-off optimal solutions, one solution must be selected by the decision maker using higher level information. Figure 2.3 represents the Pareto front approximation of two objective functions $J_1(\theta)$ and $J_2(\theta)$; blue points represents the Pareto optimal solutions; box represents the decision maker's region of interest, and any solution outside the region of interest can be discarded.

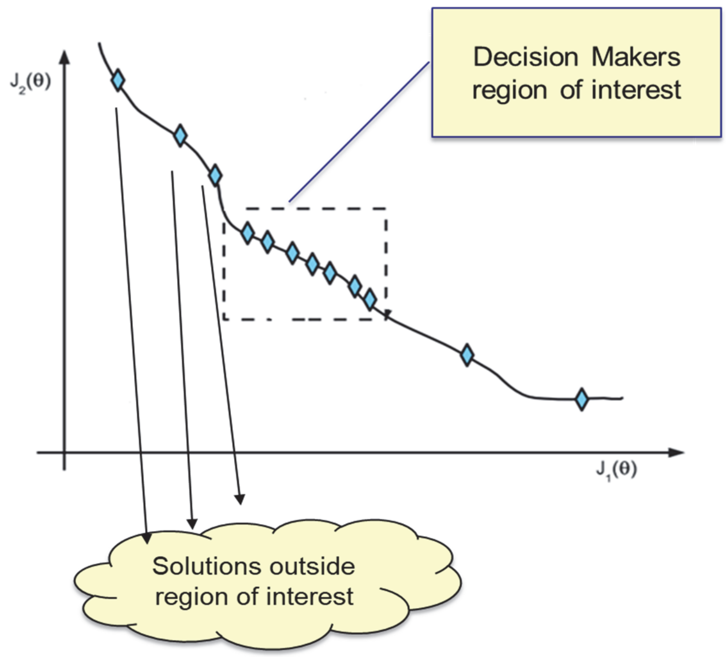


**Figure 2.3** Pareto Front Approximation

To find a preferred solution, some preference information is needed to guide the search towards the region of interest in the Pareto front. If relative preference factor among objectives are known for a specific problem, they can be used *a posteriori*, *a priori* or interactively in the optimization process to obtain feasible solutions in the region of interest. The non preference-based methods do not assign any importance to any of the objectives, *a posteriori* methods use the preference information of each objective after optimization process, *a priori* methods use information about the preferences of the objectives, that is already known, to find a preferred optimal solution during optimization. Interactive methods use preference information progressively during optimization process.

One of the earliest attempts in the preference-based MOO can be seen in [177], in those algorithms, preference information and Pareto dominance are used to find fitness and ranking of individuals in population pool. Fuzzy approach can be interactively used to set preferences [178], the decision maker can iteratively choose reference points to represent preferences until desired results are obtained. The reference point-based multiobjective optimization [176] can be used to set preference information in an objective space. Preference-based strategy is combined with the elitist nondominated sorting genetic algorithm to simultaneously find the preferred set of solutions near the reference point. Instead of finding a single optimum solution, this method finds a set of optimal or near optimal solutions near the desired region of the decision maker's interest. The goal programming technique is used in engineering design application to find a single solution that satisfies several design goals. Goal programming can be used to solve a multiobjective optimization problem by minimizing deviations from individual objectives/goals. In contrast to classical methods, this method eliminates the need of assigning individual weights to each objective. Goal programming can be incorporated into the NSGA [179] by converting each goal into an equivalent objective function of minimizing deviation from target. Even though this method can solve non-convex, multimodal optimization problems that classical methods cannot solve, it is not effective in finding the Pareto optimal solution.

Preference information can be incorporated into the NSGA-II [180] by using the biased crowding distance approach. Initially all the solutions are projected onto linear hyperplane and crowding distance values can be calculated as the ratio of the distances of neighboring solutions in the original objective space to those on the projected hyperplane. The preferred solutions with larger biased crowding distances are those which lie on the plane parallel to the selected hyperplane. This procedure requires a reference direction and a diversity control parameter for the optimization process to converge to an optimal solution.

The interactive evolutionary multiobjective optimization (IEMO) [181] can be used by the decision maker to set preferences interactively by incorporating classical decision making approaches to the multiobjective optimization procedure. This method has been used to generate solutions in region of interest in the Pareto front. In the interactive MOEA [182], objective function weights can be changed adaptively by the decision maker depending upon the locations of solutions in the current population. This can direct the search towards the region of interest in the Pareto front.

The physical-programming can be combined with *a priori* preferences [183], [189] to guide the search towards a region of interest in the Pareto front. The objective space is partitioned into several levels to represent various preferences that represents the decision maker's interests. Designer's higher level information is converted into preference functions that reflect the decision maker's interest and meaningful parameters are used for each objective. A single-objective function without weights is constructed to convert preferences to numbers. This method eliminates time consuming and trial and error procedure of weight selection, and instead it selects preference ranges which have the same units as each objective function. The preference from the decision maker can be accepted progressively [184] in each generation of multiobjective evolutionary algorithms like the NSGA-II to guide search towards the most preferred solution of choice. Preference information is used to design a monotone value function, which satisfies the decision maker's preferences, and the progressively interactive multiobjective optimization guides the direction of search towards a preferred solution.

Incorporating preferences into the multiobjective ABC algorithm is a less explored area in the field of evolutionary computation. More research in this direction is required to include the decision maker's preferences into the ABC algorithm.

## *2.6 Conclusions*

Bio-inspired algorithms are becoming a popular research topic as evolutionary computation is being applied to solve many real-world problems. Even though these algorithms can solve complex multimodal problems, they face many difficulties for designing digital filters. Compared to other algorithms, the ABC algorithm has a benefit of fewer control parameters but requires a long time for convergence. This chapter gives a brief survey on various modifications applied to the ABC algorithm and its limitations when applied to filter design problems. Furthermore, single-objective optimization is extended to multiobjective domain and prominent methods in multiobjective evolutionary algorithms have been described. Limitations involved with an increased number of objectives have also been explained. A review on preference-based multiobjective optimization and its effectiveness in reducing computation complexity has also been presented.

# CHAPTER 3

# LINEAR PHASE FIR DIFFERENTIATOR DESIGN

In this chapter a novel improvement is applied to the original ABC algorithm to improve its performance. The ABC algorithm with adaptive modification rate (called the ABC-AMR algorithm) is used to design Type 3 and Type 4 linear phase FIR differentiators. Minimax method is used to formulate the objective function. To analyze the performance of proposed improvement, results are compared with other variants of the ABC algorithm such as the gbest-guided ABC, the best-so-far selection ABC and the global best ABC; and the Parks-McClellan algorithm. Experimental result indicates that the proposed modification can reduce convergence time and minimax errors values.

This chapter is organized as follows: Section 3.1 gives an overview of linear phase FIR differentiator design, Section 3.2 presents the ABC-AMR algorithm based on an adaptive modification rate, Section 3.3 provides the minimax formulation of linear phase FIR filters, obtained results are discussed in Section 3.4 and conclusions are given in Section 3.5.

## *3.1 Introduction*

Differentiator design is an important filter design problem as it forms the building block of a diverse range of applications in biomedical engineering, communication systems, digital image processing and various other real-world scenarios. The differentiator computes the time derivative of any applied signal. The earliest approach for differentiator design includes minimax approach and eigenfilter method. Conventionally, digital differentiator filters are designed using the McClellan-Park algorithm [11], which can be extended to the design of higher order FIR differentiator [12] of any arbitrary length by applying a modification to the McClellan, Parks, and Rabiner algorithm, such that the Remez exchange method is combined with the minimum weighted Chebyshev error for obtaining optimal coefficients. This method results in large error value and fails to converge in the design of full band higher order differentiator designs. In [13], a simple and fast eigenfilter method is described to design higher order differentiators. Quadratic error function is minimized in frequency band and filter coefficients are obtained by computing eigen vector corresponding to the smallest eigen value of a positive definite symmetric matrix. In the least square design approach [14], quadratic error function is formulated by calculating absolute mean square error between the magnitude response of the ideal and approximating differentiator and filter coefficients are obtained by solving a system of linear equations. Analytical methods can be used in differentiator design, as it can simplify optimization procedure by using the matrix properties of trigonometrical functions. Simple analytic closed form relations for the least square design of higher order differentiators [18] can reduce computation time as it doesn't require to solve a system of linear equations. In another class of differentiator design, the transfer function of integrator is inverted to obtain the corresponding differentiator. An integrator transfer function can be obtained by using simple linear interpolation between magnitude responses of different Newton-cotes integrators such as rectangular, trapezoidal and Simpson integrators [19]. Digital integration techniques [20] like the Schneider kaneshige groutage, trapezoidal rule and the rectangular rule, can be interpolated and then modified to design differentiators. Conventional algorithms can efficiently solve unimodal problems but in multimodal problems, it has certain short comings such as: inability to solve discontinuous, non-differentiable error function, convergence to a sub optimal solution, unable to find global optimum in a large search space, and sensitivity to initial set points.

Due to the ability of evolutionary algorithms to solve multimodal, non-differentiable composite problems, it has been used in recent years for filter design applications. EA-based algorithms such as the ABC algorithm [15],the teaching -learning-based optimization [16] and the cuckoo search algorithm [17] have been applied to digital filter designs. Modified particle swarm optimization algorithm [21] have been used to optimize mean square error to design digital integrators and differentiators. The results obtained for second, third and fourth order differentiators by the proposed algorithm are either superior or at par with the basic PSO variants and hybrid techniques. Linear phase second order recursive integrators and differentiators can be designed using the genetic algorithm [22], and thus designed digital differentiators have linear phase response over the entire Nyquist frequency range including $\omega = 0$ radian. Wide band differentiators are designed using several optimization techniques such as simulated annealing, genetic algorithms, and Fletcher and Powell optimization [23]. And the work emphasizes on designing differentiators without relying on inverting integrators of similar order because inverting an optimized integrator does not necessarily produce an optimized differentiator. Even though these algorithms can design differentiator better than conventional methods, they require longer convergence time. As the filter length increases, in order to speed up the convergence, most of heuristic/metaheuristic evolutionary optimization algorithms need a good candidate solution at initialization. However, it is not always possible to obtain a good candidate solution prior to optimization process. Hence, in this chapter an improvement is proposed to the original ABC algorithm which eliminates the necessity of seeding initial population and linear phase Type 3 and Type 4 differentiators are designed using the ABC-AMR algorithm.

### *3.2 ABC Algorithm with Adaptive Modification Rate (ABC-AMR)*

Even though the original ABC algorithm is efficient in the optimization of multimodal and multidimensional basic functions, it has a poor performance on composite and non-separable functions. These limitations are due to an insufficient balance between exploration and exploitation capability of the search equations. Exploring new solutions in different regions of a search space is essential in the initial stages of optimization and in later stages, algorithm can apply knowledge about previous good solutions to obtain global optimum solution.

Most variants of the ABC algorithm reduce computation time by biasing a search towards the direction of the best solution on the assumption that the best solution obtained is optimum but since evolutionary algorithms are population-based random search method, there is a chance that an algorithm gets stuck at any local optimum solution. So instead of incorporating information about the best solution, a novel approach has been proposed, in which a control variable called adaptive modification rate ($AMR$) is used to determine the number of variables updated in each iteration.

In contrast to many evolutionary algorithms, where an initial population is seeded with a previously known good solution, in the proposed method, population is initialized using a uniform random distribution,

$$x_{ij} = c_j^{[L]} + \alpha * \left(c_j^{[U]} - c_j^{[L]}\right) \tag{3.1}$$

where $x_{ij}$ denotes $j$th dimension of $i$th solution, $c_j^L$ and $c_j^U$ represents maximum and minimum $j$th dimension of food source boundaries respectively and α denotes a random number in [0,1].

Once population is initialized, optimization begins with the employed bee phase, and in the original ABC algorithm, the update rate is fixed, producing a new solution, $v_{ij}$, by changing only one parameter of a parent solution $\boldsymbol{x}_i$ as described in equation 2.3. This results in a longer convergence time and does not guarantee an optimal solution in filter design problems. As seen from Chapter 2, most of the modifications applied to the ABC algorithm tries to reduce computational time by biasing a search towards the best solution which is beneficial in some applications but in the case of a composite, non-convex design problems obtaining a minimal error value is difficult. In higher order filter design problems, in order to assure a minimal error value within given design constraints, new mutated solutions in the solution space must be explored. Some research has been done in this direction [185], but with a constant modification rate ($MR$), resulting a limited solution space diversity. For each parameter $x_{ij}$, a uniformly distributed random number, $0 \leq R_{ij} \leq 1$, is produced and if the random number is less than $MR$ then the parameter $x_{ij}$ is modified as,

$$v_{ij} = \begin{cases} x_{ij} + \emptyset_{ij}\left(x_{ij} - x_{kj}\right), & \text{if } R_{ij} < MR \\ x_{ij}\,, & \text{otherwise} \end{cases} \tag{3.2}$$

where $k \in \{1,2, \dots \dots . SN\}$ is randomly chosen index that is different from $i$ and $MR$ is the constant modification rate which takes value between 0 and 1. A lower value of $MR$ may cause solutions to improve slowly while a higher value may cause too much diversity in the solution space.

For some problems, a higher value of $MR$ is appropriate while in others, a lower value is suitable. For this reason, an improvement is proposed such that its value is set to change adaptively during each search, called adaptive modification rate ($AMR$). This ensures exploration in the initial stages of optimization and the value of $AMR$ decreases towards the end concentrating more on exploitation. This improvement guarantees that inefficient sources are modified more while better ones are modified less often.

The solution update equation in onlooker and employed bee phase and adaptive modification rate ($AMR$) is determined according to the following equation,

$$v_{ij} = \begin{cases} x_{ij} + \emptyset_{ij}\left(x_{ij} - x_{kj}\right), & \text{if } R_{ij} < AMR \\ x_{ij}\,, & \text{otherwise} \end{cases}$$

$$AMR(t+1) = min\left\{1, \beta + \frac{\alpha}{\mu(t) + 0.01}\right\} \tag{3.3}$$

where $\alpha = 0.06$ and $\beta = 0.1$ and $\mu(t)$ is the ratio of number of successful mutations to total number of mutations in the population. The value of $\alpha$ and $\beta$ are set in accordance with filter design applications in single-objective and multiobjective domain.

The pseudocode for the ABC-AMR is illustrated in Table 3.1. In this pseudocode, adaptive modification rate $AMR$ is shown in line number 80, and the ratio of successful mutation to total number of mutation is calculated such that for every position update, if the corresponding objective function value is better than the global optimum of current population, it is counted as a successful mutation ($\mu_{sucessful}$) and otherwise as a unsuccessful mutation ($\mu_{failure}$).

Note that pseudocode in Table 3.1 is for minimization problem and for a maximization problem, lines 14, 29 and 53 must be changed from,

$$f\left(\boldsymbol{x}_{new}(*)\right) < f\left(\boldsymbol{x}_{old}(*)\right) \tag{3.4}$$

to

$$f\left(\boldsymbol{x}_{new}(*)\right) > f\left(\boldsymbol{x}_{old}(*)\right) \tag{3.5}$$

where $f\left(\boldsymbol{x}(*)\right)$ is the objective function value.

**Table 3.1** Pseudocode of ABC-AMR Algorithm

| Pseudocode | Comments |
|---|---|
| For $i = 1{:}SN$ | |
| For $j = 1{:}D$ | |
| $x_{ij} = c_j^{[L]} + \alpha * \left(c_j^{[U]} - c_j^{[L]}\right)$ | Randomly initialize food sources |
| Next $j$ | |
| $trial_i = 0$ | Set the limit counter as 0 |
| Next $i$ | |
| For $i = 1{:}SN$ | |
| If $i = 1$ | |
| For $j = 1{:}D$ | |
| $x_b(i,j) = x(i,j)$ | Find best solution in the population |
| Next $j$ | |
| $f\left(\boldsymbol{x}_b(i)\right) = f\left(\boldsymbol{x}(i)\right)$ | |
| Else | |
| If $f(\boldsymbol{x}_i) < f(\boldsymbol{x}_b)$ | |
| $\boldsymbol{x}_b = \boldsymbol{x}_i$ | |
| $f(\boldsymbol{x}_b) = f(\boldsymbol{x}_i)$ | Find the global minimum of the current population |
| Next $i$ | |
| Initialize $AMR$ | Initialize parameters for $AMR$ |
| While $iter \leq MaxIter$ | |
| $\mu_{sucessful} = 0$, $\mu_{failure} = 0$ | Initialize values |
| For $i = 1{:}SN$ | For each employed bee |
| For $j = 1{:}D$ | For each dimension |
| Select $x_{kj}$ | Randomly select neighboring solution |
| $v_{ij} = \begin{cases} x_{ij} + \emptyset_{ij}\left(x_{ij} - x_{kj}\right), & \text{if } R_{ij} < \text{AMR} \\ x_{ij}, & \text{otherwise} \end{cases}$ where $\emptyset_{ij} = \text{rand}[-1,1]$ | Update the position of employed bees |
| Next $j$ | |

| | | |
|---|---|---|
| 26 | Select $x_{kj}$ | If no parameter is changed randomly select a $d$ dimension of the solution. |
| 27 | $v_{id} = x_{id} + \emptyset_{id}(x_{id} - x_{kd})$ | Change at least one parameter |
| 28 | Calculate $f(\boldsymbol{v}_i)$ | Evaluate the quality of solution $\boldsymbol{v}_i$ |
| 29 | If $f(\boldsymbol{v}_i) < f(\boldsymbol{x}_i)$ | Compare between the old and new food position |
| 30 | $\boldsymbol{x}_i = \boldsymbol{v}_i$ | Select and replace with best solution. |
| 31 | $trial_i = 0$ | Reset the limit counter |
| 32 | Else | |
| 33 | $trial_i = trial_i + 1$ | |
| 34 | End if | |
| 35 | If $f(\boldsymbol{x}_b) < f(\boldsymbol{v}_i)$ | |
| 36 | $\mu_{sucessful} = \mu_{sucessful} + 1$ | |
| 37 | Else | |
| 38 | $\mu_{failure} = \mu_{failure} + 1$ | |
| 39 | End if | |
| 40 | Next $i$ | |
| 41 | For $i = 1{:}SN$ | |
| 42 | $fit(\boldsymbol{x}_i) = \begin{pmatrix} \frac{1}{1 + f(\boldsymbol{x}_i)} \; for\ f(\boldsymbol{x}_i) > 0 \\ 1 + \lvert f(\boldsymbol{x}_i)\rvert \; for\ f(\boldsymbol{x}_i) \leq 0 \end{pmatrix}$ | Calculate the fitness value of each food source |
| 43 | $p_i = \frac{fit(\boldsymbol{x}_i)}{sum(fit)}$ | Calculate probability of each food source |
| 44 | Next $i$ | |
| 45 | While $t \leq SN$ | |
| 46 | Set $i = 1$ | |
| 47 | If $rand < p_i$ | |
| 48 | $t=t+1$ | |
| 49 | For $j= 1{:}D$ | |
| 50 | Produce $v_{ij}$ using line 24 | |
| 51 | Next $j$ | |
| 52 | repeat line 29 to 34 | Apply selection between $\boldsymbol{x}_i$ and $\boldsymbol{v}_i$ |
| 53 | If $f(\boldsymbol{x}_b) < f(\boldsymbol{v}_i)$ | |
| 54 | $\mu_{sucessful} = \mu_{sucessful} + 1$ | |
| 55 | Else | |
| 56 | $\mu_{failure} = \mu_{failure} + 1$ | |
| 57 | End if | |
| 58 | End if | |
| 59 | $i = i + 1$ | |
| 60 | End while | |
| 61 | For $i = 1{:}SN$ | |
| 62 | If $trial_i > limit$ | Abandon the scout bees |

| | | |
|---|---|---|
| 63 | For $j = 1{:}D$ | |
| 64 | $SB_{ij} = c_j^{[L]} + \alpha * \left(c_j^{[U]} - c_j^{[L]}\right)$ | |
| 65 | Next $j$ | |
| 66 | $trial_i = 0$ | |
| 67 | repeat line 29 to 34 | Apply selection between $\boldsymbol{x}_i$ and $\boldsymbol{SB}_i$ |
| 68 | For $i = 1{:}SN$ | |
| 69 | If $i = 1$ | |
| 70 | For $j$= $1{:}D$ | |
| 71 | $\boldsymbol{x_b}(i,j) = \boldsymbol{x}(i,j)$ | Find best solution in the population |
| 72 | Next $j$ | |
| 73 | $f\left(\boldsymbol{x}_b(i)\right) = f\left(\boldsymbol{x}(i)\right)$ | |
| 74 | Else | |
| 75 | If $f(\boldsymbol{x}_i) < f(\boldsymbol{x}_b)$ | |
| 76 | $\boldsymbol{x}_b = \boldsymbol{x}_i$ | |
| 77 | $f(\boldsymbol{x}_b) = f(\boldsymbol{x}_i)$ | Find the global minimum of the current population |
| 78 | Next $i$ | |
| 79 | $\mu_{iter} = \dfrac{\mu_{sucessful}}{\mu_{sucessful} + \mu_{failure}}$ | |
| 80 | $AMR(iter+1) = min\left\{1, \beta + \dfrac{\alpha}{\mu_{iter} + 0.01}\right\}$ | |
| 81 | End while | Iteration $iter$ ends |

The flowchart for the ABC-AMR algorithm is given in Figure 3.1.

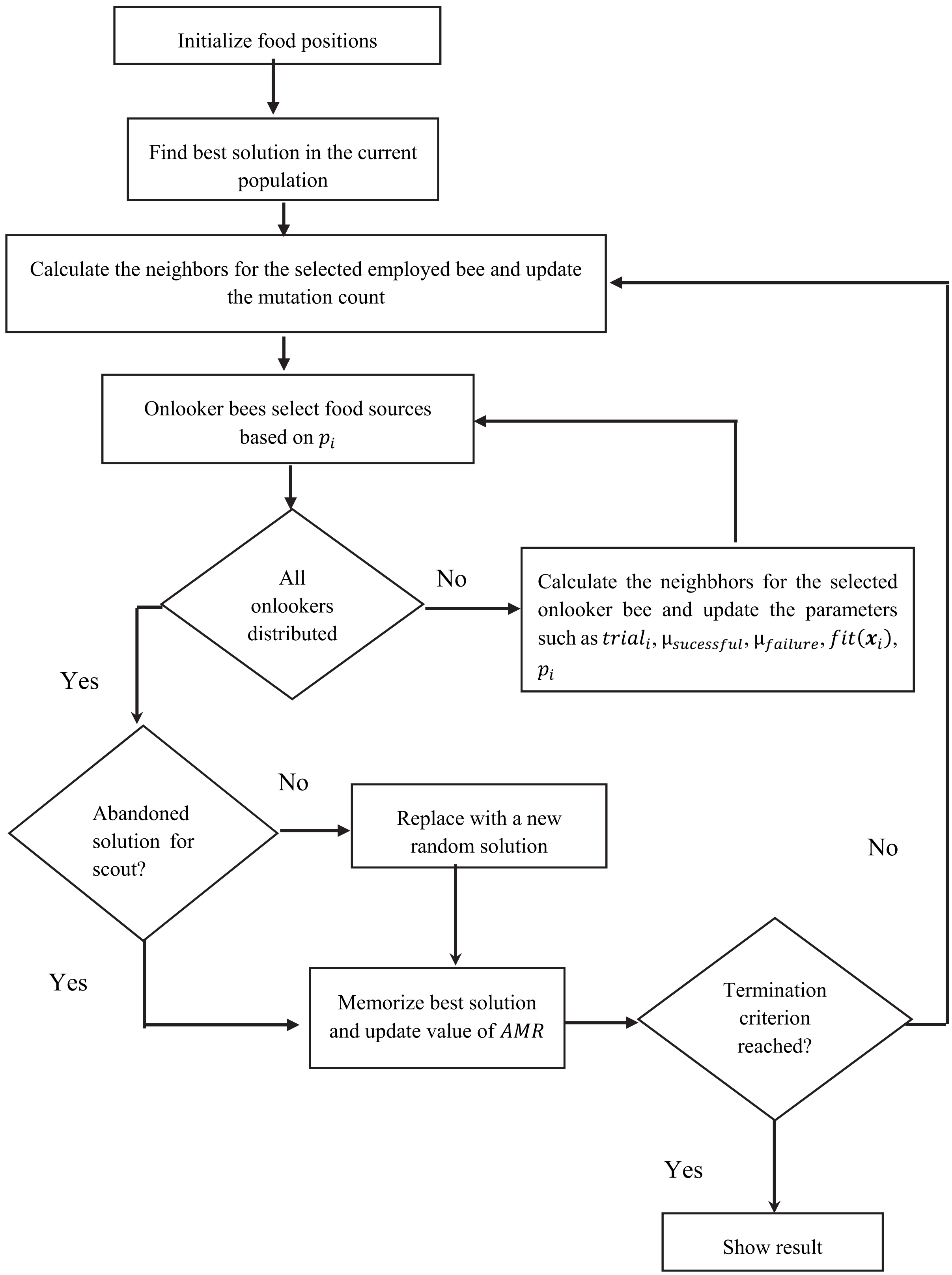


**Figure 3.1** Flowchart of ABC-AMR Algorithm

### 3.3 Minimax FIR Filter Design

The transfer function of an $N$th-order FIR filter can be expressed as,

$$H(z^{-1}) = \sum_{n=0}^{N} h(n) z^{-n} \tag{3.6}$$

The desired frequency response $H_d(w)$ is related to the desired amplitude response $A_d(w)$ by,

$$H_d(w) = je^{-j\tau w} A_d(w_i) = j\left(\frac{w}{\pi} e^{-j\tau w}\right) \quad \text{for } 0 \le w \le \pi \tag{3.7}$$

The parameter τ denotes the group delay.

The frequency response of an ideal differentiator is given by,

$$A_d(w) = \begin{cases} 0 & \text{at } w = 0 \\ \dfrac{w}{\pi} & \text{for } 0 < w < \pi \\ 1 & \text{at } w = \pi \end{cases} \tag{3.8}$$

For linear phase FIR digital filter approximation of a fullband digital differentiator, a practical range of frequency is given by $0 \le w \le w_p \le \pi$ , where $w_p$ equal to 0.9π for Type 3 and for Type 4 $w_p$ equal to π.

#### 3.3.1 Type 3 Linear Phase FIR Filters [1]

The $M$ ($M = N + 1$) impulse responses of Type 3 linear phase FIR filter can be expressed as,

$$\begin{aligned} &\mathbf{h} = [h(0), h(1),, \dots, h(n), \dots , h(M-2), h(M-1)] \\ &h(n) = -h(M-1-n) \text{ for } = 0,1,2,3, \dots, \left(\frac{M-1}{2}\right) \end{aligned} \tag{3.9}$$

The frequency response of the Type 3 linear phase FIR filter is,

$$H(\boldsymbol{c}, w) = je^{-j\frac{M-1}{2}wT} \sum_{n=0}^{\frac{M-3}{2}} 2h(n) \sin\left(\frac{M-1}{2} - n\right) wT$$
$$= je^{-j\frac{M-1}{2}wT} A(\boldsymbol{c}, w) \quad (3.10)$$

$$A(\boldsymbol{c}, w) = \mathbf{c}^{\mathbf{T}} \mathbf{sin}(w) \quad (3.11)$$

where,

$$\mathbf{sin}(w) = \left[\sin(wT) \quad \sin(2wT) \quad \sin(3wT) \quad \cdots \quad \sin\left(\frac{M-1}{2} wT\right)\right]^T$$

$$\boldsymbol{c} = \left[c_1, c_2, c_3, c_4 \ldots \ldots . c_{\left(\frac{M-1}{2}\right)}\right];\ c_0 = h\left(\frac{M-1}{2}\right) = 0 \quad (3.12)$$

$$= \left[2h\left(\frac{M-1}{2} - 1\right), 2h\left(\frac{M-1}{2} - 2\right), \cdots, 2h(2), 2h(1), 2h(0)\right]^T$$

### *3.3.2 Type 4 Linear Phase FIR Filters [1]*

For the Type 4 linear phase FIR filter $M(= N + 1)$ impulse responses can be expressed as,

$$\mathbf{h} = [h(0), h(1), h(2), \ldots, h(n), \ldots\ , h(M-2), h(M-1)]$$
$$h(n) = -h(M-1-n) \text{ for } n = 0,1,2,3, \ldots, \left(\frac{M}{2} - 1\right) \quad (3.13)$$

The frequency response of the Type 4 linear phase FIR filter is given by,

$$H(\mathbf{c}, w) = je^{-j\frac{M-1}{2}wT} \sum_{n=0}^{\frac{M}{2}-1} 2h(n)\, sin\left(\frac{M-1}{2} - n\right) w\,T$$
$$= je^{-j\frac{M-1}{2}wT} A(\mathbf{c}, w) \quad (3.14)$$

where,

$$A(\boldsymbol{c}, w) = \mathbf{c}^{\boldsymbol{T}}\,\mathbf{sin}(w)$$

$$\mathbf{c} = \left[c_0, c_1, c_2, c_3, \dots, c_{\left(\frac{M}{2}\right)}\right]^T$$

$$= \left[2h\left(\frac{M}{2} - 1\right), 2h\left(\frac{M}{2} - 2\right), \cdots, 2h(2), 2h(1), 2h(0)\right]^T \tag{3.15}$$

$$\mathbf{sin}(w) = \left[\sin\left(\frac{w}{2}\right) \quad \sin\left(\frac{3w}{2}\right) \quad \cdots \quad \cdots \quad \sin\left(\frac{M}{2}w\right)\right]^T$$

The group delay of Type 3 and Type 4 Linear phase FIR filter is given by, $\tau(w) = \frac{N}{2}$

The optimization problem searches for an optimal coefficient vector $\boldsymbol{c}$ that minimizes the weighted minimax objective function $e(\mathbf{c})$ with respect to $\mathbf{c}$ defined by,

$$\min_{\mathbf{c}} e(\mathbf{c}) = \min_{\mathbf{c}} \left[\sum_{i=1}^{I} W(w_i)|A(\mathbf{c}, w_i) - A_d(w_i)|^p\right]^{\frac{1}{p}} \quad \text{for } W(w_i) \geq 0;\ 0 \leq w_i \leq \pi \tag{3.16}$$

In equation 3.16, $W(w_i)$ is a positive frequency weighting function, and $p$ is a positive even integer.

### *3.4 Simulation Result Analysis*

The proposed ABC-AMR is used to design Type 3 and Type 4 linear phase FIR differentiators. The filter specification is given in Table 3.2, and Type 3 filters are of orders $N = 14, 26, 50$ and Type 4 filters are of orders $N = 13, 25, 49$ are designed. In order to analyze the performance of the ABC-AMR algorithm, results are compared with the original ABC and its three variants: the gbest-guided ABC algorithm (GABC) [136]; the best-so-far [138]; the ABC/best/1[141]. In addition, results are also compared with the Parks-McClellan (PM) algorithm, a classical filter design method.

The frequency grid for optimization and for peak error value calculation, and ideal amplitude response are given in Table 3.3. For peak error evaluation, the frequency grid is chosen to be denser than that of the frequency grid for optimization.

**Table 3.2** Type 3 and Type 4 Linear Phase FIR Filter Specifications

| | Description | Type 3 | | | Type 4 | | |
|---|---|---|---|---|---|---|---|
| $c_j^{[U]}$ | Upper bound of filter coefficients | 0.47 | | | 0.61 | | |
| $c_j^{[L]}$ | Lower bound of filter coefficients | -0.47 | | | -0.61 | | |
| $N$ | Filter order | 14 | 26 | 50 | 13 | 25 | 49 |
| $K$ | Distinct coefficients | 8 | 14 | 26 | 7 | 13 | 25 |
| $p$ | Least $p$th value | 128 | 128 | 64 | 128 | 128 | 128 |
| $\tau$ | Group delay | 7 | 13 | 25 | 6.5 | 12.5 | 24.5 |
| $Limit$ | Scout bee limit | 200 | 200 | 200 | 200 | 200 | 200 |
| $\alpha$ | AMR parameter 1 | 0.06 | | | | | |
| $\beta$ | AMR parameter 2 | 0.1 | | | | | |

**Table 3.3** Frequency Grid for Optimization and Error Value Calculation

| | | Frequency Grid | Ideal Amplitude Response |
|---|---|---|---|
| Type 3 | Optimization | $[0: 0.005: 0.9]$ | $[0: 0.005: 0.9]$ |
| | Peak error evaluation | $[0: 0.001: 0.9]$ | $[0: 0.001: 0.9]$ |
| Type 4 | Optimization | $[0: 0.005: 1]$ | $[0: 0.005: 1]$ |
| | Peak error evaluation | $[0: 0.001: 1]$ | $[0: 0.001: 1]$ |

For a fair comparison of performance, the five ABC algorithms are set to the same initial conditions. The population size $SN$ =50, $limit$ =200 and maximum number of function evaluation is set as; 100000 for $N = 13,14,25,$ and $26$, 200000 for $N = 49$ and $50$. The initial population for each of the filter designs is generated using equation 3.1, where the upper bound ($c_j^{[U]}$) and lower bound ($c_j^{[L]}$) of coefficients is set according to Table 3.1. The number of scout bee is limited to a maximum of one in every iteration.

To evaluate the performance of the ABC-AMR algorithm, peak error values, peak error location ($F_{pe}$), converged iteration number ($\Gamma_c$) and minimax error value are compared with those of the Parks-McClellan (PM) optimal equiripple FIR filter design [11] (by Matlab function firpm.m), the original ABC algorithm, the gbest-guided ABC algorithm[136], the best-so-far selection ABC algorithm [138], the global best ABC algorithm [141] and the results are given in Table 3.4 and Table 3.5. The simulations are performed using intel core i7-4790, 3.60 GHz with 12GB RAM desktop computer. The results are ranked according to peak error $R_{pe}$ and minimax error $R_{MM}$. For Type 3 and Type 4 FIR differentiator designs, the ABC-AMR has peak error rank $R_{pe} = 1$ and minimax error rank $R_{MM} = 1$. The error ranking indicates that the ABC-AMR always has the lowest minimax and peak errors for all designed filters. The converged iteration number $\Gamma_c$ indicates that the ABC-AMR algorithm can always converge to the lowest minimax error with the least CPU time. The CPU time (in seconds) are the time required by each design to converge to its least minimax error value. It can be seen that the ABC-AMR algorithm requires the least CPU time than those of the original ABC algorithm and its three variants. For the ABC-AMR algorithm, the value of the modification rate is changed adaptively. During the initial stage of iterations, the adaptive modification rate values are higher and updating all or many of the design parameters to fully explore the search space; but during the later stage of iterations, the adaptive modification rate values are lower and updating a few or no design parameters. For other variants of the ABC algorithm, their modification rates are fixed throughout optimization thus changing at most a fixed number of design parameters in all iterations. The PM algorithm requires the least CPU time in each of the designs because it is based on the Remez exchange algorithm which is a non-evolutionary algorithm.

Figures 3.2-3.13 show the magnitude and error value plots for differentiators of orders $N = 13, 25, 49, 14, 26, \text{and } 50$. Magnitude response plot indicates that the ABC-AMR can design linear phase FIR differentiator with desired magnitude response and equiripple minimax error in passband. The error convergence plot indicates that the minimax error decreases rapidly in initial stages and gradually in later stages to search around the neighborhood of known solutions to find a better solution.

**Table 3.4** Error Values and Iteration Time for Type 3 Linear Phase FIR Differentiator

| | $N$ | Algorithm | $R_{pe}$ | Peak error | $F_{pe}$ | $\Gamma_c$ | CPU(S) | $R_{MM}$ | Minimax error |
|---|---|---|---|---|---|---|---|---|---|
| Type 3 | 14 | PM | 6 | 0.066117615511389 | 0.900 | - | 0.069407 | 2 | 0.066117887296875 |
| | | ABC | 2 | 0.051514296309586 | 0.900 | 40312 | 2275.5053 | 1 | 0.052006124426246 |
| | | GABC | 5 | 0.051514296597022 | 0.900 | 39332 | 6946.6687 | 1 | 0.052006124426246 |
| | | Best-so-far | 3 | 0.051514296474592 | 0.900 | 48407 | 2712.00006 | 1 | 0.052006124426246 |
| | | ABC/best/1 | 4 | 0.051514296522112 | 0.900 | 55610 | 4196.290 | 1 | 0.052006124426246 |
| | | ABC-AMR | 1 | 0.051514296303430 | 0.900 | 9908 | 735.1374 | 1 | 0.052006124426246 |
| | 26 | PM | 6 | 0.028220598415362 | 0.900 | - | 0.073618 | 5 | 0.028222359050006 |
| | | ABC | 2 | 0.006029913166200 | 0.881 | 73849 | 7126.505 | 1 | 0.006064388026611 |
| | | GABC | 4 | 0.007429116163160 | 0.883 | 70573 | 7408.914 | 3 | 0.007416102739156 |
| | | Best-so-far | 3 | 0.006029915563162 | 0.881 | 92444 | 9385.135 | 2 | 0.006064388311630 |
| | | ABC/best/1 | 5 | 0.007724479423304 | 0.900 | 91415 | 10240.644 | 4 | 0.007731308990839 |
| | | ABC-AMR | 1 | 0.006029913032173 | 0.881 | 44342 | 5575.3419 | 1 | 0.006064388026611 |
| | 50 | PM | 6 | 0.007403673670594 | 0.900 | - | 0.091590 | 6 | 0.007441118372404 |
| | | ABC | 3 | 0.002819160922123 | 0.892 | 192477 | 24994.6830 | 2 | 0.002731140067982 |
| | | GABC | 2 | 0.002763787333546 | 0.866 | 191644 | 25672.9374 | 3 | 0.002832611168084 |
| | | Best-so-far | 4 | 0.002862640017806 | 0.900 | 197454 | 28055.4130 | 4 | 0.002925127734139 |
| | | ABC/best/1 | 5 | 0.002933139606401 | 0.834 | 196236 | 27452.7855 | 5 | 0.002975775743748 |
| | | ABC-AMR | 1 | 0.002593144300438 | 0.867 | 118575 | 24749.1102 | 1 | 0.002626326481536 |

**Table 3.5** Error Values and Iteration Time for Type 4 Linear Phase FIR Differentiator

| | $N$ | Algorithm | $R_{pe}$ | Peak error | $F_{pe}$ | $\Gamma_c$ | CPU(S) | $R_{MM}$ | Minimax error |
|---|---|---|---|---|---|---|---|---|---|
| Type 4 | 13 | PM | 6 | 0.015923710156487 | 1.000 | - | 0.067788 | 5 | 0.015923755003422 |
| | | ABC | 4 | 0.013717944150863 | 1.000 | 36592 | 2069.2264 | 3 | 0.013833071007229 |
| | | GABC | 3 | 0.013717589695070 | 1.000 | 16460 | 946.8203 | 2 | 0.013832566298263 |
| | | Best-so-far | 5 | 0.013830923466301 | 1.000 | 38456 | 2261.7191 | 4 | 0.013935503360277 |
| | | ABC/best/1 | 2 | 0.013716611003032 | 1.000 | 50855 | 3806.2243 | 1 | 0.013828588348102 |
| | | ABC-AMR | 1 | 0.013716610869623 | 1.000 | 7632 | 634.804 | 1 | 0.013828588348102 |
| | 25 | PM | 6 | 0.007804819698411 | 1.000 | - | 0.072492 | 6 | 0.007805357815019 |
| | | ABC | 3 | 0.007234436308776 | 0.969 | 65254 | 6298.564 | 3 | 0.007273918355556 |
| | | GABC | 2 | 0.007194377579454 | 0.969 | 69280 | 7271.8700 | 2 | 0.007236540757836 |
| | | Best-so-far | 5 | 0.007411637186501 | 0.968 | 88425 | 8980.629 | 5 | 0.007439586292932 |
| | | ABC/best/1 | 4 | 0.007270852693430 | 0.969 | 80012 | 9144.50309 | 4 | 0.007306913927617 |
| | | ABC-AMR | 1 | 0.007160755179815 | 0.969 | 38445 | 5114.9564 | 1 | 0.007203605712570 |
| | 49 | PM | 5 | 0.003821668222839 | 1.000 | - | 0.090561 | 6 | 0.003823629391263 |
| | | ABC | 2 | 0.003739585793212 | 0.984 | 188963 | 22769.9445 | 3 | 0.003741829681883 |
| | | GABC | 3 | 0.003740358604837 | 0.984 | 183675 | 21974.7351 | 2 | 0.003740424216878 |
| | | Best-so-far | 6 | 0.003828893477635 | 0.984 | 178464 | 24787.4834 | 5 | 0.003813753805983 |
| | | ABC/best/1 | 4 | 0.003770426383939 | 0.984 | 185770 | 25179.9125 | 4 | 0.003769995018554 |
| | | ABC-AMR | 1 | 0.003655797073296 | 0.984 | 112977 | 20596.7065 | 1 | 0.003671289904950 |

The half symmetric filter coefficients, $h(n) = -h(M-1-n)$ for Type 3 and Type 4 linear phase FIR differentiators using the ABC-AMR and its variants are given in Table 3.6 to Table 3.15.

**Table 3.6** Half Symmetric Filter Coefficients of Type 3 Differentiator Using Original ABC Algorithm

| $h(n)$ | Filter order $N = 14$ | Filter order $N = 26$ | Filter order $N = 50$ |
|---|---|---|---|
| $h(0)$ | 0.037714736076781 | 0.004954265024730 | 0.002009877760662 |
| $h(1)$ | -0.030646457946389 | -0.005142636311658 | -0.001962936003393 |
| $h(2)$ | 0.043867404494912 | 0.007690350241261 | 0.002702320368751 |
| $h(3)$ | -0.063006616462289 | -0.011025642990155 | -0.003715612764887 |
| $h(4)$ | 0.093235497566831 | 0.015327310488592 | 0.004918063724640 |
| $h(5)$ | -0.150362207508295 | -0.020848200236187 | -0.006233546758396 |
| $h(6)$ | 0.313848202259717 | 0.027971747369298 | 0.007873408152949 |
| $h(7)$ | 0.000000000000000 | -0.037321606415275 | -0.009565069968085 |
| $h(8)$ | | 0.050014192921637 | 0.011554864554546 |
| $h(9)$ | | -0.068291703677568 | -0.013713425247137 |
| $h(10)$ | | 0.097417827390313 | 0.015994776897288 |
| $h(11)$ | | -0.153257097029880 | -0.018580888154193 |
| $h(12)$ | | 0.315328318461610 | 0.021273554338286 |
| $h(13)$ | | 0.000000000000000 | -0.024282180962679 |
| $h(14)$ | | | 0.027567845613665 |
| $h(15)$ | | | -0.031169035563579 |
| $h(16)$ | | | 0.035439986834320 |
| $h(17)$ | | | -0.040264001348586 |
| $h(18)$ | | | 0.046397246036947 |
| $h(19)$ | | | -0.054129634329298 |
| $h(20)$ | | | 0.064833998659169 |
| $h(21)$ | | | -0.080712127820842 |
| $h(22)$ | | | 0.107003047556673 |
| $h(23)$ | | | -0.159867741929711 |
| $h(24)$ | | | 0.318652110954695 |
| $h(25)$ | | | 0.000000000000000 |

**Table 3.7** Half Symmetric Filter Coefficients of Type 4 Differentiator Using Original ABC Algorithm

| $h(n)$ | Filter order $N = 13$ | Filter order $N = 25$ | Filter order $N = 49$ |
|---|---|---|---|
| $h(0)$ | 0.008264611602191 | 0.003456458479103 | 0.001131662874766 |
| $h(1)$ | -0.003734160867159 | -0.000931000562800 | -0.000825551624798 |
| $h(2)$ | 0.005323149013052 | 0.001576209070183 | 0.000359541930270 |
| $h(3)$ | -0.008563001766520 | -0.001097322552584 | -0.000197411511262 |
| $h(4)$ | 0.016496826494759 | 0.001522229229158 | 0.000178128965830 |
| $h(5)$ | -0.045258766102300 | -0.001903632624033 | -0.000216197304269 |
| $h(6)$ | 0.405500512078587 | 0.001961829606336 | 0.000512598346314 |
| $h(7)$ | | -0.003629656791606 | -0.000594771104760 |
| $h(8)$ | | 0.005750120979190 | -0.000092578681664 |
| $h(9)$ | | -0.007959226461326 | -0.000632159746196 |
| $h(10)$ | | 0.015824641136634 | 0.000748440655474 |
| $h(11)$ | | -0.045392750674101 | -0.000297826283784 |
| $h(12)$ | | 0.405388123899742 | 0.000893869910396 |
| $h(13)$ | | | -0.000616016544671 |
| $h(14)$ | | | 0.001397836740974 |
| $h(15)$ | | | -0.000518928633199 |
| $h(16)$ | | | 0.001654519679030 |
| $h(17)$ | | | -0.001838600059342 |
| $h(18)$ | | | 0.002276457328537 |
| $h(19)$ | | | -0.003225241109209 |
| $h(20)$ | | | 0.005198778418921 |
| $h(21)$ | | | -0.008287336282587 |
| $h(22)$ | | | 0.016488580851259 |
| $h(23)$ | | | -0.044547729139694 |
| $h(24)$ | | | 0.405599491947542 |

**Table 3.8** Half Symmetric Filter Coefficients of Type 3 Differentiator Using Global Best ABC Algorithm

| $h(n)$ | Filter order $N = 14$ | Filter order $N = 26$ | Filter order $N = 50$ |
|---|---|---|---|
| $h(0)$ | 0.037714736009138 | 0.004674426070205 | 0.002152664796011 |
| $h(1)$ | -0.030646457789720 | -0.004965705205898 | -0.002107274777169 |
| $h(2)$ | 0.043867404499983 | 0.009029362614518 | 0.002977840459958 |
| $h(3)$ | -0.063006616609127 | -0.011376635083456 | -0.004119921812824 |
| $h(4)$ | 0.093235497472416 | 0.015015536012812 | 0.005429088813692 |
| $h(5)$ | -0.150362207509868 | -0.020971347981832 | -0.006984835210369 |
| $h(6)$ | 0.313848202229764 | 0.028793847811993 | 0.008732118907782 |
| $h(7)$ | 0.000000000000000 | -0.038025614268367 | -0.010709339351377 |
| $h(8)$ | | 0.049710001677923 | 0.012888102615146 |
| $h(9)$ | | -0.068207792550969 | -0.015249874167740 |
| $h(10)$ | | 0.098405398965465 | 0.017813766307454 |
| $h(11)$ | | -0.153535094502451 | -0.020541738180927 |
| $h(12)$ | | 0.314632861951947 | 0.023476101659197 |
| $h(13)$ | | 0.000000000000000 | -0.026607031077811 |
| $h(14)$ | | | 0.029995663126626 |
| $h(15)$ | | | -0.033709092440984 |
| $h(16)$ | | | 0.037895451620700 |
| $h(17)$ | | | -0.042752690137101 |
| $h(18)$ | | | 0.048681725063819 |
| $h(19)$ | | | -0.056287618596107 |
| $h(20)$ | | | 0.066720562906411 |
| $h(21)$ | | | -0.082272439369876 |
| $h(22)$ | | | 0.108281166416444 |
| $h(23)$ | | | -0.160667144184671 |
| $h(24)$ | | | 0.319099195823076 |
| $h(25)$ | | | 0.000000000000000 |

**Table 3.9** Half Symmetric Filter Coefficients of Type 4 Differentiator Using Global Best ABC Algorithm

| $h(n)$ | Filter order $N = 13$ | Filter order $\mathrm{N} = 25$ | Filter order $\mathrm{N} = 49$ |
|---|---|---|---|
| $h(0)$ | 0.008224495336062 | 0.003714560943920 | 0.001358938937626 |
| $h(1)$ | -0.003819631604500 | -0.000844219812620 | -0.000195168872497 |
| $h(2)$ | 0.005301540181976 | 0.001303180541707 | 0.000728909070773 |
| $h(3)$ | -0.008532369555134 | -0.001292029712979 | -0.000401877288253 |
| $h(4)$ | 0.016504191425383 | 0.001380624610192 | -0.000194379703133 |
| $h(5)$ | -0.045236848269014 | -0.001801990478631 | -0.000447296071939 |
| $h(6)$ | 0.405522128780396 | 0.002447324393796 | 0.000648866905644 |
| $h(7)$ | | -0.003435283644091 | -0.000403099276249 |
| $h(8)$ | | 0.005195581228609 | 0.000296007917183 |
| $h(9)$ | | -0.008446820306135 | -0.000360913544612 |
| $h(10)$ | | 0.016072605578480 | 0.000539755869478 |
| $h(11)$ | | -0.044903330116540 | -0.000428096993863 |
| $h(12)$ | | 0.405575314265407 | 0.000666429183279 |
| $h(13)$ | | | -0.000641986366391 |
| $h(14)$ | | | 0.001158207299409 |
| $h(15)$ | | | -0.001230224391820 |
| $h(16)$ | | | 0.001447145485385 |
| $h(17)$ | | | -0.001564822512776 |
| $h(18)$ | | | 0.002620272424253 |
| $h(19)$ | | | -0.003523829967103 |
| $h(20)$ | | | 0.004846977621542 |
| $h(21)$ | | | -0.007966822714279 |
| $h(22)$ | | | 0.016453944602314 |
| $h(23)$ | | | -0.045302200987820 |
| $h(24)$ | | | 0.405106841718536 |

**Table 3.10** Half Symmetric Filter Coefficients of Type 3 Differentiator Using Best-so-far ABC Algorithm

| $h(n)$ | Filter order $N = 14$ | Filter order $N = 26$ | Filter order $N = 50$ |
|---|---|---|---|
| $h(0)$ | 0.037714736121476 | 0.004954259527733 | -0.000032836966237 |
| $h(1)$ | -0.030646457730699 | -0.005142644076803 | -0.000132886675808 |
| $h(2)$ | 0.043867404265145 | 0.007690345957815 | 0.000594670162447 |
| $h(3)$ | -0.063006617025036 | -0.011025643056894 | -0.001256382067019 |
| $h(4)$ | 0.093235497182417 | 0.015327309040087 | 0.002126595529783 |
| $h(5)$ | -0.150362207674508 | -0.020848256847276 | -0.002826070041617 |
| $h(6)$ | 0.313848202241384 | 0.027971580836796 | 0.003839059235563 |
| $h(7)$ | 0.000000000000000 | -0.037321769301587 | -0.004921526740999 |
| $h(8)$ | | 0.050014138214706 | 0.006322080206127 |
| $h(9)$ | | -0.068291703563541 | -0.008204954658998 |
| $h(10)$ | | 0.097417826919175 | 0.010108023416536 |
| $h(11)$ | | -0.153257096703735 | -0.011895291241129 |
| $h(12)$ | | 0.315328318546392 | 0.014117905894228 |
| $h(13)$ | | 0.000000000000000 | -0.017167992661310 |
| $h(14)$ | | | 0.020711081721939 |
| $h(15)$ | | | -0.024417034543224 |
| $h(16)$ | | | 0.028601936480747 |
| $h(17)$ | | | -0.033566791871325 |
| $h(18)$ | | | 0.041014515033587 |
| $h(19)$ | | | -0.047980076612729 |
| $h(20)$ | | | 0.060082956339386 |
| $h(21)$ | | | -0.076950530947972 |
| $h(22)$ | | | 0.104042198831171 |
| $h(23)$ | | | -0.157826070471847 |
| $h(24)$ | | | 0.317792797476610 |
| $h(25)$ | | | 0.000000000000000 |

**Table 3.11** Half Symmetric Filter Coefficients of Type 4 Differentiator Using Best-so-far ABC Algorithm

| $h(n)$ | Filter order $N = 13$ | Filter order $N = 25$ | Filter order $N = 49$ |
|---|---|---|---|
| $h(0)$ | 0.007851524961023 | 0.002206301273539 | 0.000554713367324 |
| $h(1)$ | -0.003837151453875 | -0.001989888149987 | -0.000974404720332 |
| $h(2)$ | 0.005885549611941 | 0.001945525719095 | 0.000579176022998 |
| $h(3)$ | -0.008099425356908 | -0.000871349385500 | -0.000608190926336 |
| $h(4)$ | 0.016445823551601 | 0.000847765652946 | -0.000084377500657 |
| $h(5)$ | -0.045901717204137 | -0.002593026113114 | -0.000196225732242 |
| $h(6)$ | 0.405063346127365 | 0.002922275640087 | 0.000225542164776 |
| $h(7)$ | | -0.001870539119004 | -0.000645997763767 |
| $h(8)$ | | 0.006380672747193 | 0.000337599333984 |
| $h(9)$ | | -0.007744922791641 | -0.000857886312354 |
| $h(10)$ | | 0.016739464093374 | -0.000077314771834 |
| $h(11)$ | | -0.044798826920954 | -0.000852370706714 |
| $h(12)$ | | 0.405400275989127 | 0.000922844421198 |
| $h(13)$ | | | -0.000733090405356 |
| $h(14)$ | | | 0.000845385941315 |
| $h(15)$ | | | -0.000737090063026 |
| $h(16)$ | | | 0.001871503343967 |
| $h(17)$ | | | -0.001913582204346 |
| $h(18)$ | | | 0.001807030605793 |
| $h(19)$ | | | -0.003438095023633 |
| $h(20)$ | | | 0.005377618816994 |
| $h(21)$ | | | -0.008212045233431 |
| $h(22)$ | | | 0.015858611079945 |
| $h(23)$ | | | -0.045511893173979 |
| $h(24)$ | | | 0.405209005225721 |

**Table 3.12** Half Symmetric Filter Coefficients of Type 3 Differentiator Using ABC/Best/1 Algorithm

| $h(n)$ | Filter order $N = 14$ | Filter order $N = 26$ | Filter order $N = 50$ |
|---|---|---|---|
| $h(0)$ | -0.037714736164276 | 0.004220039506768 | 0.001839105420516 |
| $h(1)$ | 0.030646457581555 | -0.004288276151991 | -0.001910199952512 |
| $h(2)$ | -0.043867404512256 | 0.007199752790122 | 0.003149886559634 |
| $h(3)$ | 0.063006616804097 | -0.010876656635048 | -0.003953919537754 |
| $h(4)$ | -0.093235497256590 | 0.015637756607592 | 0.005188939125532 |
| $h(5)$ | 0.150362207655403 | -0.019971948902219 | -0.006704364927861 |
| $h(6)$ | -0.313848202233923 | 0.027302859780655 | 0.008603755688190 |
| $h(7)$ | 0.000000000000000 | -0.035986080161137 | -0.010310992382171 |
| $h(8)$ | | 0.049952503552586 | 0.012412525242013 |
| $h(9)$ | | -0.068782874196260 | -0.015087715398807 |
| $h(10)$ | | 0.097355606114765 | 0.017339422474736 |
| $h(11)$ | | -0.153232418301578 | -0.019995862873859 |
| $h(12)$ | | 0.314589340588941 | 0.022986793678869 |
| $h(13)$ | | 0.000000000000000 | -0.026083225161564 |
| $h(14)$ | | | 0.029586217620524 |
| $h(15)$ | | | -0.033042633225191 |
| $h(16)$ | | | 0.037373435814157 |
| $h(17)$ | | | -0.042439810134593 |
| $h(18)$ | | | 0.048050996709027 |
| $h(19)$ | | | -0.055763240640195 |
| $h(20)$ | | | 0.066436979889549 |
| $h(21)$ | | | -0.082091894225675 |
| $h(22)$ | | | 0.107949454389511 |
| $h(23)$ | | | -0.160306615097844 |
| $h(24)$ | | | 0.319250472974861 |
| $h(25)$ | | | 0.000000000000000 |

**Table 3.13** Half Symmetric Filter Coefficients of Type 4 Differentiator Using ABC/Best/1 Algorithm

| $h(n)$ | Filter order $N = 13$ | Filter order $N = 25$ | Filter order $N = 49$ |
|---|---|---|---|
| $h(0)$ | 0.008255554448067 | 0.003290001482898 | 0.000711829080052 |
| $h(1)$ | -0.003764118367036 | -0.000927459053777 | -0.001143547922293 |
| $h(2)$ | 0.005327397631421 | 0.001656340302054 | 0.000225497378645 |
| $h(3)$ | -0.008549461429752 | -0.001291014586894 | -0.000623050252667 |
| $h(4)$ | 0.016463064983330 | 0.001304670229115 | 0.000176823483357 |
| $h(5)$ | -0.045268017757269 | -0.001925551586051 | -0.000100220958177 |
| $h(6)$ | 0.405514079881610 | 0.002473261030463 | 0.000434827497338 |
| $h(7)$ | | -0.003160254335341 | -0.000450897482858 |
| $h(8)$ | | 0.005219359329427 | 0.000013801901191 |
| $h(9)$ | | -0.008534704725573 | -0.000673806087883 |
| $h(10)$ | | 0.016309447504654 | 0.000661476094253 |
| $h(11)$ | | -0.044670512112261 | -0.000676693091833 |
| $h(12)$ | | 0.405611896389999 | 0.000372303822229 |
| $h(13)$ | | | -0.000657419115955 |
| $h(14)$ | | | 0.001198338948212 |
| $h(15)$ | | | -0.001056419142790 |
| $h(16)$ | | | 0.001600542003052 |
| $h(17)$ | | | -0.001662869289421 |
| $h(18)$ | | | 0.002478121058722 |
| $h(19)$ | | | -0.003356937098225 |
| $h(20)$ | | | 0.004851174768479 |
| $h(21)$ | | | -0.008321459153870 |
| $h(22)$ | | | 0.016497901242642 |
| $h(23)$ | | | -0.044979642057608 |
| $h(24)$ | | | 0.405204708831877 |

**Table 3.14** Half Symmetric Filter Coefficients of Type 3 Differentiator Using ABC-AMR Algorithm

| $h(n)$ | Filter order $\mathrm{N} = 14$ | Filter order $\mathrm{N} = 26$ | Filter order $\mathrm{N} = 50$ |
|---|---|---|---|
| $h(0)$ | 0.037714736176857 | 0.004954265018781 | -0.001910167610739 |
| $h(1)$ | -0.030646457800335 | -0.005142636541968 | 0.001672136644021 |
| $h(2)$ | 0.043867404141397 | 0.007690350125654 | -0.002289726027502 |
| $h(3)$ | -0.063006617005233 | -0.011025643088221 | 0.002992403272284 |
| $h(4)$ | 0.093235497362153 | 0.015327310486198 | -0.003741074729426 |
| $h(5)$ | -0.150362207551728 | -0.020848200304400 | 0.004515940240083 |
| $h(6)$ | 0.313848202383412 | 0.027971747495007 | -0.005250402589426 |
| $h(7)$ | 0.000000000000000 | -0.037321606067086 | 0.005893854058011 |
| $h(8)$ | | 0.050014193374240 | -0.006382049157276 |
| $h(9)$ | | -0.068291703248154 | 0.006631870688049 |
| $h(10)$ | | 0.097417827781593 | -0.006552144938983 |
| $h(11)$ | | -0.153257096940842 | 0.006047719642627 |
| $h(12)$ | | 0.315328318498894 | -0.005012048776954 |
| $h(13)$ | | 0.000000000000000 | 0.003330449046257 |
| $h(14)$ | | | -0.000868567279514 |
| $h(15)$ | | | -0.002529766108257 |
| $h(16)$ | | | 0.007055912845817 |
| $h(17)$ | | | -0.012967351753279 |
| $h(18)$ | | | 0.020645267217845 |
| $h(19)$ | | | -0.030703784871900 |
| $h(20)$ | | | 0.044250581180524 |
| $h(21)$ | | | -0.063513051960077 |
| $h(22)$ | | | 0.093732312489866 |
| $h(23)$ | | | -0.150750496049649 |
| $h(24)$ | | | 0.314059635244287 |
| $h(25)$ | | | 0.000000000000000 |

**Table 3.15** Half Symmetric Filter Coefficients of Type 4 Differentiator Using ABC-AMR Algorithm

| $h(n)$ | Filter order $N = 13$ | Filter order $N = 25$ | Filter order $N = 49$ |
|---|---|---|---|
| $h(0)$ | 0.008255554456441 | 0.003873473894564 | 0.001857807284670 |
| $h(1)$ | -0.003764118423671 | -0.000911167085171 | -0.000247312422273 |
| $h(2)$ | 0.005327397697098 | 0.001030826173696 | 0.000231224785166 |
| $h(3)$ | -0.008549461520826 | -0.001220503836268 | -0.000254024769798 |
| $h(4)$ | 0.016463065018200 | 0.001489876281312 | 0.000271610748705 |
| $h(5)$ | -0.045268017676092 | -0.001881306278383 | -0.000285543084896 |
| $h(6)$ | 0.405514079772859 | 0.002473041630763 | 0.000328745741485 |
| $h(7)$ | | -0.003420698035191 | -0.000354708072560 |
| $h(8)$ | | 0.005071899604802 | 0.000398652302049 |
| $h(9)$ | | -0.008337174823228 | -0.000439066664149 |
| $h(10)$ | | 0.016275604286083 | 0.000502872526030 |
| $h(11)$ | | -0.045094556374526 | -0.000575643796129 |
| $h(12)$ | | 0.405346984265227 | 0.000665768494149 |
| $h(13)$ | | | -0.000783077632344 |
| $h(14)$ | | | 0.000941385883182 |
| $h(15)$ | | | -0.001134359900530 |
| $h(16)$ | | | 0.001423518084956 |
| $h(17)$ | | | -0.001816444057892 |
| $h(18)$ | | | 0.002412641689390 |
| $h(19)$ | | | -0.003372281187760 |
| $h(20)$ | | | 0.005011515285196 |
| $h(21)$ | | | -0.008298331977075 |
| $h(22)$ | | | 0.016217604219189 |
| $h(23)$ | | | -0.045056663350083 |
| $h(24)$ | | | 0.405298004227175 |

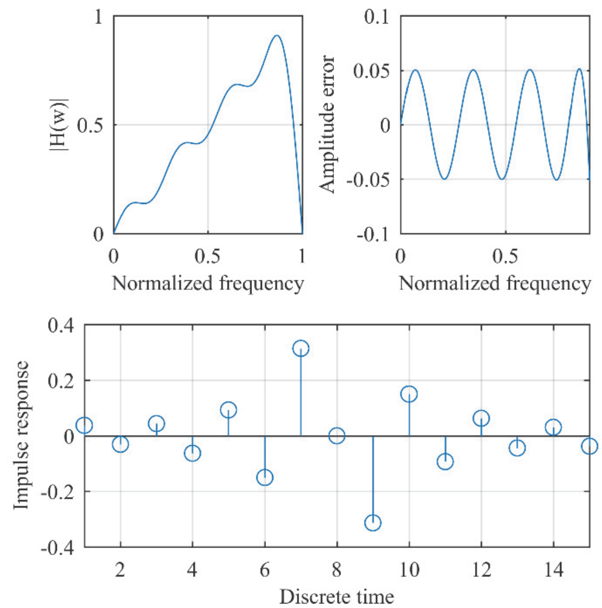


**Figure 3.2** Magnitude Response, Passband Error and Impulse Response of Filter Order $N = 14$

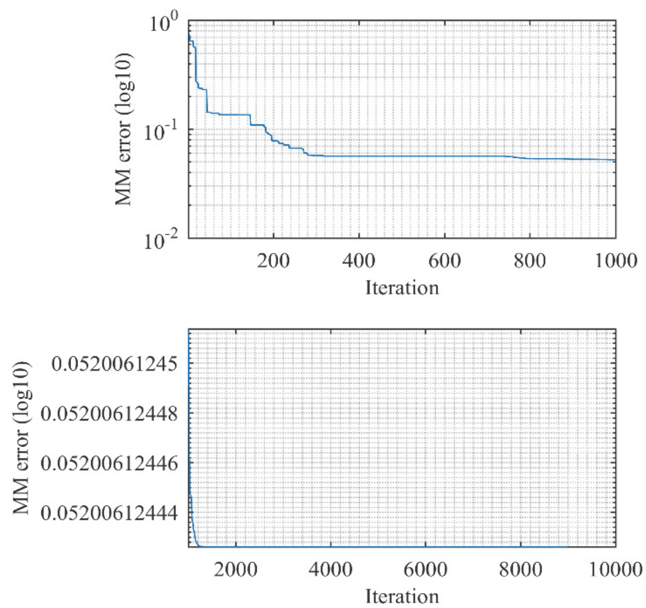


**Figure 3.3** Minimax Error Convergence Curve of Filter Order $N = 14$

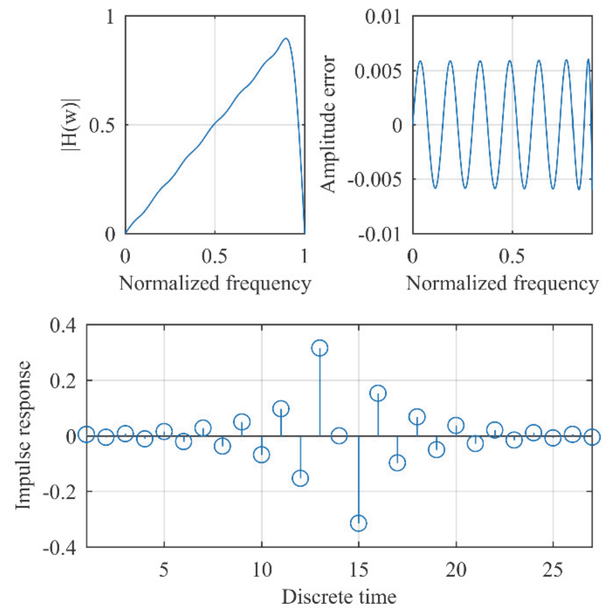


**Figure 3.4** Magnitude Response, Passband Error and Impulse Response of Filter Order $N = 26$

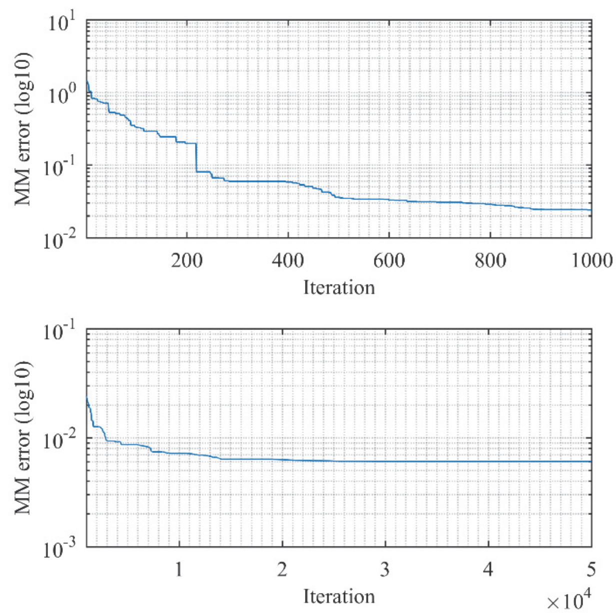


**Figure 3.5** Minimax Error Convergence Curve of Filter Order $N = 26$

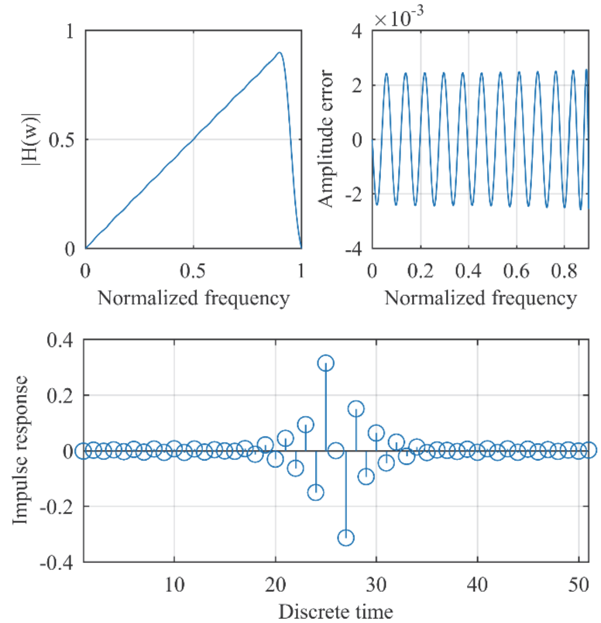


**Figure 3.6** Magnitude Response, Passband Error and Impulse Response of Filter Order $N = 50$

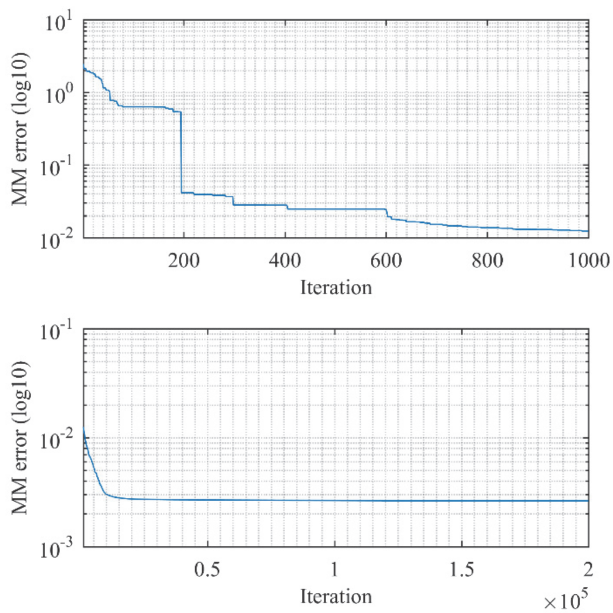


**Figure 3.7** Minimax Error Convergence Curve of Filter Order $N = 50$

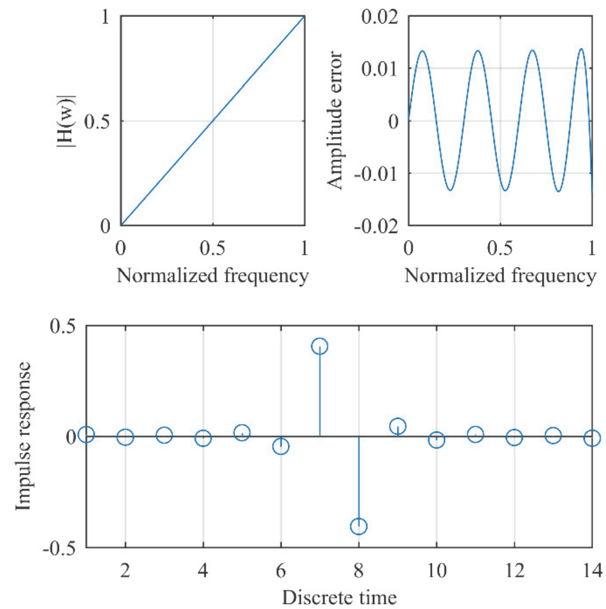


**Figure 3.8** Magnitude Response, Passband Error and Impulse Response of Filter Order $N = 13$

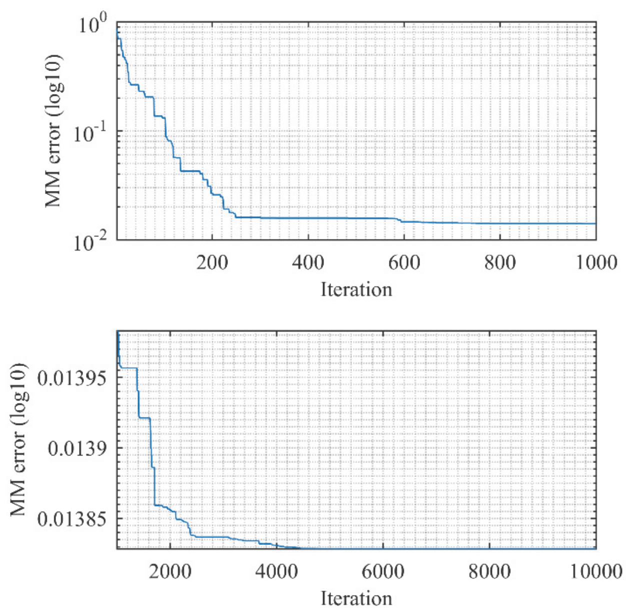


**Figure 3.9** Minimax Error Convergence Curve of Filter Order $N = 13$

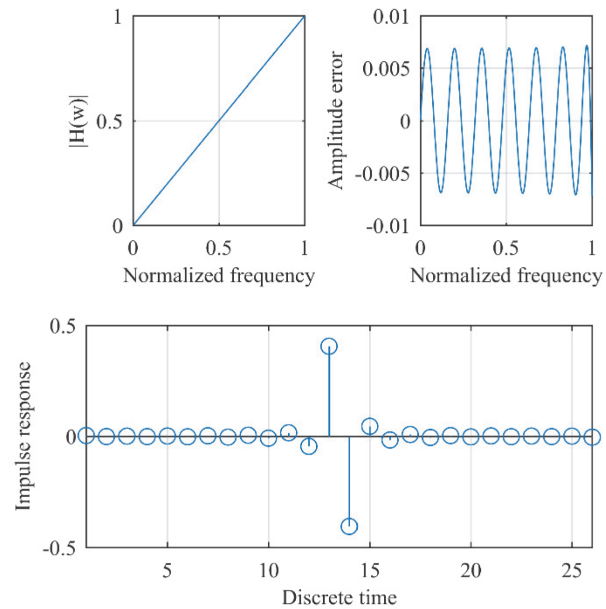


**Figure 3.10** Magnitude Response, Passband Error and Impulse Response of Filter Order $N = 25$

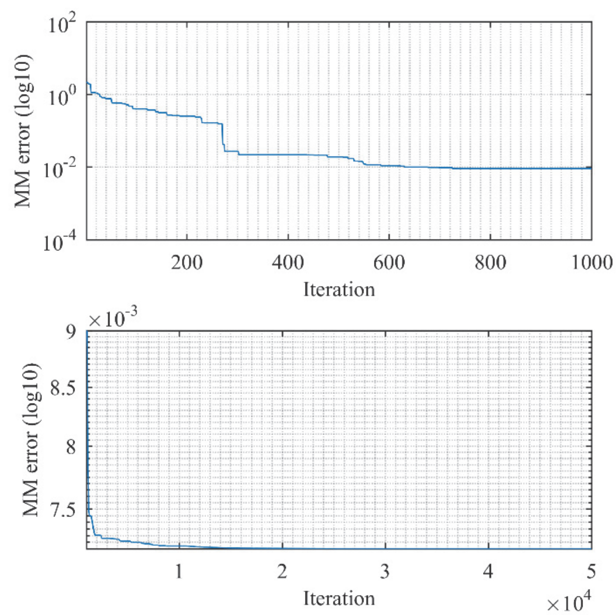


**Figure 3.11** Minimax Error Convergence Curve of Filter Order $N = 25$

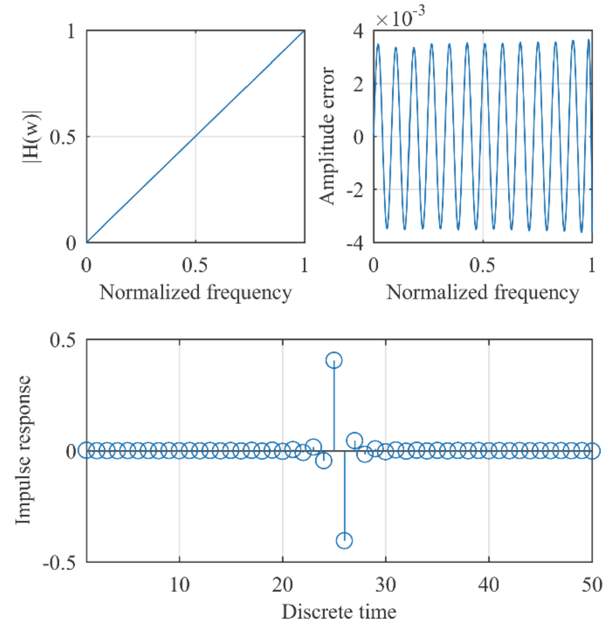


**Figure 3.12** Magnitude Response, Passband Error and Impulse Response of Filter Order $N = 49$

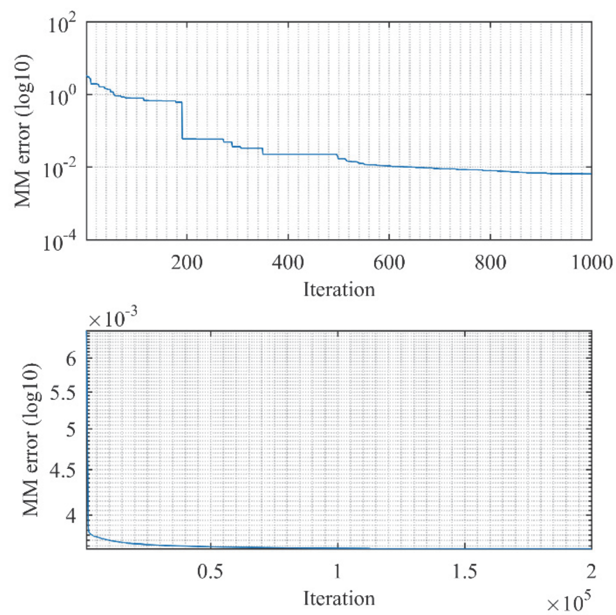


**Figure 3.13** Minimax Error Convergence Curve of Filter Order $N = 49$

### *3.5 Conclusions*

In this chapter, a novel improvement to the ABC algorithm, called the ABC-AMR is proposed. In the ABC-AMR, the diversity of a search space is adaptively controlled by increasing the rate of exploration in early stages and increasing the rate of exploitation in later stages of optimization. To evaluate the performance of the proposed ABC-AMR algorithm in filter design applications, the original ABC algorithm and its three variants are used to design linear phase Types 3 and 4 FIR differentiators. Simulation results indicate that the ABC-AMR can reach lower peak error and minimax error and with smaller numbers of iterations. Unlike other evolutionary algorithms, the initial population of the ABC-AMR algorithm needs not be seeded with a good candidate solution, but can be randomly initialized within a range, and thereby ensuring diversity in a search space.

# CHAPTER 4

# SPARSE FIR FILTER DESIGN

Linear phrase FIR filters are widely used in digital signal processing, and communication and medical imaging due to their inherent stability, and exact linear phase. However, their implementation cost is high due to large number of arithmetic operations involved. In direct-form linear phase FIR filters, the number of multipliers and adders is equal to the length of impulse responses. Computational cost can be minimized if the number of adder/multiplier units is decreased, which can be achieved by reducing the number of non-zero coefficients. A sparse FIR filter contains zero coefficients, so that multipliers and adders corresponding to those zero coefficients can be eliminated, resulting in lower hardware cost and power consumption.

This chapter is organized as follows: Section 4.1 gives an introduction about sparse filter design methods, Section 4.2 describes the minimax objective function formulation and iterative shrinkage algorithm for the design of linear phase sparse FIR filters. Section 4.3 gives a brief description about the constrained ABC-AMR algorithm; simulation results are described in Section 4.4 and conclusions are given in Section 4.5.

## *4.1 Introduction*

Attempts to reduce the implementation cost includes interpolated and extrapolated FIR filter design techniques, and frequency response masking digital filter designs etc.

In interpolated FIR filter design method [40], filter structure is cascaded into two sections, in which first section generates a sparse set of impulse response values with every $L$th sample being non zero and second section uses interpolation technique to generate rest of the samples. Interpolated finite impulse response filter requires $1/L$ of the multipliers and adders of an equivalent FIR filter. Also, two sections can be iteratively [41] designed using Remez multiple exchange algorithm until the difference between successive stages are within the given tolerance limits. Frequency masking technique [44]-[45], is used to design filters with very narrow transition band width. The basic principle is as follows; each delay element of the prototype filter is replaced by a given number of delays resulting in a filter with periodic passbands and very sharp transition bands, and a masking filter is used to extract desired band. The extrapolated filter design techniques [42],[46] utilize a quasi-periodic nature of impulse response of the FIR filter, which consists of a center lobe with the largest magnitude and decreasing side lobes away from the center. Approximating smaller magnitude side lobe as a scaled version of another side lobe will produce only small degradation in frequency response. Linear programming approach is used to optimize scale factors of the side lobe.

Due to advancements in sparse representation, design of sparse filters has gained increasing attention in recent years. Initially, sparsity of the filter is evaluated as a highly non-convex $l_0$ norm of filter coefficient vector. Finding a global optimum for such kind of problems are difficult, and an exhaustive search is required for optimal sparse coefficients which increases the computational complexity for higher order filters. In order to overcome non-convexity in design problems, an iterative design algorithm based on weighted least squares approach is proposed in [25]-[26]. Non-convex problem is successively transformed into a series of constrained sub-problems and these sub-problems are solved using successive iteration algorithm [27]. When linear programming is used for sparse filter design [24], initially, the impulse responses of a non-sparse filter is iteratively thinned until the frequency constraints are violated, and then the impulse responses of the filter are minimized using $l_1$ norm, and finally fixes the coefficients that should be constrained to zero in the following iterations. Sparse filter design problem can be formulated as a quadratic constrained problem with the following constraints: weighted least squares constraint on frequency response; constraint on mean squared error estimation; and constraint on signal to noise ratio in detection. The quadratic constraints can be either combined with low complexity backward selection algorithm [28] or exact algorithms based on branch and bound combinatorial optimization procedure [29]. A joint optimization approach optimizing both the filter order and sparsity is given in [30], which balances the filtering performance and implementation efficiency. The objective function is formulated as weighted $l_0$ norm and iterative reweighted least squares (IRLS) algorithm is employed to solve the error function. In greedy algorithm, for linear phase sparse filter design [32], coefficients in which the middle value of its feasible range is closest to zero is set to zero, whereas all other coefficients are free to change. Discrete optimization approaches for sparse filter design is explained in [35]-[36]. The $l_0$ norm-based optimization process is non-convex and NP hard, and due to computational complexity, they are not used in the design of higher order sparse FIR filters. Alternatively, sparse filter design can be relaxed from $l_0$ norm to $l_1$ norm. In [34], a novel $l_1$ norm-based optimization is described, in which, instead of selecting all coefficients some of the insignificant non zero coefficients are chosen to be zero. Recent advancements in $l_0$ and $l_1$-based sparse filter design is described in [31].

With the advancement in the evolutionary computation, various bio-inspired algorithms like the cuckoo search [38]-[39], and the ABC [37] have been applied to sparse filter design. The ABC algorithm is a swarm-based metaheuristic search algorithm that iteratively improves the quality of a solution with respect to its fitness value. In this chapter, the constrained ABC-AMR algorithm is used to design sparse FIR filters, using $l_0$ norm optimization subject to design constraints. In constrained optimization problems, the Deb's tournament selection operator and probabilistic selection scheme is used to select feasible solutions [135]. In contrast to $l_1$ optimization-based methods, which take all coefficients into minimization, the proposed method keeps some of the significant coefficients unchanged and identifies locations of zero coefficients that need to be minimized.

## *4.2 Sparse FIR Filter Design*

For Type 1 linear phase FIR filters [1] of filter order $N$, impulse response consists of $(N + 1)$ coefficients, represented as,

$$\boldsymbol{h} = [h(0), h(1), h(2), \dots \dots, h(N)] \tag{4.1}$$

Due to even symmetry, impulse response can be stated as,

$$h_n = h_{N-n} \quad \text{for} \quad n = 0,1,2, \dots\dots\dots\dots\dots \frac{N-1}{2} \tag{4.2}$$

The set of distinct impulse responses **h** of a Type 1 linear phase FIR filter can be represented by a more compact coefficient vector $\boldsymbol{c}$ as,

$$\mathbf{c} = \left[c_0, c_1, c_2, \dots\dots c_{\frac{N}{2}}\right] \tag{4.3}$$

where vector **c** is even symmetric such that,

$$\begin{aligned} c_0 &= h_{\frac{N}{2}}; \\ c_n &= 2h_{\frac{N}{2}-n} = 2h_{\frac{N}{2}+n}, \quad n = 1,2, \dots\dots\dots\dots\dots \frac{N}{2} \end{aligned} \tag{4.4}$$

The frequency response of $N$th-order even symmetry linear phase FIR filter is given by,

$$\begin{aligned} H\left(e^{jw}\right) &= e^{-j\left(\frac{N}{2}\right)w}\left\{c_0 + \sum_{n=1}^{\frac{N}{2}} c_n \cos(nw)\right\} \\ &= e^{-j\left(\frac{N}{2}\right)w} A(w) \end{aligned} \tag{4.5}$$

The amplitude response $A(w)$ can be given by,

$$A(w) = \mathbf{c}^{\boldsymbol{T}}\, \mathbf{cos}(w) \tag{4.6}$$

where $\mathbf{cos}(w) = \left[1 \quad \cos(wT) \quad \cos(2wT) \quad \cdots \quad \cos\left(\frac{N}{2}wT\right)\right]^T$

Desired amplitude response of $A_d(w)$ for Type 1 linear phase lowpass FIR filter is given by,

$$A_d(w) = \begin{cases} 1 & 0 \le w \le w_p \\ 0 & w_s \le w \le \pi \end{cases} \tag{4.7}$$

where $w_p$ and $w_s$ are normalized cut off frequencies of passband and stopband respectively.

In the proposed approach for sparse filter design problem, the number of non-zero coefficients should be minimized while the amplitude responses $A(w)$ is chosen to be constrained within the desired passband and stopband tolerance given by,

$$\begin{gathered} 1-\delta_p \leq |E_{\boldsymbol{c}}(w_i)| \leq\ 1+\delta_p\ \ 0 \leq w_i \leq w_p \\ -\delta_s \leq |E_{\boldsymbol{c}}(w_i)| \leq\ \delta_s\ \ w_s \leq w_i \leq \pi; \\ w_i\ \in\ \Omega_I \end{gathered} \tag{4.8}$$

where $\Omega_I$ denotes union of passband and stopband frequencies of interest.

$$\begin{gathered} E_{\boldsymbol{c}}(w) = e_p(\mathbf{c}) + e_s(\boldsymbol{c}) \\ e_p(\mathbf{c}) = \left[\sum_{i=1}^{I_p} W_p(w_i)\big||A(\mathbf{c},w_i)| - A_d(w_i)\big|^p\right]^{\frac{1}{p}} \\ \text{for } W_p(w_i) \geq 0;\ 0 \leq w_i \leq w_p \\ e_s(\mathbf{c}) = \left[\sum_{i=1}^{I_s} W_s(w_i)\big||A(\mathbf{c},w_i)| - A_d(w_i)\big|^p\right]^{\frac{1}{p}} \\ \text{for } W_s(w_i) \geq 0; w_s \leq w_i \leq \pi \\ W_p(w_i) = W_s(w_i) = 1 \text{ for } w_i\ \in\ \Omega_I \end{gathered} \tag{4.9}$$

where $e_p(\mathbf{c})$ is the passband error response and $e_s(\mathbf{c})$ is the stopband error response and $\Omega_I$ denotes union of passband and stopband frequencies of interest.

The $l_0$ norm calculates the number of non-zero coefficients. Even though $l_0$ optimization procedure is non-convex, NP hard and difficult to obtain global optimum, $l_0$ norm has certain advantages. The $l_1$ norm does not guarantee an optimal solution in constrained optimization problems. Also, it is difficult to determine the locations of filter coefficients which can be set to 0. Iterative shrinkage algorithm combined with the constrained ABC-AMR, identifies the crucial locations for zero coefficients and increases sparsity of the designed filter. The proposed constrained ABC-AMR algorithm is shown to be effective in solving non-convex, non-separable multimodal functions.

### 4.2.1 Iterative Shrinkage Algorithm

Initially, two subsets $\boldsymbol{\mathcal{Z}}^{(t)}$ and $\boldsymbol{\mathcal{NZ}}^{(t)}$ are defined within the coefficient set $\boldsymbol{c}$,where $\boldsymbol{\mathcal{Z}}^{(t)}$ and $\boldsymbol{\mathcal{NZ}}^{(t)}$ are indices of zero coefficients and non-zero coefficients of $\boldsymbol{c}$ respectively. They are initialized as $\boldsymbol{\mathcal{Z}}^{0} = \{\emptyset\}$ and $\boldsymbol{\mathcal{NZ}}^{0} = \left\{0,1 \dots\dots\dots\dots\dots.\frac{N}{2}\right\}$.

In each iteration, the optimization algorithm is used to solve the below problem,

$$\begin{gathered}
\min_{\mathbf{c}} \left\|\boldsymbol{c}_{\tau}{}^{(t)}\right\|_0 \\
\text{s.t } 1-\delta_p \le |E_{\boldsymbol{c}}(w_i)| \le 1+\delta_p \quad 0 \le w_i \le w_p \\
-\delta_s \le |E_{\boldsymbol{c}}(w_i)| \le \delta_s \quad w_s \le w_i \le \pi \\
c_k = 0, \ \forall k \in \mathcal{Z}^{(t-1)} \\
w_i \in \Omega_I
\end{gathered} \tag{4.10}$$

where $\delta_p$, $\delta_s$ denotes the passband ripple and stopband ripple respectively, $\boldsymbol{c}_{\tau}{}^{(t)}$ represents the coefficients chosen for $l_0$ optimization in current iteration $\boldsymbol{c}_{\tau}{}^{(t)} = [c_{i1}, c_{i2} \dots\dots\dots\dots.. c_{i\tau}]$ where $\tau^t$ is defined as,

$$\tau^t = \{n \,|\boldsymbol{c}_n(t-1) < M(t-1)\} \tag{4.11}$$

where $M(t-1)$ is the coefficient threshold value updated in every iteration. For updating the value of $M(t)$, first a subset of $\boldsymbol{\mathcal{NZ}}^{(\boldsymbol{t})}$ is defined such that,

$$\mathbb{C}^t = \{n | m_1 \le |\boldsymbol{c}_n(t)| \le m_2\} \tag{4.12}$$

where $[m_1, m_2]$ is the search domain and $M(t)$ is calculated as,

$$M(t) = \begin{cases} \min|\mathbb{C}^t| & \text{if } \mathbb{C}^t \ne \emptyset \\ M(t-1) & \text{if } \mathbb{C}^t = \emptyset \end{cases} \tag{4.13}$$

After each iteration $\boldsymbol{\mathcal{Z}}^{(\mathrm{t})}$ and $\boldsymbol{\mathcal{NZ}}^{(\mathrm{t})}$ subset of the coefficient indices is defined by,

$$\begin{gathered} \boldsymbol{\mathcal{Z}}^{(t)} = \boldsymbol{\mathcal{Z}}^{(t-1)} \cup \{\tau \ |\boldsymbol{c}_{\tau}(t) = 0)\} \\ \boldsymbol{\mathcal{NZ}}^{(t)} = \boldsymbol{\mathcal{NZ}}^{(t-1)} - \boldsymbol{\mathcal{Z}}^{(t)} \end{gathered} \tag{4.14}$$

### *4.3 Constrained Artificial Bee Colony Algorithm*

Initially, the ABC algorithm was only applied to unconstrained problems but later it was extended to constrained optimization problems. The constrained ABC algorithm uses the Deb's rule instead of a greedy selection to choose the best solution in a feasible region and infeasible solutions are discarded based on their violation values. For the design of sparse filters, the constrained ABC algorithm [135] is combined with the ABC-AMR for faster convergence.

The algorithm starts with a random initial population according to equation 3.1. In the employed bee phase, each of the employed bee searches for good solutions in its vicinity according to equation 3.3. Instead of a greedy selection in the original ABC algorithm, in the constrained ABC, the best solution is selected between the current solution, $\boldsymbol{x}_i$ and new solution, $\boldsymbol{x}_k$ according the Deb's rule [135],

$$\boldsymbol{x}_i = \begin{cases} \boldsymbol{x}_i & \text{if } \left(\gamma_{\boldsymbol{x}_i} \leq 0\right) \wedge \left(\gamma_{\boldsymbol{x}_k} > 0\right) \\ \boldsymbol{x}_k & \text{if } \left(\gamma_{\boldsymbol{x}_i} \leq 0\right) \wedge \left(\gamma_{\boldsymbol{x}_k} \leq 0\right) \wedge (f_k < f_i) \\ \boldsymbol{x}_i & \text{if } (\gamma_{\boldsymbol{x}_i} > 0) \wedge (\gamma_{\boldsymbol{x}_k} > 0) \wedge (\gamma_{\boldsymbol{x}_i} < (\gamma_{\boldsymbol{x}_k}) \end{cases} \tag{4.15}$$

where $\gamma_{\boldsymbol{x}_i}, \gamma_{\boldsymbol{x}_k}$ are the violation index values of $\boldsymbol{x}_i$ and $\boldsymbol{x}_k$; $f_i, f_k$ are the objective function values for the solutions $\boldsymbol{x}_i$ and $\boldsymbol{x}_k$, respectively. The fitness value of food locations is determined by equation 2.4.

The onlooker bees select food sources according to their probability values. The probability of the food sources can be calculated as follows,

$$p_i = \begin{cases} 0.5 + \left(\frac{fit_i}{\sum_{k=1}^{SN} fit_k}\right) \times 0.5 & \text{if solution is feasible} \\ \left(1 - \frac{\gamma_i}{\sum_{k=1}^{SN} \gamma_k}\right) \times 0.5 & \text{otherwise} \end{cases} \tag{4.16}$$

where $fit_i$ and $fit_k$ are the fitness value of $\boldsymbol{x}_i$ and $\boldsymbol{x}_k$, respectively. Solutions are selected proportional to their fitness value and inversely proportional to violation values. Similar to the employed bees, the onlooker bees produce new food locations by applying equation 3.3.

A solution which cannot be improved after predetermined trials becomes a scout bee and is abandoned. The scout bees will then randomly search for a new solution.

### *4.4 Design Examples and Results*

The constrained ABC-AMR algorithm is combined with iterative shrinkage algorithm to design sparse Type 1 linear phase lowpass FIR filters and results are compared with other design methods in literature such as the minimum-increase method [24], and the smallest coefficient method [24] and the partial $l_1$ optimization [33].

The initial coefficients are set using Parks-McClellan (PM) algorithm. The number of food locations $SN$ is set as 50 and $p$ is set as 128. The constraints set using passband ripple $\delta_p$, and stopband ripple $\delta_s$. Table 4.1 summarizes the design specifications; sparse filters are designed for orders $N = 60, 70, 80$. Maximum passband attenuation is $\pm 0.5$ dB and minimal stopband attenuation is set as 60 dB. The designs are performed using intel core i7-4790, 3.60 GHz with 12GB RAM desktop computer.

**Table 4.1**. Sparse FIR Lowpass Filter Specification

| | |
|---|---|
| Passband region | [0, 0.3π] |
| Stopband region | [0.5 π, π] |
| Filter order $N$ | 60,70,80 |
| Maximum passband attenuation $R_p$ | Within ±0.5dB of unity |
| $[m_1, m_2]$ | $[10^{-6}, 10^{-3}]$ |
| Minimum stopband attenuation $R_s$ | 60 dB |

The constraints are set using passband ripple $\delta_p$, and stopband ripple $\delta_s$. The maximum passband attenuation $R_p$ (in dB) is related to the passband ripple $\delta_p$ as,

$$\delta_p = \frac{10^{R_p/20} - 1}{10^{R_p/20} + 1} \tag{4.17}$$

The minimum stopband attenuation $R_s$ (in dB) is related to the stopband ripple $\delta_s$ as,

$$\delta_s = 10^{-R_s/20} \tag{4.18}$$

### 4.4.1 Sparse FIR Filter of Order $N = 60$

In this example, sparse FIR filter of order $N = 60$, is designed using iterative shrinkage algorithm and the constrained ABC-AMR algorithm. In Table 4.2, passband peak error and stopband peak error is compared with the results obtained from the minimum-increase method [24], and the smallest coefficient method [24] and the partial $l_1$ optimization [33]. From Table 4.2, for the same sparsity constrained ABC-AMR can achieve better passband and stopband errors compared to other design methods.

**Table 4.2.** Peak Error Results of Sparse FIR Filter of Order $N = 60$

| Alg. | Passband Peak Error | Stopband Peak Error |
|---|---|---|
| Constrained ABC-AMR | 0.028201403241469 | 4.491284356122893e-04 |
| Minimum-increase [24] | 0.027972285993324 | 2.517087760104883e-04 |
| Smallest coefficient [24] | 0.028222427657121 | 5.485674097077593e-04 |
| Partial $l_1$ optimization [33] | 0.028222427657120 | 5.485674097078625e-04 |

The minimum coefficient values and the number of zero coefficients of sparse FIR filter of order $N = 60$ are given in Table 4.3, constrained ABC-AMR has the lowest coefficient value compared to other design methods.

**Table 4.3** Minimum Coefficient Value of Sparse FIR Filter of Order $N = 60$

| Alg. | Sparsity | Minimum coefficient value |
|---|---|---|
| Constrained ABC-AMR | 32 | 0 |
| Minimum-increase [24] | 30 | 1.005784065934075e-18 |
| Smallest coefficient [24] | 32 | 1.204953400396407e-17 |
| Partial $l_1$ optimization [33] | 32 | 1.195249450476534e-17 |

The plots for magnitude response, impulse response, passband and stopband errors of sparse FIR filter of order $N = 60$ obtained using constrained ABC-AMR is shown in Figure 4.1

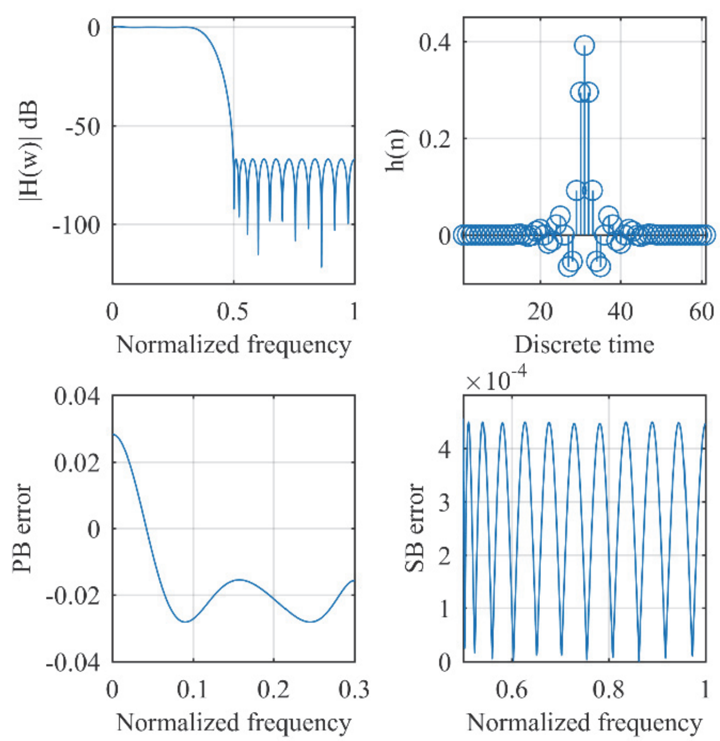


**Figure 4.1** Magnitude Response, Impulse Response, Passband and Stopband Errors of Sparse FIR Filter of Order $N = 60$

The enlarged impulse responses obtained using constrained ABC-AMR, minimum increase and partial $l_1$ optimization is shown in Figure 4.2.

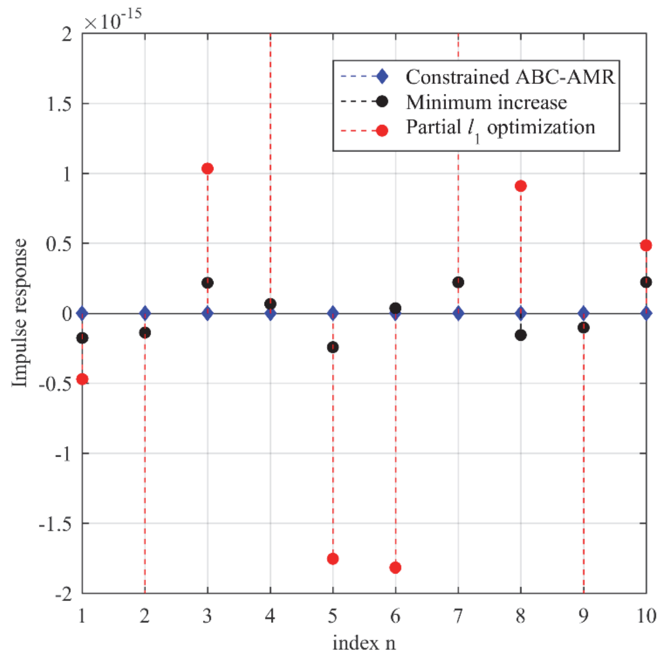


**Figure 4.2** Enlarged Impulse Response of Sparse FIR Filter of Order $N = 60$

### *4.4.2 Sparse FIR Filter of Order N = 70*

In this example, sparse FIR filter of order $N = 70$, is designed using iterative shrinkage algorithm and the constrained ABC-AMR algorithm. In Table 4.4, passband peak error and stopband peak error are compared with results obtained from the minimum-increase method [24], and the smallest coefficient method [24] and the partial $l_1$ optimization [33]. From Table 4.4, for the same sparsity constrained ABC-AMR can achieve similar or better passband and stopband errors compared to other design methods.

**Table 4.4**. Peak Error Results of Sparse FIR Filter of Order $N = 70$

| Alg. | Passband Peak Error | Stopband Peak Error |
|---|---|---|
| Constrained ABC-AMR | 0.028222468513358 | 4.480700642792157e-04 |
| Minimum-increase [24] | 0.027972285993374 | 2.517087760109814e-04 |
| Smallest coefficient [24] | 0.028222427657520 | 5.485674097059533e-04 |
| Partial $l_1$ optimization [33] | 0.028222427657520 | 5.485674097057302e-04 |

The minimum coefficient values and the numbers of zero coefficients of sparse FIR filter of order $N = 70$ are given in Table 4.5, constrained ABC-AMR has the lowest coefficient value compared to other design methods.

**Table 4.5** Minimum Coefficient Value of Sparse FIR Filter of Order $N = 70$

| Alg. | Sparsity | Minimum coefficient value |
|---|---|---|
| Constrained ABC-AMR | 42 | 0 |
| Minimum-increase [24] | 40 | 4.799699851515894e-19 |
| Smallest coefficient [24] | 42 | 7.173038352436877e-18 |
| Partial $l_1$ optimization [33] | 42 | 7.248318064152319e-18 |

Figure 4.3 shows the plots for magnitude response, impulse response, passband and stopband errors of magnitude response for sparse filter for an order $N = 70$, obtained using constrained ABC-AMR algorithm.

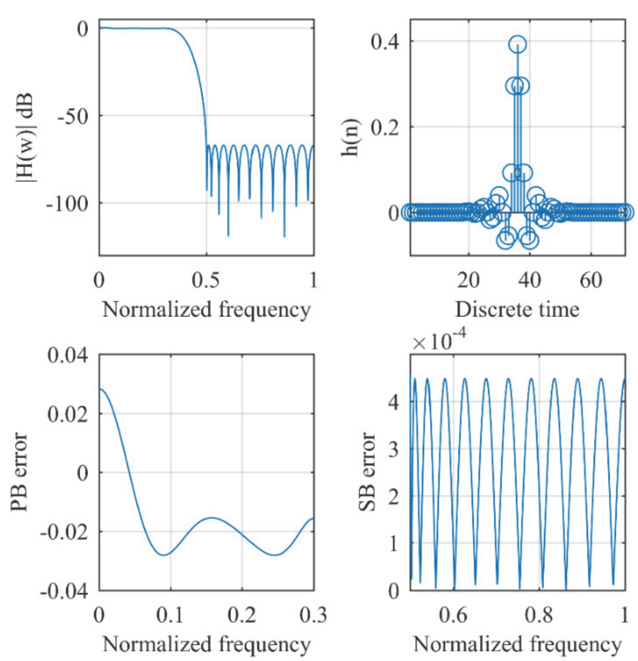


**Figure 4.3** Magnitude Response, Impulse Response, Passband and Stopband Errors of Sparse FIR Filter of Order $N = 70$

The enlarged impulse responses obtained using constrained ABC-AMR, minimum increase and partial $l_1$ optimization is shown in Figure 4.4.

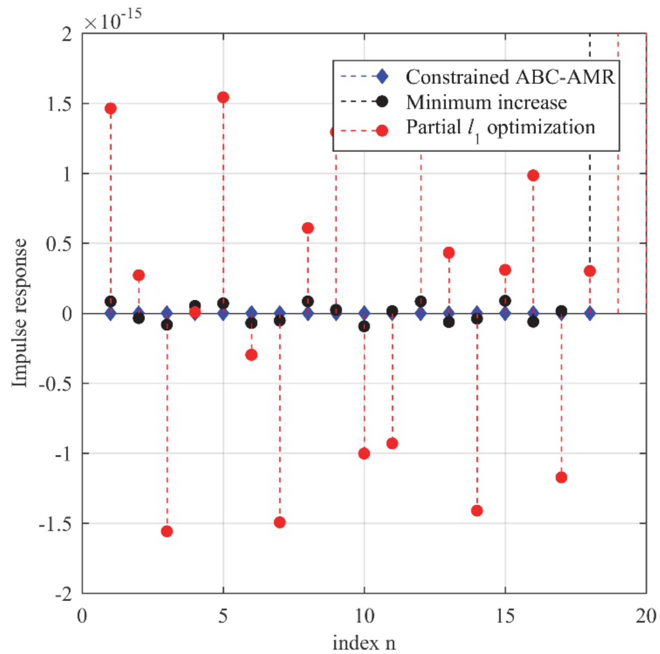


**Figure 4.4** Enlarged Impulse Response of Sparse FIR Filter of Order $N = 70$

### *4.4.3 Sparse FIR Filter of Order $N = 80$*

In this example, sparse FIR filter of order $N = 80$, is designed using the iterative shrinkage algorithm and the constrained ABC-AMR algorithm. In Table 4.6, passband peak error and stopband peak error are compared with the results obtained from the minimum-increase method [24], and the smallest coefficient method [24] and the partial $l_1$ optimization [33]. From Table 4.6, for the same sparsity constrained ABC-AMR can achieve similar or better passband and stopband errors compared to other design methods.

**Table 4.6**. Peak Error Results of Sparse FIR Filter of Order $N = 80$

| Alg. | Passband Peak Error | Stopband peak Error |
|---|---|---|
| Constrained ABC-AMR | 0.028705964702497 | 9.329212691181199e-04 |
| Minimum-increase [24] | 0.027971448361442 | 1.970800923940808e-04 |
| Smallest coefficient [24] | 0.028222427657755 | 4.480593261947385e-04 |
| Partial $l_1$ optimization [33] | 0.028706632647080 | 9.318344341936180e-04 |

The minimum coefficient values and the number of zero coefficients of sparse FIR filter of order $N = 80$ are given in Table 4.7, constrained ABC-AMR has the lowest coefficient value compared to other design methods.

**Table 4.7** Minimum Coefficient Value of Sparse FIR Filter of Order $N = 80$

| Alg. | Sparsity | Minimum coefficient value |
|---|---|---|
| Constrained ABC-AMR | 56 | 0 |
| Minimum-increase [24] | 50 | 3.597447722312056e-17 |
| Smallest coefficient [24] | 52 | 7.980770973682628e-18 |
| Partial $l_1$ optimization [33] | 56 | 6.327944906487736e-17 |

Figure 4.5 shows the plots for magnitude response, impulse response, passband and stopband errors of sparse FIR filter order $N = 80$ obtained using constrained ABC-AMR.

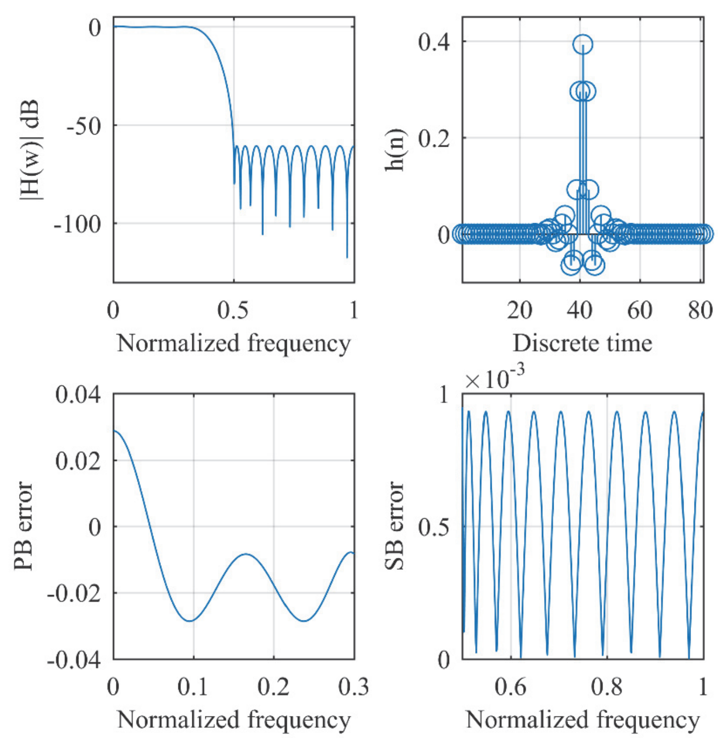


**Figure 4.5** Magnitude Response, Impulse Response, Passband and Stopband Errors of Sparse FIR Filter of Order $N = 80$

The enlarged impulse responses obtained using constrained ABC-AMR, minimum increase and partial $l_1$optimization is shown in Figure 4.6.

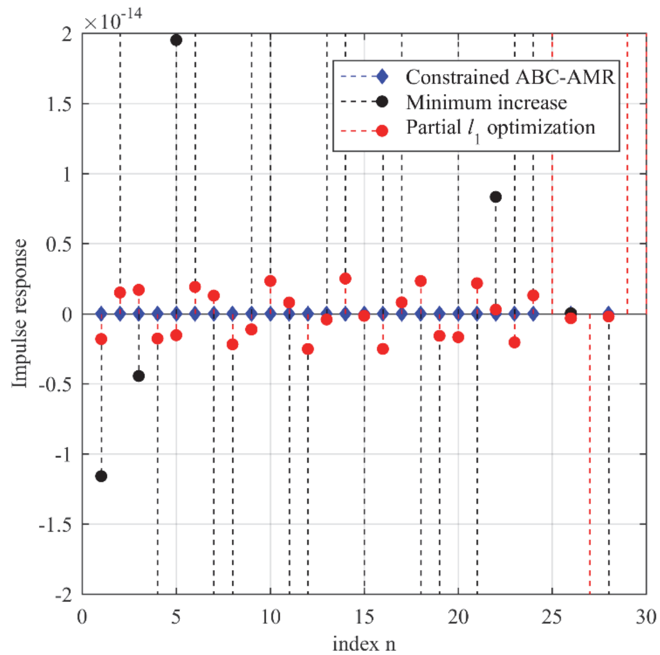


**Figure 4.6** Enlarged Impulse Response of Sparse FIR filter of Order $N = 80$

The sparse filter coefficients obtained using the combined iterative shrinkage approach and the constrained ABC-AMR method is listed in Table 4.8.

**Table 4.8** Filter Coefficients of Sparse FIR Filter of Filter Order $N = 60,70,80$

| $h(n)$ | Filter order $N = 60$ | Filter order $N = 70$ | Filter order $N = 80$ |
|---|---|---|---|
| $h(0)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(1)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(2)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(3)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(4)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(5)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(6)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(7)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(8)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(9)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(10)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(11)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(12)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(13)$ | 0.001229760122581 | 0.000000000000000 | 0.000000000000000 |
| $h(14)$ | 0.001957628258966 | 0.000000000000000 | 0.000000000000000 |
| $h(15)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(16)$ | -0.003116100726042 | 0.000000000000000 | 0.000000000000000 |
| $h(17)$ | -0.001048710518735 | 0.000000000000000 | 0.000000000000000 |
| $h(18)$ | 0.007693412368413 | 0.001229690709782 | 0.000000000000000 |
| $h(19)$ | 0.011894709063846 | 0.001957182197884 | 0.000000000000000 |
| $h(20)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(21)$ | -0.017303208740767 | -0.003113973466924 | 0.000000000000000 |
| $h(22)$ | -0.011851694043208 | -0.001044575239062 | 0.000000000000000 |
| $h(23)$ | 0.021051207067630 | 0.007697807996681 | 0.000000000000000 |
| $h(24)$ | 0.039220288229915 | 0.011897609370965 | 0.000974263193208 |
| $h(25)$ | 0.000000000000000 | 0.000000000000000 | 0.000000000000000 |
| $h(26)$ | -0.064915315929322 | -0.017303600595197 | -0.001818077007100 |
| $h(27)$ | -0.054062389098616 | -0.011851953083718 | 0.000000000000000 |
| $h(28)$ | 0.092252992041992 | 0.021051322404193 | 0.007238345709738 |
| $h(29)$ | 0.295399045895760 | 0.039220579689895 | 0.010783492850610 |
| $h(30)$ | 0.391392195116046 | 0.000000000000000 | 0.000000000000000 |
| $h(31)$ | | -0.064915414757148 | -0.016104522528231 |
| $h(32)$ | | -0.054062286944487 | -0.011133059084284 |
| $h(33)$ | | 0.092253213971575 | 0.020407743523944 |
| $h(34)$ | | 0.295399421405326 | 0.038315167841337 |
| $h(35)$ | | 0.391392421193828 | 0.000000000000000 |
| $h(36)$ | | | -0.064510594576277 |
| $h(37)$ | | | -0.054215769916657 |
| $h(38)$ | | | 0.091974683046617 |
| $h(39)$ | | | 0.296072720787263 |
| $h(40)$ | | | 0.392736793244963 |

### *4.5 Conclusions*

In this chapter, the constrained ABC-AMR algorithm has been used together with the iterative shrinkage algorithm to design minimax sparse linear phase FIR lowpass filters. As far as the design constraints are not violated, the coefficient values at certain crucial locations can be set as zero. The design results obtained is compared with other design methods such as the partial $l_1$ optimization [33], the minimum-increase method [24], the smallest coefficient method [24]. Although other design algorithms can decrease coefficient values, the constrained ABC-AMR algorithm can reduce impulse responses at insignificant locations to zero. From the peak error comparison tables, the ABC-AMR algorithm demonstrates similar or better passband and stopband errors compared to other design methods.

# CHAPTER 5

# MULTIOBJECTIVE APPROACH FOR ASYMMETRIC FIR FILTER DESIGN

Linear phase FIR filters with symmetric/antisymmetric impulse responses are characterized by long and fixed delay which is undesirable for some applications. Asymmetric FIR filter design is useful due to the following reasons:

i. The long delay associated with a linear phase FIR filter can be removed through replacing the linear phase requirement by an approximation of linear phase requirement in the passband.
ii. A nonlinear phase FIR filter can be designed to simultaneously approximate desired magnitude and group delay responses.

In this chapter, the multiobjective ABC algorithm is used to design asymmetric FIR filters to satisfy simultaneously desired magnitude response and group delay response. In the proposed method, preferences are set *a priori* using a reference point approach.

The chapter is organized as follows: Section 5.1 gives an introduction about asymmetric FIR filter design methods; Section 5.2 describes asymmetric FIR filter design problem, Section 5.3 gives a description about the reference-point-based MOABC, design results are given in Section 5.4 and conclusions are discussed in Section 5.5.

## *5.1 Introduction*

A literature survey shows that FIR filters with asymmetric filter coefficients can be designed using several optimization methods. FIR filters with arbitrary magnitude and phase responses can be designed using iterative reweighted least squares algorithm [55] by a mixed use of least squares ($L_2$), and Chebyshev ($L_\infty$) optimization algorithms. The absolute mean square error between frequency responses of deigned filter and desired filter can be formulated as a quadratic function and can be solved using a set of linear equations to obtain the optimal filter coefficients [54]. The digital filter with specified stopband gains and total stopband energy can be obtained using least squares approach with the stopband subjected to maximum gain constraints [47]. Nonlinear optimization algorithm based on the iterative use of the generalized multiple exchange (GME) can be utilized to design optimal filters that simultaneously meet group delay and magnitude responses specifications [48]. The optimization of complex error function, for the design of complex frequency response FIR filter, can be performed according to the $L_2$ norm subject to inequality constraints [49]. Iterative algorithm starting from the Kuhn-Tucker optimality conditions is used to solve a system of nonlinear equations.

Most of the above mentioned asymmetric FIR filter design methods use a combination of least square ($L_2$) and Chebyshev ($L_\infty$) norms for all frequency bands. $L_2$ norm is easy to compute and frequently used for various signal processing applications, but they produce large errors near the discontinuities between two desired responses. The total energy of the aliased signals must be minimized if an input signal spectrum is wideband and uniformly distributed. If the passband is narrow, energy can be aliased from wide stopband to narrow passbands. Even though, the aliased energy can be minimized using least-squares criterion ($L_2$), the designed filters will have large gains near the edge of their stopband, otherwise known as the Gibbs phenomenon. Also, $L_\infty$ norm minimizes the amplitude distortion in passband but fails to optimize the gain and total energy in stopband.

Evolutionary algorithms are capable of handling complex, multimodal design problems, and hence they can be incorporated into multiobjective optimization problems. Elitist nondominated sorting genetic algorithm (NSGA) is used to design FIR filters of predefined amplitude response and group delay characteristics [56]. Three different objective functions based on passband and stopband amplitude response error, and group delay error are used. Bio-inspired algorithms such as the ABC algorithm [50], the cuckoo search algorithm (CSA) [51], the teaching learning-based optimization (TLBO) [52], the harmony search algorithm (HSA) [59] and iterative self-learning algorithm [60] can be used in the design of asymmetric FIR filters. In this chapter, the multiobjective ABC algorithm is used to design asymmetric FIR filters and the design results are compared to the multiobjective differential evolution.

## 5.2 *Asymmetric FIR Filter Design [1]*

A $N$th-order asymmetric FIR filter consists of $(N+1)$ impulse responses and can be represented by a distinct coefficient vector **c** as,

$$\mathbf{c} = [c_0, c_1, c_2, c_3, \dots, c_N]^{\boldsymbol{T}} \tag{5.1}$$

The frequency response $H(w)$ of a FIR filter can be expressed as,

$$\begin{aligned} H(w) &= \sum\nolimits_{n=0}^{N} c_n z^{-n} \big|_{z=e^{jwT}} \\ &= \sum_{n=0}^{N} c_n \cos(\omega nT) - j \sum_{n=0}^{N} c_n \sin(\omega nT) \\ &= |\, H(\omega)| e^{j\theta(w)} \end{aligned} \tag{5.2}$$

In equation 5.2, the magnitude response $|H(w)|$is equal to,

$$|H(w)| = \left\{ \left[ \sum\nolimits_{n=0}^{N} c_n \cos nwT \right]^2 + \left[ \sum\nolimits_{n=0}^{N} c_n \sin nwT \right]^2 \right\}^{\frac{1}{2}} \tag{5.3}$$

and the group delay of asymmetric FIR filter can be expressed as,

$$\tau(w) = -\frac{\partial \theta(w)}{\partial wT} \tag{5.4}$$

where $\theta(w)$ is the phase response.

The objective function of the magnitude error $e_{mag}(\mathbf{c})$ is defined by,

$$e_{mag}(\mathbf{c}) = \left[ \sum\nolimits_{i=1}^{I} W(w_i) \big| |H(w)| - |H_d(w_i)| \big|^p \right]^{\frac{1}{P}} \quad \text{for } \forall w_i \in \Omega_I \tag{5.5}$$

where $p$ is a positive even integer; $H_d(w_i) = 1$ in passband and $H_d(w_i) = 0$ in stopband(s); and $\Omega_I$ denotes the union of frequency bands of interest.

Similarly, the objective function of the group delay error $e_{gd}(\mathrm{c})$ among the passband can be calculated by,

$$e_{gd}(\mathbf{c}) = \left[\sum_{i=1}^{I} W(w_i)\big||\tau(w)| - |\tau_d(w_i)|\big|^p\right]^{\frac{1}{P}} \quad \text{for } w_{p1} \leq w_i \leq w_{p2} \tag{5.6}$$

where $p$ is a positive even integer; $\tau_d(w)$ is the desired group delay.

The optimization problem for designing digital FIR filters searches for an optimal coefficient vector $\mathbf{c}$ that minimizes the minimax errors in the magnitude and group delay responses simultaneously.

### *5.3 Reference Point-Based Multiobjective ABC Algorithm*

A reference point method represents a preference-based multiobjective optimization approach which directs the search towards the region of interest of the decision maker. The preference-based methods have various advantages such as computationally efficiency, faster convergence and better scalability in higher objective space. One of the earliest approaches in the reference point method is described in [186], in which the optimal solutions near the reference point $\bar{\mathbf{z}} = (\bar{z}_1, \bar{z}_2, \dots \dots .. \bar{z}_m)$, is obtained by solving the scalarizing function $s(f(\boldsymbol{x}))$,

$$s(f(\boldsymbol{x})) = \max_{i=1,2..m} [w_i . (f_i(\boldsymbol{x}) - \bar{\mathbf{z}}_i)] \tag{5.7}$$

where $w_i$ is the weighting vector and $f_i$ is objective function values. The reference point $\bar{\mathbf{z}}_i$ guides the search toward the desired region while weight vector $w_i$ provides more detailed information about the Pareto optimal point. The main drawback of this approach is that the problem is formulated as single-objective optimization problem and provides only one solution in each run according to the decision maker's preferences. If the user is dissatisfied with the current solution and procedure is repeated with a new set of reference points. In practice, several runs of algorithm are required to reach the optimal solutions.

In the reference point dominance-based approach, solutions near the reference point is preferred while preserving the order induced by the Pareto dominance. The user provides a set of reference points and for each reference point the normalization Euclidean distance known as the preference operator [176] is calculated as,

$$d_{IR} = \sqrt{\sum_{i=1}^{M} \left( \frac{f_i(\boldsymbol{x}) - \boldsymbol{R}_i}{f_i^{max} - f_i^{min}} \right)^2} \tag{5.8}$$

where $d_{IR}$ is the normalization Euclidean distance or the preference operator from individual $\boldsymbol{I}$ to reference point $\boldsymbol{R}$; $M$ is the number of objectives; $f_i^{max}$ and $f_i^{min}$ are the population maximum and minimum objective value of $i$th objective. The solutions near the reference point is assigned rank 1, next closet solution is given rank 2 and so on. The flowchart for the reference point-based MOABC is given in Figure 5.1.

When the reference point-based MOABC is used to design asymmetric FIR filters, the optimization process search for an optimal coefficient that minimizes both the amplitude response error and group delay error. The initial population is generated from a uniform random distribution within the upper and lower limit specified in Table 5.1. The objective function value is calculated for each food location and nondominated sorting is performed, finally ranks are assigned based on the preference operator $d_{IR}$. During each iteration $t$, onlooker and employed bees are updated according to equation 3.3.

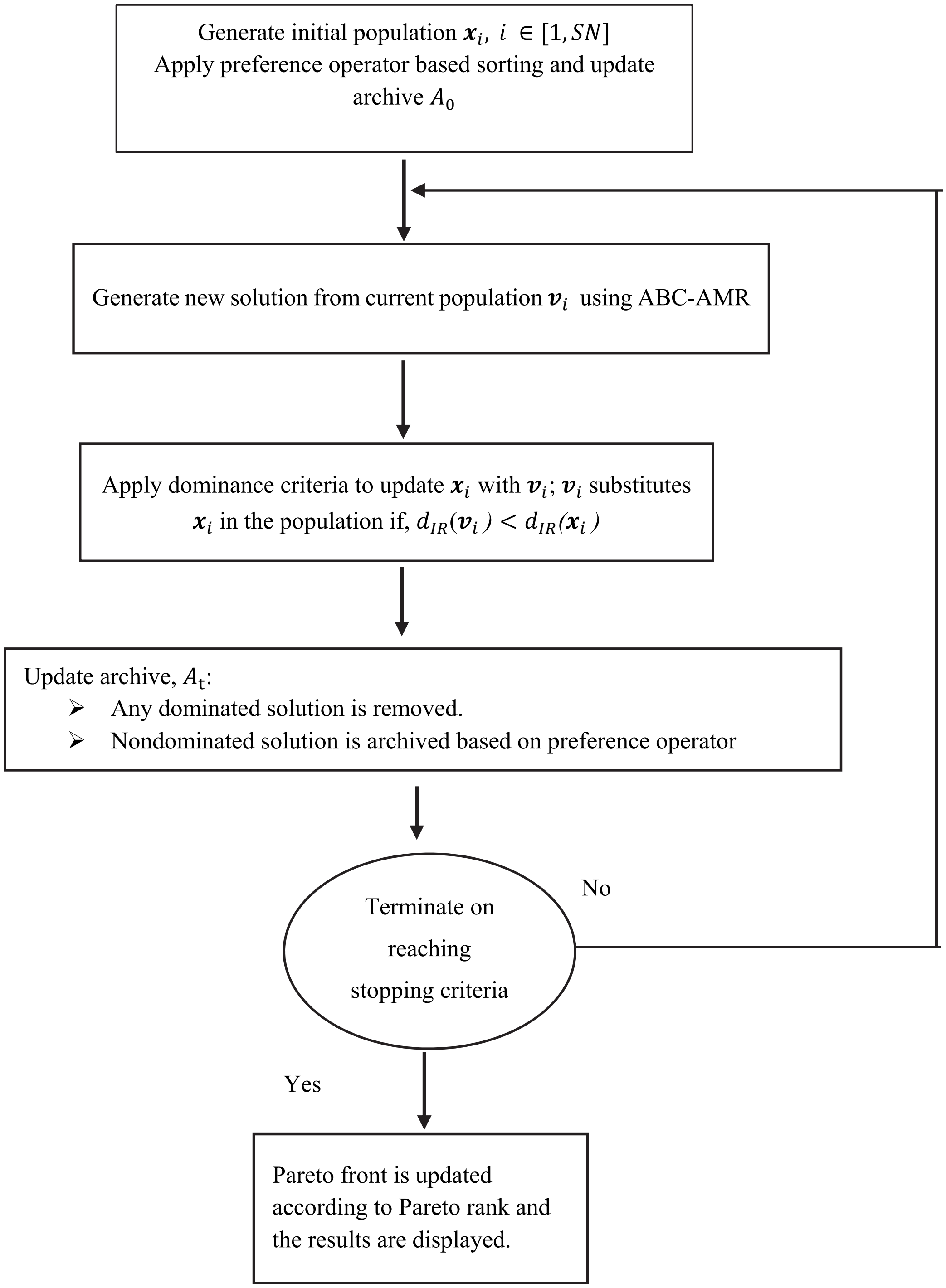


**Figure 5.1** Flowchart of Reference Point-Based MOABC

The new solution is evaluated against old solution based on its distance from the reference point,

$$\boldsymbol{x}^i\Big|_t = \begin{cases} \boldsymbol{x}^i\Big|_t & if \quad d_{IR}(\boldsymbol{v}^i\Big|_t) > d_{IR}(\boldsymbol{x}^i\Big|_t) \\ \boldsymbol{v}^i\Big|_t & if \quad d_{IR}(\boldsymbol{v}^i\Big|_t) < d_{IR}(\boldsymbol{x}^i\Big|_t) \end{cases} \tag{5.9}$$

The onlooker bee searches for new food location based on the probability value calculated using equation 2.5 and the employed bees that cannot be improved after predetermined number of trials is abandoned as scout bees. The size of the archive is maintained within the predefined limit using a preference vector operator. On reaching termination criteria, the top solutions are chosen according to their ranks.

### *5.4 Design Examples and Results*

In this section, asymmetric FIR filters of order 24 and group delay value of 10 are designed using the reference point-based MOABC. The optimization algorithm simultaneously optimizes amplitude error response and group delay error response.

The filter coefficients are initialized randomly, and the results are compared to the reference point-based multiobjective differential evolution algorithm, the obtained results are favorable to MOABC. A detailed description about the multiobjective differential evolution algorithm (MODE) is given in [188].

The ABC parameters such as the population size $SN = 50$, and limit is set as $200$. The parameters for the MOABC and the MODE are given in Table 5.1; frequency grid for optimization, error calculation, and weights for optimization are given in Table 5.2; filter specifications are given in Table 5.3; Reference points are set using the tolerance limits in passband magnitude, stopband magnitude and peak group delay. The designs are performed using an intel core i7-4790, 3.60 GHz with 12GB RAM desktop computer.

**Table 5.1** Parameters of MOABC and MODE

| Symbol | Description | Lowpass | Highpass | Bandpass | Bandstop |
|---|---|---|---|---|---|
| $p$ | Least $p$ th-order | 128 | 128 | 128 | 128 |
| $SN$ | Colony Size of ABC | 50 | 50 | 50 | 50 |
| $N_p$ | Population of DE | 50 | 50 | 50 | 50 |
| $F$ | Scaling factor of DE | 0.5 | 0.5 | 0.5 | 0.5 |
| $CR$ | Crossover rate of DE | 0.5 | 0.5 | 0.5 | 0.5 |

**Table 5.2** Frequency Grids for Asymmetric FIR Filter Design

| | |
|---|---|
| Optimization | $F_0 = [0.0.005{:}\,1]$ |
| Peak error evaluation | $F_e = [0.0.001{:}\,1]$ |
| Frequency weights for $0 \le w_i \le \pi$, $W(w_i)$ | 1 |

**Table 5.3** Asymmetric FIR Filter Specifications

| Symbol | Description | Lowpass | Highpass | Bandpass | Bandstop |
|---|---|---|---|---|---|
| $w_{p1}$ | Passband edge 1 | 0.3π | 0.55 π | 0.25π | 0.4 π |
| $w_{s1}$ | Stopband edge 1 | 0.4π | 0.45 π | 0.35π | 0.3 π |
| $w_{p2}$ | Passband edge 2 | - | - | 0.6π | 0.65π |
| $w_{s2}$ | Stopband edge 2 | - | - | 0.7π | 0.55π |
| $\delta_{s1}$ | Stopband 1 tolerance limit | 0.05 | 0.06 | 0.06 | 0.06 |
| $\delta_{s2}$ | Stopband 2 tolerance limit | - | | 0.06 | - |
| $\delta_{p1}$ | Passband 1 tolerance limit | 0.05 | 0.06 | 0.06 | 0.06 |
| $\delta_{p2}$ | Passband 2 tolerance limit | - | - | - | 0.06 |
| $\delta_g$ | Group delay tolerance limit | 0.005 | 0.006 | 0.006 | 0.006 |

### *5.4.1 Asymmetric FIR Lowpass Filter*

Asymmetric FIR lowpass filter of order 24 and group delay 10 is designed using the proposed reference-point-based MOABC approach and the results are compared to those of the reference-point-based MODE algorithm. The reference point is set as follows: $R_i = [\delta_{p1}, \delta_{s1}, \delta_g]$ according to Table 5.3. The three objective functions for minimax error approximation are formulated as follows,

$f_1(\mathrm{c})$ represents the minimax magnitude error in passband,

$$f_1(\mathrm{c}) = \left[\sum\nolimits_{i=1}^{\Omega_p} W(w_i)\big||H(w_i)| - 1\big|^{128}\right]^{\frac{1}{128}} \tag{5.10}$$

$f_2(\mathrm{c})$ represents the minimax magnitude error in stopband,

$$f_2(\mathrm{c}) = \left[\sum\nolimits_{i=\Omega_s}^{\mathrm{I}} W(w_i)\big||H(w_i)|\big|^{128}\right]^{\frac{1}{128}} \tag{5.11}$$

$f_3(\mathrm{c})$ represents the minimax group delay error in passband,

$$f_3(\mathrm{c}) = \left[\sum\nolimits_{i=1}^{I} W(w_i)\big||\tau(w_i)| - 10\big|^{128}\right]^{\frac{1}{128}} \tag{5.12}$$

where $W(w_i) = 1$ from equations 5.10 - 5.12.

Magnitude response obtained for asymmetric FIR lowpass filter using the multiobjective ABC and the multiobjective DE are plotted in Figure 5.2 and Figure 5.3. The Pareto front approximation for the multiobjective ABC and the multiobjective DE are plotted in Figure 5.4 and Figure 5.5. The upper and lower limits of passband, stopband and group delay errors of Pareto optimal solutions obtained using the MOABC and the MODE are given in Table 5.4. Unlike MODE, MOABC has a smaller range for each of the objective functions, as MOABC tries to improve the solutions near the reference point defined by the user instead of searching entire pareto front. This reduces the computational cost and complexity.

**Table 5.4** Objective Function Range for Asymmetric FIR Lowpass Filter

| | | MOABC | MODE |
|---|---|---|---|
| $Pe_{PB}$ | $e_{PB}^{U}$ | 0.052152117007106 | 0.054160533620728 |
| | $e_{PB}^{L}$ | 0.052001009935788 | 0.036827044396720 |
| $Pe_{SB}$ | $e_{SB}^{U}$ | 0.050737738312313 | 0.053503693033970 |
| | $e_{SB}^{L}$ | 0.050673982376137 | 0.037126763296944 |
| $Pe_{GD}$ | $e_{GD}^{U}$ | 0.004373960888387 | 0.626099033120376 |
| | $e_{GD}^{L}$ | 0.004318531058836 | 0.004165909983975 |

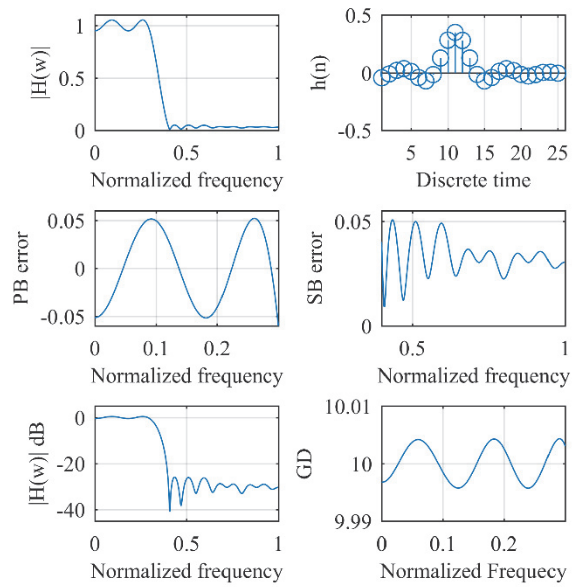


**Figure 5.2** Magnitude Response, Impulse Response, Passband and Stopband Errors of Asymmetric FIR Lowpass Filter Using MOABC

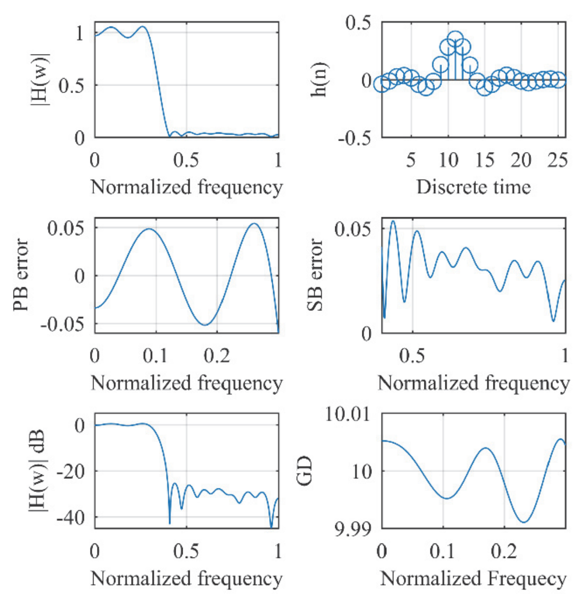


**Figure 5.3** Magnitude Response, Impulse Response, Passband and Stopband Errors of Asymmetric FIR Lowpass Filter Using MODE

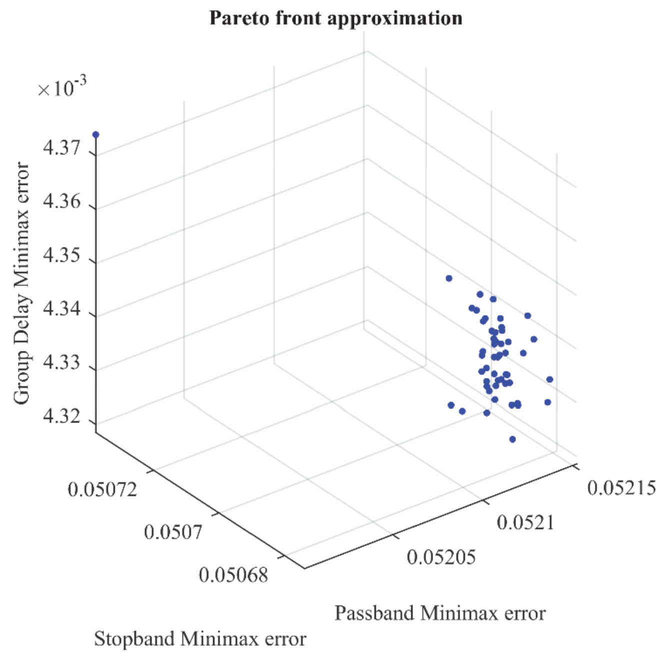


**Figure 5.4** Pareto Front Approximation of Asymmetric FIR Lowpass Filter Using MOABC

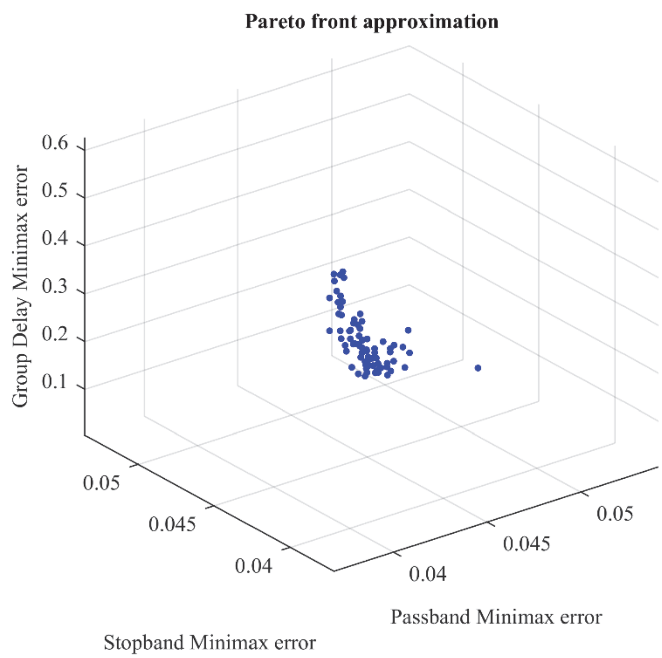


**Figure 5.5** Pareto Front Approximation of Asymmetric FIR Lowpass Filter Using MODE

The peak error values from two designs are compared in Table 5.5. Simulation results indicate that FIR lowpass filter of group delay 10 designed using the multiobjective ABC algorithm has lower passband peak error, stopband peak error and group delay error in comparison to those obtained by the multiobjective DE designs. Filter coefficients of asymmetric FIR lowpass filter of order $N = 24$ and group delay 10 obtained using the MOABC and the MODE are listed in Table 5.6

**Table 5.5** Peak Error Values of Asymmetric FIR Lowpass Filter

| | MOABC | MODE |
|---|---|---|
| Passband peak error | 0.052138791166861 | 0.054107564642413 |
| Stopband peak error | 0.050686147474734 | 0.053503693033970 |
| Group delay error | 0.004320014372624 | 0.005498552713419 |

**Table 5.6** Coefficients of Asymmetric FIR Lowpass Filter

| $h(n)$ | MOABC | MODE |
|---|---|---|
| $h(0)$ | -0.042716129862988 | -0.039669318958197 |
| $h(1)$ | -0.009532115856082 | -0.012020458958296 |
| $h(2)$ | 0.021065732735079 | 0.024931762696377 |
| $h(3)$ | 0.036477433192541 | 0.035650920233609 |
| $h(4)$ | 0.011651438615107 | 0.014025902815955 |
| $h(5)$ | -0.041217814823204 | -0.041809354025453 |
| $h(6)$ | -0.070303617460113 | -0.069082250031345 |
| $h(7)$ | -0.015171539449710 | -0.015559591138835 |
| $h(8)$ | 0.128282573784506 | 0.130469056416274 |
| $h(9)$ | 0.281875692201088 | 0.280708282808087 |
| $h(10)$ | 0.349015124922104 | 0.349929433024378 |
| $h(11)$ | 0.279978387027116 | 0.282293026185234 |
| $h(12)$ | 0.128396581731314 | 0.126723515153492 |
| $h(13)$ | -0.013858102405083 | -0.012338432392482 |
| $h(14)$ | -0.070969455028233 | -0.070178727886812 |
| $h(15)$ | -0.040927384153750 | -0.041058175666972 |
| $h(16)$ | 0.012395293905058 | 0.013442684705968 |
| $h(17)$ | 0.036111131300656 | 0.039042561037000 |
| $h(18)$ | 0.019695789414305 | 0.019116744091872 |
| $h(19)$ | -0.013481170730628 | -0.012720784205868 |
| $h(20)$ | -0.027773994531193 | -0.027401753123643 |
| $h(21)$ | -0.015312108416339 | -0.012215226484171 |
| $h(22)$ | 0.003282086782674 | 0.000941382486533 |
| $h(23)$ | 0.004688277571827 | 0.005714631773760 |
| $h(24)$ | -0.002918631575345 | -0.002816191307997 |

### 5.4.2 *Asymmetric FIR Highpass Filter*

Asymmetric FIR highpass filter of order 24 and group delay 10 is designed using the proposed reference-point-based MOABC approach and results are compared with the reference-point-based MODE algorithm. Three objective functions are formulated to represent minimax magnitude error in passband $f_1(\mathrm{c})$, minimax magnitude error in stopband $f_2(\mathrm{c})$, and minimax group delay error in passband $f_3(\mathrm{c})$. The reference point is set as follows: $R_i = [\delta_{p1}, \delta_{s1}, \delta_g]$ according to Table 5.3. The plots of magnitude response, impulse response, passband and stopband peak errors of the highpass digital filters are shown in Figure 5.6 and Figure 5.7 respectively. The Pareto front approximation for the multiobjective ABC and the multiobjective DE are plotted in Figure 5.8 and 5.9.

The upper and lower limits of the minimax passband magnitude error, minimax stopband magnitude error and minimax passband group delay error of asymmetric FIR highpass using the MOABC and the MODE are given in Table 5.7.

**Table 5.7** Objective Function Range for Asymmetric FIR Highpass Filter

| | | MOABC | MODE |
|---|---|---|---|
| $Pe_{PB}$ | $e_{PB}^U$ | 0.061789588849616 | 0.069870090172511 |
| | $e_{PB}^L$ | 0.061544917669254 | 0.040678252895968 |
| $Pe_{SB}$ | $e_{SB}^U$ | 0.062016321116392 | 0.071038091506139 |
| | $e_{SB}^L$ | 0.061424559156228 | 0.040310924288638 |
| $Pe_{GD}$ | $e_{GD}^U$ | 0.006304138820070 | 0.892754974163104 |
| | $e_{GD}^L$ | 0.005883467242846 | 0.006183613321788 |

The passband, stopband and group delay peak error values from the two designs are compared in Table 5.8. Simulation results indicate that FIR highpass filter of group delay 10 designed using the multiobjective ABC algorithm has lower passband peak error, stopband peak error and group delay error in comparison to those of the multiobjective DE designs.

**Table 5.8** Peak Error Values of Asymmetric FIR Highpass Filter

| | MOABC | MODE |
|---|---|---|
| Passband peak error | 0.061789588849616 | 0.069870090172511 |
| Stopband peak error | 0.061424559156228 | 0.066432945192353 |
| Group delay error | 0.005962542752997 | 0.008932567749717 |

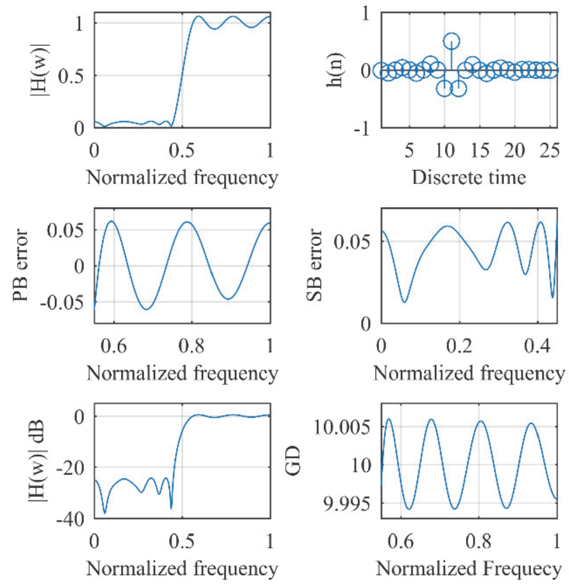


**Figure 5.6** Magnitude Response, Impulse Response, Passband and Stopband Errors of Asymmetric FIR Highpass Filter Using MOABC

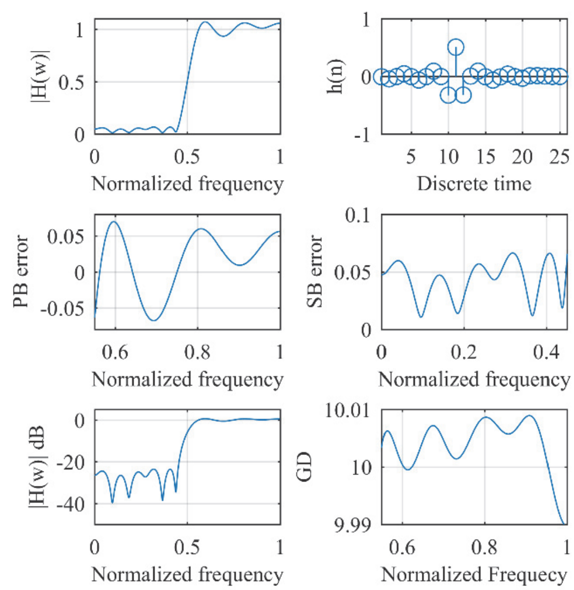


**Figure 5.7** Magnitude Response, Impulse Response, Passband and Stopband Errors of Asymmetric FIR Highpass Filter Using MODE

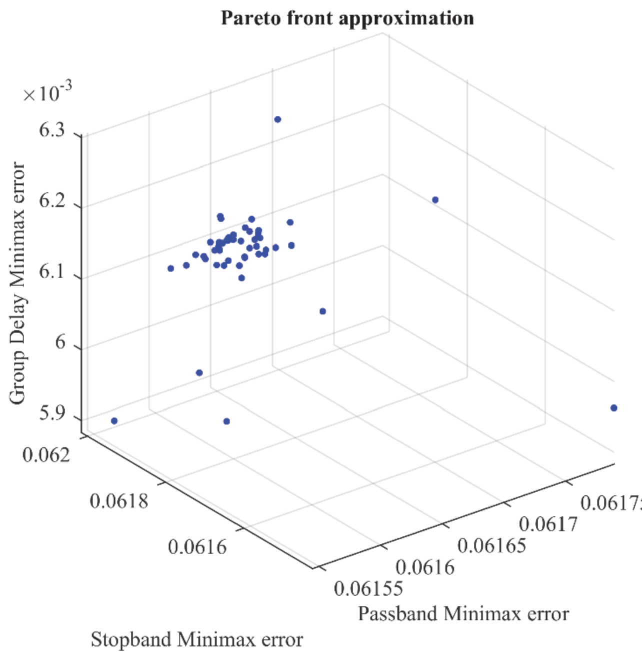


**Figure 5.8** Pareto Front Approximation of Asymmetric FIR Highpass Filter Using MOABC

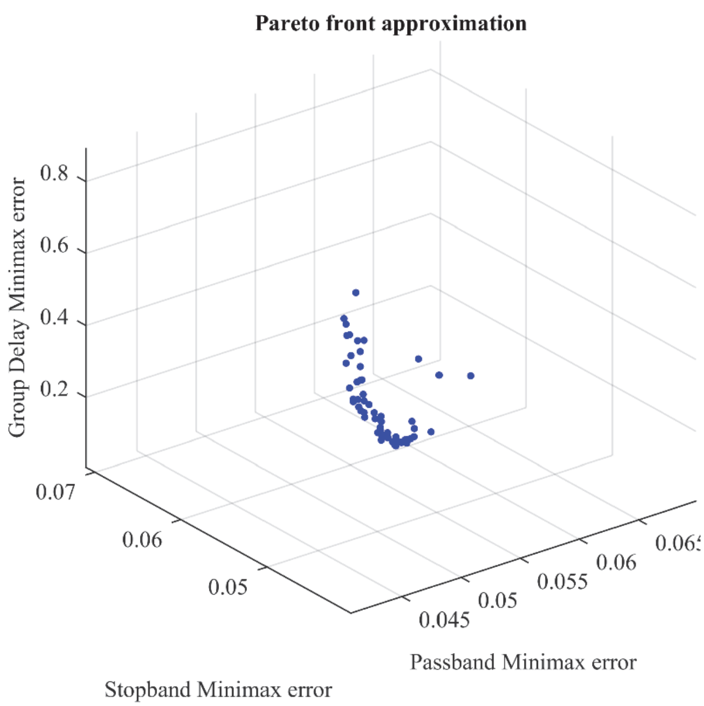


**Figure 5.9** Pareto Front Approximation of Asymmetric FIR Highpass Filter Using MODE

Filter coefficients of order $N = 24$ and group delay 10, asymmetric FIR highpass filter obtained using the reference-point-based MOABC and the reference-point-based MODE are listed in Table 5.9.

**Table 5.9** Coefficients of Asymmetric FIR Highpass Filter

| $h(n)$ | MOABC | MODE |
|---|---|---|
| $h(0)$ | -0.006490631823948 | -0.004767963442825 |
| $h(1)$ | -0.050197039268114 | -0.040107699017188 |
| $h(2)$ | -0.000157367354158 | -0.001654803726508 |
| $h(3)$ | 0.041789006819530 | 0.047333349706460 |
| $h(4)$ | 0.003121072627210 | -0.003273162108325 |
| $h(5)$ | -0.055261575010827 | -0.062107420416884 |
| $h(6)$ | 0.001980086364495 | -0.002428123014813 |
| $h(7)$ | 0.103711906715006 | 0.091613323183115 |
| $h(8)$ | 0.002652593880740 | -0.001232973843750 |
| $h(9)$ | -0.318542092428193 | -0.324801529545916 |
| $h(10)$ | 0.499786604333063 | 0.507147944217247 |
| $h(11)$ | -0.321364047365022 | -0.321948672387241 |
| $h(12)$ | -0.003426374900522 | 0.005640409768723 |
| $h(13)$ | 0.096449813363240 | 0.093520903730638 |
| $h(14)$ | -0.002947694083502 | -0.004314375769320 |
| $h(15)$ | -0.058522603732333 | -0.063539435238503 |
| $h(16)$ | -0.001674706587845 | -0.007094116461884 |
| $h(17)$ | 0.037296077006058 | 0.041958958289250 |
| $h(18)$ | 0.002750968283580 | -0.002784913861289 |
| $h(19)$ | -0.037361055794202 | -0.030148661355791 |
| $h(20)$ | 0.008837359851564 | 0.012262341543969 |
| $h(21)$ | 0.008118915422543 | 0.014994722271082 |
| $h(22)$ | -0.000671996814803 | 0.007293286836324 |
| $h(23)$ | -0.003799469766575 | 0.001415790340192 |
| $h(24)$ | -0.002108536652073 | -0.000442791368306 |

### *5.4.3 Asymmetric FIR Bandpass Filter*

Asymmetric FIR bandpass filter of order 24 and group delay 10 is designed using the proposed reference-point-based MOABC approach. The minimax error approximation functions in the multiobjective space are set as follows, $f_1(\mathrm{c})$ represents the minimax magnitude error in passband, $f_2(\mathrm{c})$ and, $f_3(\mathrm{c})$ represents the minimax magnitude error in stopband 1 and stopband 2 respectively, $f_4(\mathrm{c})$ represents the minimax group delay error in passband. The reference points are set as follows: $R_i = [\delta_p, \delta_{s1}, \delta_{s2}, \delta_g]$ as given in Table 5.3. The peak error values of asymmetric FIR bandpass for passband magnitude response, stopband magnitude response and passband group delay using the reference-point-based MOABC and MODE are given in Table 5.10. The upper and lower peak error limits of passband magnitude, stopband magnitude and passband group delay using the reference-point-based MOABC and MODE are obtained as shown in Table 5.11. Design results indicate that FIR bandpass filter of group delay 10 designed using the multiobjective ABC algorithm has lower passband magnitude peak error, stopband magnitude peak errors and passband group delay peak error than the results obtained using the multiobjective DE.

Plots of frequency responses using both the design methods are given in Figure 5.10-Figure 5.11.

**Table 5.10** Peak Error Values of Asymmetric FIR Bandpass Filter

| | MOABC | MODE |
|---|---|---|
| Passband magnitude error | 0.062797982630476 | 0.065628078408351 |
| Stopband1 magnitude error | 0.060139429882185 | 0.065332461365080 |
| Stopband2 magnitude error | 0.060552848906849 | 0.065877538077204 |
| Group delay error | 0.003704585067247 | 0.005968057189008 |

**Table 5.11** Objective Function Range for Asymmetric FIR Bandpass Filter

| | | MOABC | MODE |
|---|---|---|---|
| $Pe_{PB}$ | $e_{PB}^U$ | 0.062839255867158 | 0.065628078408351 |
| | $e_{PB}^L$ | 0.062761387004337 | 0.048775531431478 |
| $Pe_{SB2}$ | $e_{SB1}^U$ | 0.060221101142016 | 0.065429924019549 |
| | $e_{SB1}^L$ | 0.060030433484600 | 0.049708538297435 |
| $Pe_{SB2}$ | $e_{SB2}^U$ | 0.060618291356963 | 0.065877538077204 |
| | $e_{SB2}^L$ | 0.060473621200255 | 0.048832081164414 |
| $Pe_{GD}$ | $e_{GD}^U$ | 0.003704585067247 | 0.315435732758012 |
| | $e_{GD}^L$ | 0.003542782429378 | 0.005968057189008 |

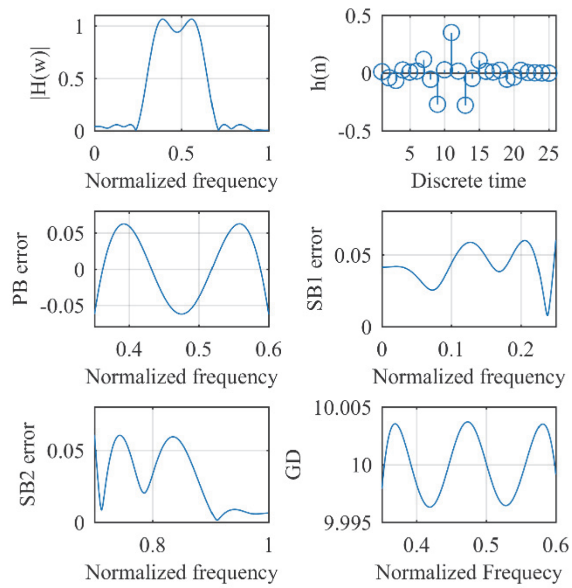


**Figure 5.10** Magnitude Response, Impulse Response, Passband and Stopband Errors of Asymmetric FIR Bandpass Filter Using MOABC

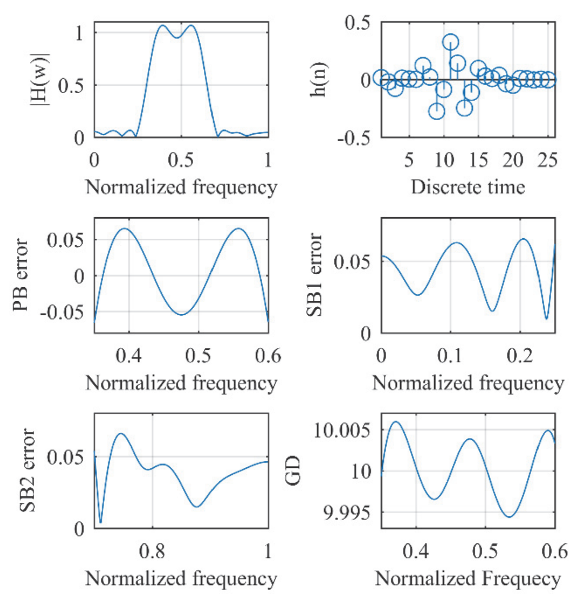


**Figure 5.11** Magnitude Response, Impulse Response, Passband and Stopband Errors of Asymmetric FIR Bandpass FIR Filter Using MODE

Filter coefficients of order $N = 24$, and group delay 10, FIR bandpass filter designed using the MOABC and the MODE is shown in Table 5.12.

**Table 5.12** Coefficients of Asymmetric FIR Bandpass Filter

| $h(n)$ | MOABC | MODE |
|---|---|---|
| $h(0)$ | 0.012951482189904 | 0.017297153032816 |
| $h(1)$ | -0.039546034729322 | -0.019474918438111 |
| $h(2)$ | -0.062560878577823 | -0.075960389893364 |
| $h(3)$ | 0.028730286568553 | 0.012832480626508 |
| $h(4)$ | 0.009447907031255 | 0.005526395023910 |
| $h(5)$ | 0.018404819846923 | 0.002444692855810 |
| $h(6)$ | 0.119642885613247 | 0.122901120736397 |
| $h(7)$ | -0.053046519147178 | 0.022499578347350 |
| $h(8)$ | -0.269363719466660 | -0.275160082271532 |
| $h(9)$ | 0.029817463422804 | -0.085636850847237 |
| $h(10)$ | 0.350490484645877 | 0.323657456051355 |
| $h(11)$ | 0.018193036666201 | 0.142902017171884 |
| $h(12)$ | -0.277762631857184 | -0.246561604531935 |
| $h(13)$ | -0.044698187894366 | -0.110510795927713 |
| $h(14)$ | 0.111818416308811 | 0.096896861498355 |
| $h(15)$ | 0.016849125631122 | 0.030838001527176 |
| $h(16)$ | 0.010148901020572 | 0.010200967771343 |
| $h(17)$ | 0.026808780676231 | 0.037754591511162 |
| $h(18)$ | -0.051813730042961 | -0.035789340080249 |
| $h(19)$ | -0.034217988852655 | -0.048221688388749 |
| $h(20)$ | 0.025119656957318 | 0.010401246924535 |
| $h(21)$ | 0.006028708903840 | 0.007539636922988 |
| $h(22)$ | 0.004099368158720 | -0.002224379283858 |
| $h(23)$ | 0.002906484316684 | 0.003422404322880 |
| $h(24)$ | 0.000346340260336 | -0.001014117371160 |

### *5.4.4 Asymmetric FIR Bandstop Filter*

Asymmetric FIR bandstop filter of order 24 and group delay 10 is designed using the proposed reference-point-based MOABC approach. Objective functions for minimax errors are formulated as follows; $f_1(\mathrm{c})$ and $f_2(\mathrm{c})$ represents the minimax magnitude errors in passband 1 and passband 2 respectively, $f_3(\mathrm{c})$ represents the minimax magnitude error in stopband, $f_4(\mathrm{c})$ and $f_5(\mathrm{c})$ represents the minimax group delay errors in passband 1 and passband 2 respectively. Reference points are set as follows: $R_i = [\delta_{p1}, \delta_{p2}, \delta_{s1}, \delta_g, \delta_g]$ according to Table 5.3. The peak error values obtained are listed in Table 5.13. The upper and lower peak error limits of passband magnitude, stopband magnitude and passband group delay obtained using the reference-point-based MOABC and MODE algorithms are compared in Table 5.14. Simulation results indicate that FIR bandstop filter designed using the multiobjective ABC algorithm has lower passband magnitude peak errors, stopband magnitude peak error and passband group delay errors than the results obtained using the multiobjective DE.

**Table 5.13** Peak Error Values of Asymmetric FIR Bandstop Filter

| | MOABC | MODE |
|---|---|---|
| Passband 1 magnitude error | 0.051215606573326 | 0.056895217471357 |
| Passband 2 magnitude error | 0.055797702268403 | 0.061661241102966 |
| Stopband magnitude error | 0.061653129912316 | 0.066137076403756 |
| Group delay 1 peak error | 0.003098649096721 | 0.006889124566342 |
| Group delay 2 peak error | 0.004373552107509 | 0.006044379455240 |

**Table 5.14** Objective Function Range for Asymmetric FIR Bandstop Filter

| | | MOABC | MODE |
|---|---|---|---|
| $Pe_{PB1}$ | $e_{PB1}^U$ | 0.051266076268417 | 0.066115904535004 |
| | $e_{PB1}^L$ | 0.051191984215891 | 0.044067817134488 |
| $Pe_{PB2}$ | $e_{PB2}^U$ | 0.055825715188505 | 0.065708665758640 |
| | $e_{PB2}^L$ | 0.055727764765201 | 0.043911406326871 |
| $Pe_{SB}$ | $e_{SB}^U$ | 0.061670446803766 | 0.066137076403756 |
| | $e_{SB}^L$ | 0.061648931209908 | 0.044338243703762 |
| $Pe_{GD1}$ | $e_{GD1}^U$ | 0.003156622663099 | 0.615125380742207 |
| | $e_{GD1}^L$ | 0.003065476179955 | 0.002986743919777 |
| $Pe_{GD2}$ | $e_{GD2}^U$ | 0.004460186281236 | 0.489269490433387 |
| | $e_{GD2}^L$ | 0.004365659959170 | 0.004016657752175 |

Plots of magnitude response, impulse response, passband and stopband errors of the FIR bandstop filter designed using the MOABC and the MODE algorithms are shown in Figures 5.12 and 5.13, respectively.

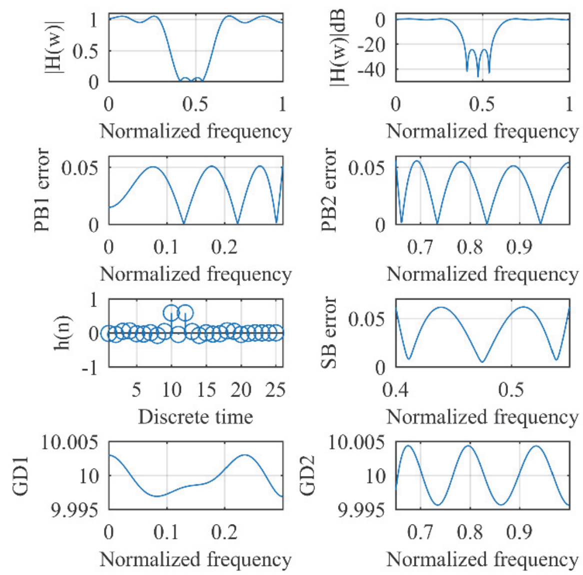


**Figure 5.12** Magnitude Response, Impulse Response, Passband and Stopband Errors of Asymmetric FIR Bandstop Filter Using MOABC

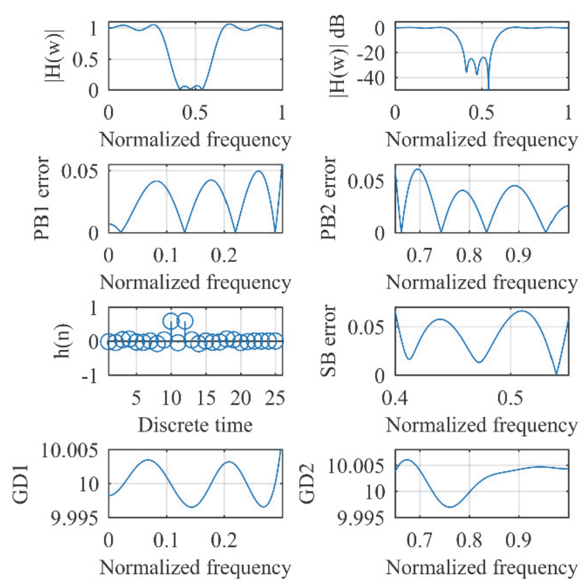


**Figure 5.13** Magnitude Response, Impulse Response, Passband and Stopband Errors of Asymmetric FIR Bandstop FIR Filter Using MODE

Filter coefficients of order $N = 24$, and group delay 10 asymmetric FIR bandstop filter designed using the MOABC and the MODE is shown in Table 5.15.

**Table 5.15** Coefficients of Asymmetric FIR Bandstop Filter

| $h(n)$ | MOABC | MODE |
|---|---|---|
| $h(0)$ | -0.018421089920395 | -0.018730097292734 |
| $h(1)$ | -0.045309875930292 | -0.042919072876450 |
| $h(2)$ | 0.044877744143236 | 0.041511988287556 |
| $h(3)$ | 0.059759471075057 | 0.060665709488127 |
| $h(4)$ | -0.026376962224459 | -0.030077210176051 |
| $h(5)$ | -0.036026794370626 | -0.038482376878884 |
| $h(6)$ | 0.003218959716513 | 0.003138745706479 |
| $h(7)$ | -0.075310482282336 | -0.074921970083302 |
| $h(8)$ | 0.037134665828635 | 0.033741712326813 |
| $h(9)$ | 0.586961490251204 | 0.585658922543081 |
| $h(10)$ | -0.046812134783624 | -0.049314293068551 |
| $h(11)$ | 0.587270992234954 | 0.590724610696998 |
| $h(12)$ | 0.038069700705038 | 0.034422968665259 |
| $h(13)$ | -0.075702872329833 | -0.077238686069151 |
| $h(14)$ | 0.001255180985708 | 0.000367856010122 |
| $h(15)$ | -0.036041595816818 | -0.037139759158003 |
| $h(16)$ | -0.024145950636344 | -0.027212401973508 |
| $h(17)$ | 0.060133315223306 | 0.060730167959274 |
| $h(18)$ | 0.042935368051682 | 0.039277561427109 |
| $h(19)$ | -0.045686019257138 | -0.043609544761427 |
| $h(20)$ | -0.016874170193133 | -0.017331867748600 |
| $h(21)$ | 0.000451675309408 | 0.000603840148797 |
| $h(22)$ | -0.000862221117489 | -0.000559279117441 |
| $h(23)$ | -0.000294168210020 | -0.000346498500380 |
| $h(24)$ | 0.000470336278089 | 0.000011627727241 |

## *5.5 Conclusions*

In this chapter, asymmetric FIR filters which simultaneously satisfies both magnitude and group delay specifications are designed using the reference-point-based multiobjective ABC algorithm. Lowpass, highpass, bandpass, and bandstop filters of order 24 and group delay of 10 are designed. In the reference-point-based MOABC approach, instead of approximating the whole Pareto front, the search is directed towards the regions of interest of the decision maker. By introducing the preferences of the decision maker during optimization phase, the computational complexities associated with approximating the whole Pareto front is reduced. The simulation results confirm that the reference-point-based MOABC algorithm can be used to obtain lower peak errors in magnitude response and group delay response than those of the reference-point-based MODE. In contrast to the differential evolution algorithm, the onlooker bee phase in the ABC algorithm employs a secondary search in refined regions of the solution space. This ensures that the MOABC can simultaneously minimize all objective functions towards a better solution.

# CHAPTER 6

# MULTIOBJECTIVE APPROACH FOR IIR FILTER DESIGN

FIR filters are inherently stable and has exact linear phase, however they have certain drawbacks such as higher group delay and it requires more hardware components compared to infinite impulse response (IIR) filters for the same set of magnitude and group delay specifications. Also, FIR filters cannot be used in audio signal processing applications where long delays are undesirable. Although IIR filters can achieve much better performance than FIR filters, there are some difficulties faced while designing them such as:

1. IIR filter design is a non-convex optimization problem with many local minima on error surfaces and thus the global optimum solution is difficult to be found and verified.
2. If both magnitude and group delay characteristics need to be optimized, stability constraints must be incorporated into the design procedures. But, when denominator order is greater than 2, stability domain cannot be expressed as a convex set in terms of the denominator coefficients.

In this chapter, the preference-point-based multiobjective ABC algorithm is used to design IIR filters. Physical programming (PP) technique is used to set the preferences and spherical pruning technique is used to maintain the external archive size. The chapter is organized as follows: Section 6.1 gives an introduction about IIR filter design methods; Section 6.2 describes IIR filter design problem formulation, Section 6.3 gives a brief description about the physical programming approach, the spherical pruning technique and the preference-point-based MOABC; design results of IIR lowpass, highpass and bandpass filters are given in Section 6.4 and conclusions are given in Section 6.5.

## 6.1 Introduction

Classical design methods of IIR filters include impulse invariant method, matched-z transformation, and bilinear transformation. Although these methods can design stable IIR digital filters, these methods can only be applied to transform standard analog filters, such as lowpass, highpass, bandpass and bandstop filters, into their digital counterparts satisfying magnitude response characteristic.

Given a prototype lowpass IIR digital filter meeting specified passband and stopband specifications, spectral transformation [195] is the most common technique for designing IIR lowpass, highpass, bandpass and bandstop digital filters meeting the same passband and stopband specifications. Other approaches to design digital filters include linear transformation of classical LC filters [196], lattice modeling [197], and wave digital filters [198]-[199].

In [61], an effective method for designing short coefficient wordlength IIR digital filters is described by first equalizing passband and stopband statistical word lengths before optimization. Passive second order digital filters can be designed by applying linear transformation on two-port gyrator circuit [99]-[100], [110] to realize first-order and second-order multioutput digital filters [101]-[104]. First-order and second-order tunable and variable passive digital filters [105]-[106], can be designed from passive digital filters [99]-[100] by changing the values of respective filter coefficients. The work is also extended to 1-D high-order passive digital filter design [107]-[109] and 2-D passive digital filter structure [110]. Tunable filters of higher order and sharp cut off frequency can be designed by introducing analytical expression [108] for filter coefficients of both first-order and second-order passive digital filter sections or only to second-order section. Adaptive IIR digital filters [86] can be partially stabilized by using an adaptive feedback gain resulting in an increase in convergence speed. Adaptive IIR filters can be applied to noise reduction and echo cancellation. By applying a saturation function such as bipolar ramp function and bipolar sigmoid function at output of an adaptive IIR digital filter, undesirable effects of instability arising in the filter can be avoided [87]. IIR filters with equiripple passband, stopband and linear phase passband can be obtained by using Remez exchange algorithm [66]-[67]. The method adopts the combination of a mirror image numerator polynomial to approximate equiripple magnitude response in passband and stopband. Furthermore, an all-pole transfer function is used to provide a constant group delay in the passband. Nearly linear phase IIR filter design can be achieved in [75], which group delay deviation is minimized under the constraints of maximum passband attenuation and minimum stopband attenuation. Stability constraint is incorporated in an optimization problem as a set of linear inequality constraints, by designing the filter as a cascade of second-order sections. In [71], IIR filter design with a new stability constraint based on argument principle is introduced. Weighted least square IIR filters can be designed using partial second-order factorization [76]; and minimax IIR filters with second-order factor updates is described in [74]; minimax IIR filter design using semidefinite programming (SDP) relaxation technique [72] and iterative second order cone programming (SOCP) [73] have been respectively advanced. A review on recent advancements in FIR filter approximation by IIR filter is presented in [68].

IIR filter design is a multimodal optimization problem, converging to a global optimum is not often possible using iterative gradient-based search algorithms, as the problem is highly sensitive to its starting points and requires a continuous and differentiable cost function. So stochastic and bio-inspired algorithms that are independent of gradient calculation can be used for designing IIR filters. Evolutionary algorithms such as harmony search algorithm [77]-[78] and differential evolution algorithm [187] can be applied for IIR filter design. IIR filters with linear phase passband of lower orders can be designed by local search operator enhanced multiobjective evolutionary algorithm (LS-MOEA) [79]-[80]. In this method, each of the IIR filter coefficients is represented as a combination of control and coefficient genes and optimization process searches for an optimal coefficient with minimum magnitude and phase response error.

Unlike linear phase FIR filters which involves the optimization of only amplitude response, IIR filter design requires the simultaneous optimization of magnitude and phase responses, and thus IIR filter design problem is formulated as multiobjective optimization problem rather than a single-objective optimization problem. Multiobjective evolutionary algorithms such as the multiobjective ABC algorithm [81], the multiobjective teaching learning-based optimization [82], the multiobjective cuckoo search algorithm [83] can be used to design higher order IIR filters. In such problems, the designer generates a set of alternative trade-offs, called Pareto optimal solutions, instead of a single optimum solution. The set of nondominated solutions are optimal such that none of the solutions in the entire search space is superior, when all the objectives are considered. Multiobjective decision making finds numerous applications in the field of engineering design, scientific experiments and business decision making.

Even though, the numerous solutions present in the Pareto front are optimal, the user needs only one solution for every practical application. The decision maker will have a region of interest in the objective space, and the quality of solutions outside those regions are not a concern for the designer. In the optimization process, a search can be guided towards the region of interest of the decision maker if the preference information can be incorporated into the search process. In this chapter, physical programming approach is used to incorporate the preferences of the decision maker into the multiobjective ABC algorithm. Using the multiobjective ABC algorithm, various IIR filters have been designed. Since IIR filters lack inherent stability, stability constraints need to be incorporated in their design procedure.

### *6.2 IIR Filter Design [1]*

Cascade-form realization of an IIR digital filter with numerator and denominator order $M = N$, can be expressed as,

$$
\begin{aligned}
H(z) &= b_0 \prod_{n=1}^{\frac{N}{2}} \frac{B_n(z)}{A_n(z)} \\
&= b_0 \prod_{n=1}^{\frac{N}{2}} \frac{(1 + b_{1n}z^{-1} + b_{2n}z^{-2})}{(1 + a_{1n}z^{-1} + a_{2n}z^{-2})} \qquad (6.1) \\
&= \sum_{m=0}^{\infty} \boldsymbol{c}_m z^{-n}
\end{aligned}
$$

In equation 6.1, $b_{in}$ and $a_{in}$ for $i = 1,2$ and $n = 1$ to $\frac{N}{2}$ are real valued coefficients and $b_0$ is a scaling constant. The coefficients $\boldsymbol{c}(m)$ for $m \geq 0$ represents the impulse response values of the IIR digital filter. The corresponding coefficient vector $\boldsymbol{c}$ consisting of $(2N + 1)$ distinct coefficients which can be expressed as,

$$\boldsymbol{c} = \left[b_{11} b_{21} a_{11} a_{21} \ldots\ldots b_{1\frac{N}{2}} b_{2\frac{N}{2}} a_{1\frac{N}{2}} a_{2\frac{N}{2}} b_0\right]^T \tag{6.2}$$

The stability triangle of a 2nd-order denominator transfer function $A(z)$ offers a necessary and sufficient condition to ensure stability which requires,

$$\begin{gathered} -2 < a_1 < 2 \\ -1 < a_2 < 1 \\ -(a_1 + 1) < a_2 < (a_1 - 1) \end{gathered} \tag{6.3}$$

where $A(z) = 1 + a_1 z^{-1} + a_2 z^{-2}$.

Substituting $z = e^{jw}$ into equation 6.1, the frequency response of $Nth$-order cascade IIR filter can be expressed as,

$$\begin{aligned} H(w) = H(z)|_{z=e^{jwT}} &= b_0 \prod_{n=1}^{\frac{N}{2}} \frac{\left(1 + b_{1n} e^{-jwT} + b_{2n} e^{-j2wT}\right)}{\left(1 + a_{1n} e^{-jwT} + a_{2n} e^{-j2wT}\right)} \\ &= b_0 \prod_{n=1}^{\frac{N}{2}} \frac{[1 + \sum_{i=1}^{2} b_{in} cos\ iwT] - j[\sum_{i=1}^{2} b_{in} sin\ iwT]}{[1 + \sum_{i=1}^{2} a_{in} cos\ iwT] - j[\sum_{i=1}^{2} a_{in} sin\ iwT]} \\ &= |H(w)| e^{j\theta(w)} \end{aligned} \tag{6.4}$$

The magnitude response $|H(w)|$ is equal to,

$$|H(w)| = |b_0| \prod_{n=1}^{\frac{N}{2}} \left\{ \frac{[1 + \sum_{i=1}^{2} b_{in} cos\ iwT]^2 + [\sum_{i=1}^{2} b_{in} sin\ iwT]^2}{[1 + \sum_{i=1}^{2} a_{in} cos\ iwT]^2 + [\sum_{i=1}^{2} a_{in} sin\ iwT]^2} \right\}^{\frac{1}{2}} \tag{6.5}$$

The phase response $\theta(w)$ can be expressed as,

$$\theta(w) = \arg H(w)$$

$$= \sum_{n=1}^{\frac{N}{2}} \left\{ -tan^{-1}\left[\frac{\sum_{i=1}^{2} b_{in} sin\ iwT}{1 + \sum_{i=1}^{2} b_{in} cos\ iwT}\right] + tan^{-1}\left[\frac{\sum_{i=1}^{2} a_{in} sin\ iwT}{1 + \sum_{i=1}^{2} a_{in} cos\ iwT}\right]\right\} \tag{6.6}$$

The group delay can be expressed as,

$$\tau(w) = -\frac{\partial\theta(w)}{\partial wT}$$

$$= \sum_{n=1}^{\frac{N}{2}} \left\{ \frac{1}{1 + c(n)^2} \frac{\partial c(n)}{\partial wT} - \frac{1}{1 + d(n)^2} \frac{\partial d(n)}{\partial wT} \right\} \tag{6.7}$$

where,

$$c(n) = \frac{\sum_{i=1}^{2} b_{in} sin\ iwT}{1 + \sum_{i=1}^{2} b_{in} cos\ iwT} \qquad \text{for } n = 1\ to\ \frac{N}{2}$$

$$d(n) = \frac{\sum_{i=1}^{2} a_{in} sin\ iwT}{1 + \sum_{i=1}^{2} a_{in} cos\ iwT} \qquad \text{for } n = 1\ to\ \frac{N}{2} \tag{6.8}$$

Taking partial derivatives of $c(n)$, for $n = 1\ to\ \frac{N}{2}$,

$$\frac{\partial c(n)}{\partial wT} = \frac{[1 + \sum_{i=1}^{2} b_{in} cos\ iwT][\sum_{i=1}^{2} ib_{in} cos\ iwT]}{[1 + \sum_{i=1}^{2} b_{in} cos\ iwT]^2} + \frac{[1 + \sum_{i=1}^{2} b_{in} sin\ iwT][\sum_{i=1}^{2} ib_{in} sin\ iwT]}{[1 + \sum_{i=1}^{2} b_{in} cos\ iwT]^2} \tag{6.9}$$

Taking partial derivatives of $d(n)$, for $m = 1\ to\ \frac{M}{2}$,

$$\frac{\partial d(n)}{\partial wT} = \frac{[1 + \sum_{i=1}^{2} a_{in} cos\ iwT][\sum_{i=1}^{2} ia_{in} cos\ iwT]}{[1 + \sum_{i=1}^{2} a_{in} cos\ iwT]^2} + \frac{[1 + \sum_{i=1}^{2} a_{in} sin\ iwT][\sum_{i=1}^{2} ia_{in} sin\ iwT]}{[1 + \sum_{i=1}^{2} a_{in} cos\ iwT]^2} \tag{6.10}$$

The objective function of the minimax magnitude error $e_{mag}(\mathbf{c})$ is defined by,

$$e_{mag}(\mathbf{c}) = \left[ \sum_{w_i \in \Omega_I} \left| |H(\mathrm{c}, w_i)| - H_d(w_i) \right|^p \right]^{\frac{1}{p}} \quad \text{for } \forall w_i \in \Omega_I \tag{6.11}$$

where $H_d(w_i) = 1$ in passband and $H_d(w_i) = 0$ in stopband(s); and $\Omega_I$ denotes union of frequency bands of interest and $p$ is a positive integer.

Similarly, the objective function of the minimax group delay error $e_{gd}(\mathbf{c})$ among the passband can be calculated by,

$$e_{gd}(\mathbf{c}) = \left[ \sum_{w_i} |\tau(\mathbf{c}, w_i) - \tau_d(w_i)|^p \right]^{\frac{1}{p}} \quad \text{for } w_{p1} \leq w_i \leq w_{p2} \tag{6.12}$$

where $\tau_d(w)$ is the desired group delay.

The maximum passband attenuation $R_p$ (in dB) is related to passband ripple $\delta_p$ as,

$$\delta_p = \frac{10^{R_p/20} - 1}{10^{R_p/20} + 1} \tag{6.13}$$

The minimum stopband attenuation $R_s$ (in dB) is related to stopband ripple $\delta_s$ as,

$$\delta_s = 10^{-R_s/20} \tag{6.14}$$

The multiobjective optimization problem for IIR filter design searches for an optimal coefficient vector $\mathbf{c}$ that minimizes the objective functions $e_{mag}(\mathbf{c})$ and $e_{gd}(\mathbf{c})$ simultaneously.

### 6.3 Physical-Programming-Based Multiobjective ABC Algorithm

The preferences are incorporated into the multiobjective ABC algorithm *a priori* and explicitly using the global physical programming approach and the size of the external archive is updated using the spherical pruning technique.

#### 6.3.1 Physical Programming Approach

The physical programming (PP) [188] is a technique for multiobjective optimization that formulates the design objectives into an understandable language and enables a designer to express preferences to each of the objective functions. In this approach, a designer expresses his preferences related to each objective function with details using the information available about the problem at the optimization phase. In PP, the decision maker expresses his preferences using different degrees of desirability: highly desirable (HD), desirable (D), tolerable (T), undesirable (U) and highly undesirable (HU). There are eight preference functions classified into 4 soft classes and 4 hard classes [190]. The soft class functions are as follows: Class 1S (smaller is better), Class 2S (larger is better), Class 3S (value is better), Class 4S (range is better). The hard class functions are as follows: Class 1H (must be smaller), Class 2H (must be larger), Class 3S (must be equal), Class 4S (must be in range). The selection of class function by a designer depends on the degree of sharpness of his preferences. In this design, Class 1S function is chosen to set preferences.

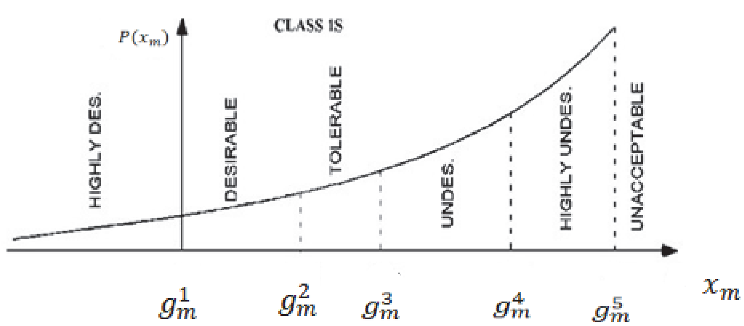


**Figure 6.1** 1S Class Function: Smaller is the Better [190]

PP can be used as a selection mechanism to store and replace solutions in the external archive. Given a set of preferences $\mathfrak{B}$ with $N$ ranges for $M$ objectives [188].

$$\mathfrak{B} = \begin{bmatrix} g_1^1 & g_1^2 & \dots & g_1^{\mathrm{N}} \\ g_2^1 & g_2^1 & \dots & g_2^N \\ \dots & \dots & \dots & \dots \\ g_m^1 & g_m^2 & \dots & g_M^{\mathrm{N}} \end{bmatrix} \tag{6.15}$$

When $N = 5$, the preference set can be set using the following different ranges:

**HD**: Highly desirable if $g_m^0 \leq g_m(\boldsymbol{x}) \leq g_m^1$

**D**: Desirable if $g_m^1 \leq g_m(\boldsymbol{x}) \leq g_m^2$

**T**: Tolerable if $g_m^2 \leq g_m(\boldsymbol{x}) \leq g_m^3$

**U**: Undesirable if $g_m^3 \leq g_m(\boldsymbol{x}) \leq g_m^4$

**HU**: Highly undesirable $g_m^4 \leq g_m(\boldsymbol{x}) \leq g_m^5$

For $m$ th objective, the class function is defined as,

$$n_m^{1s}(\boldsymbol{x}) = \begin{cases} f_m^N(\boldsymbol{x}) & if\, g_m(\boldsymbol{x}) > g_m^N \\ f_m^k(\boldsymbol{x}) & if\, g_m(\boldsymbol{x}) \in [g_m^{k-1}, \dots \dots g_m^k], k\epsilon[1,2,3 \dots \dots \dots N] \\ 0 & if\ g_m(\boldsymbol{x}) < g_m^0 \end{cases} \tag{6.16}$$

Preference function $f_m^k(\boldsymbol{x})$ in the range $k$ for the objective function value $g_m(\boldsymbol{x})$ is defined as,

$$f_m^k(\boldsymbol{x}) = \alpha_{k-1} + \Delta\alpha_k \left( \frac{g_m(\boldsymbol{x}) - g_m^{k-1}}{g_m^k - g_m^{k-1}} \right) \tag{6.17}$$

where,

$\alpha_0 = 0; \alpha_1 = \alpha_{ini}; \quad \alpha_{ini} \geq 0$

$\Delta\alpha_k = \alpha_k - \alpha_{k-1} \;\; ; \alpha_k = \alpha_{k-1}.M; \;\; (1 \leq k \leq N\,)$

Minimization is performed for soft class as follows,

$$\min_{\boldsymbol{x}} J(\boldsymbol{x}) = \frac{1}{n_{sc}} \sum_{m=1}^{n_{sc}} n_m^{1s}(\boldsymbol{x}) \tag{6.18}$$

$n_{sc}$ is the number of objectives in decision space.

A lower value of class function is always preferred over a higher value. The obtained solutions in the Pareto front are analyzed using the spherical pruning (SP) algorithm. The algorithm selects one solution for each spherical sector, according to the norm. This maintains a diversity in the Pareto front and prevents converging to a single Pareto optimal solution.

### *6.3.2 Spherical Pruning Technique*

If physical programming (PP) is used as such, it will evolve an entire population to a single Pareto optimal solution. Therefore, it must be merged with other mechanisms to maintain diversity in the Pareto front. Spherical pruning can maintain diversity in the Pareto front. The basic idea of spherical pruning is to analyze the proposed solutions in the current Pareto front approximation $J_p^*$ by using normalized spherical coordinates from a reference solution. With such an approach, it is possible to attain a good distribution along the Pareto front. The algorithm selects one solution for each spherical sector, according to a given norm or measure.

Given two solutions $\boldsymbol{\theta}^1$and $\boldsymbol{\theta}^2$ from a set, $\boldsymbol{\theta}^1$ has preference in the spherical sector over $\boldsymbol{\theta}^2$ if,

$$[\Lambda_\epsilon(\boldsymbol{\theta}^1) = \Lambda_\epsilon(\boldsymbol{\theta}^2)] \wedge [\|(J(\boldsymbol{\theta}^1)\|_p < \ \|(J(\boldsymbol{\theta}^2)\|_p \tag{6.19}$$

where,

$\|J(\boldsymbol{\theta})\|_p = \left(\sum_{q=1}^{m} \left|J_q(\boldsymbol{\theta})\right|^p\right)^{\frac{1}{p}}$ is a suitable $p$-norm, $\Lambda_\epsilon(\boldsymbol{\theta}^1)$ is the spherical sector defined as $\Lambda_\epsilon(\boldsymbol{\theta}^1) = \left[\frac{\beta_1(J(\boldsymbol{\theta}^1)}{\Lambda_1^{J_P^*}}, \dots\dots\dots\dots\dots\dots\dots, \frac{\beta_{m-1}(J(\boldsymbol{\theta}^1)}{\Lambda_{m-1}^{J_P^*}}\right]$, $\Lambda_1^{J_P^*}$ is the spherical grid on the m-dimensional space defined as $\Lambda_P^J = \left[\frac{\beta_1{}^U - \beta_1{}^L}{\beta_1{}^\epsilon}, \dots\dots, \frac{\beta_{m-1}{}^U - \beta_{m-1}{}^L}{\beta_{m-1}{}^\epsilon}\right]$, the upper and lower limit is defined as $\boldsymbol{\beta}^U = \left[\max \beta_1(J(\boldsymbol{\theta}^i) \dots\dots \max \beta_{m-1}(J(\boldsymbol{\theta}^i)\right]$ and $\boldsymbol{\beta}^L = \left[\min \beta_1(J(\boldsymbol{\theta}^i) \dots\dots \min \beta_{m-1}(J(\boldsymbol{\theta}^i)\right]$.

In this implementation, spherical pruning mechanism is used to confine the size of the external archive within the predetermined value. A detailed description about spherical pruning technique can be seen from [188].

### *6.3.3 Physical Programming Multiobjective ABC Algorithm*

The physical programming multiobjective ABC algorithm follows the steps of single-objective ABC algorithm, starting with a random initial population $\boldsymbol{X}_{0i},\ i \in [1, SN]$ where $SN$, the total number of food locations is generated using equation 2.2. For an $N$ th order IIR filter, the food source vector is of length $(2N+1)$ and is in the form of $\boldsymbol{c} = \left[b_{11} b_{21} a_{11} a_{21} \dots\dots b_{1\frac{N}{2}} b_{2\frac{N}{2}} a_{1\frac{N}{2}} a_{2\frac{N}{2}} b_0\right]^T$ where $b_0$ is the scaling constant and the limits are set for each of the coefficients to incorporate the stability constraints. Nondominated sorting is then performed on the population to obtain top solutions, and the initial archive $E_0$, is generated.

The new solutions in the employed bee phase are generated using the ABC algorithm as described in equation 2.3 and dominance criteria uses a pareto dominance approach instead of a greedy selection to replace current food source by a new food source,

$$\boldsymbol{x}^i\big|_t = \begin{cases} \boldsymbol{x}^i\big|_t & if \quad g_{pp}\left(f_m\left(\boldsymbol{v}^i\big|_t\right)\right) < g_{pp}\left(f_m\left(\boldsymbol{x}^i\big|_t\right)\right) \\ \boldsymbol{v}^i\big|_t & if \quad g_{pp}\left(f_m\left(\boldsymbol{v}^i\big|_t\right)\right) \geq g_{pp}\left(f_m\left(\boldsymbol{x}^i\big|_t\right)\right) \end{cases} \tag{6.20}$$

The fitness value for the food source is calculated as follows,

$$fit_m\left(\boldsymbol{x}^i\big|_t\right) = \left(\begin{array}{ll} \dfrac{1}{1+g_{pp}\left(f_m(\boldsymbol{x}^i|_t)\right)}, & \text{if } g_{pp}\left(f_m\left(\boldsymbol{x}^i\big|_t\right)\right) > 0 \\ 1+g_{pp}\left(f_m\left(\boldsymbol{x}^i\big|_t\right)\right), & \text{if } g_{pp}\left(f_m\left(\boldsymbol{x}^i\big|_t\right)\right) \leq 0 \end{array}\right) \tag{6.21}$$

where $g_{pp}\left(f_m\left(\boldsymbol{v}^i\big|_t\right)\right)$ is the global physical programming index of food source $\boldsymbol{v}_i$ for the objective function $m$. In the onlooker bee phase, a food source is selected according to its probability value calculated using equation 2.5 and searches for food locations near the good quality food sources. A solution which cannot be improved after several predetermined trials becomes a scout bee and is abandoned. The scout bees will then randomly search for a new solution. Nondominated sorting is performed and best solutions are archived as $E_t$. The size and diversity of the external archive is maintained using spherical pruning technique. When the algorithm is terminated, the final Pareto front is updated as $J_P^* = \mathrm{E}_t$. The flowchart for the physical programming multiobjective ABC is given in Figure 6.2 and the pseudocode is given in Table 6.1.

**Table 6.1** Pseudocode of Physical-Programming-based MOABC

1. Generate initial population $X_0$ with $SN$ individuals;
2. Evaluate the fitness of initial population $X_0$;
3. Apply nondominated sorting criteria on $X_0$ to obtain initial Archive $E_0$;
4. while stopping criterion do not satisfied do
5. Read iteration number $t$;
6. For all food sources
   Generate new food sources $\boldsymbol{v}^i\big|_t$ from current food source $\boldsymbol{x}^i\big|_t$ using ABC algorithm
7. Evaluate new food location $\boldsymbol{v}^i\big|_t$
   Using global Physical Programming approach update the population $\boldsymbol{x}^i\big|_t$ with $\boldsymbol{v}^i\big|_t$
8. End for
9. Apply dominance on criterion $X_t$ to obtain $E_t$;
10. Apply pruning mechanism to prune $E_t$
11. $t = \mathrm{t} + 1$;
12. End while
13. Algorithm terminates, Pareto front $J_P$ is updated by $J_P^* = E_t$.

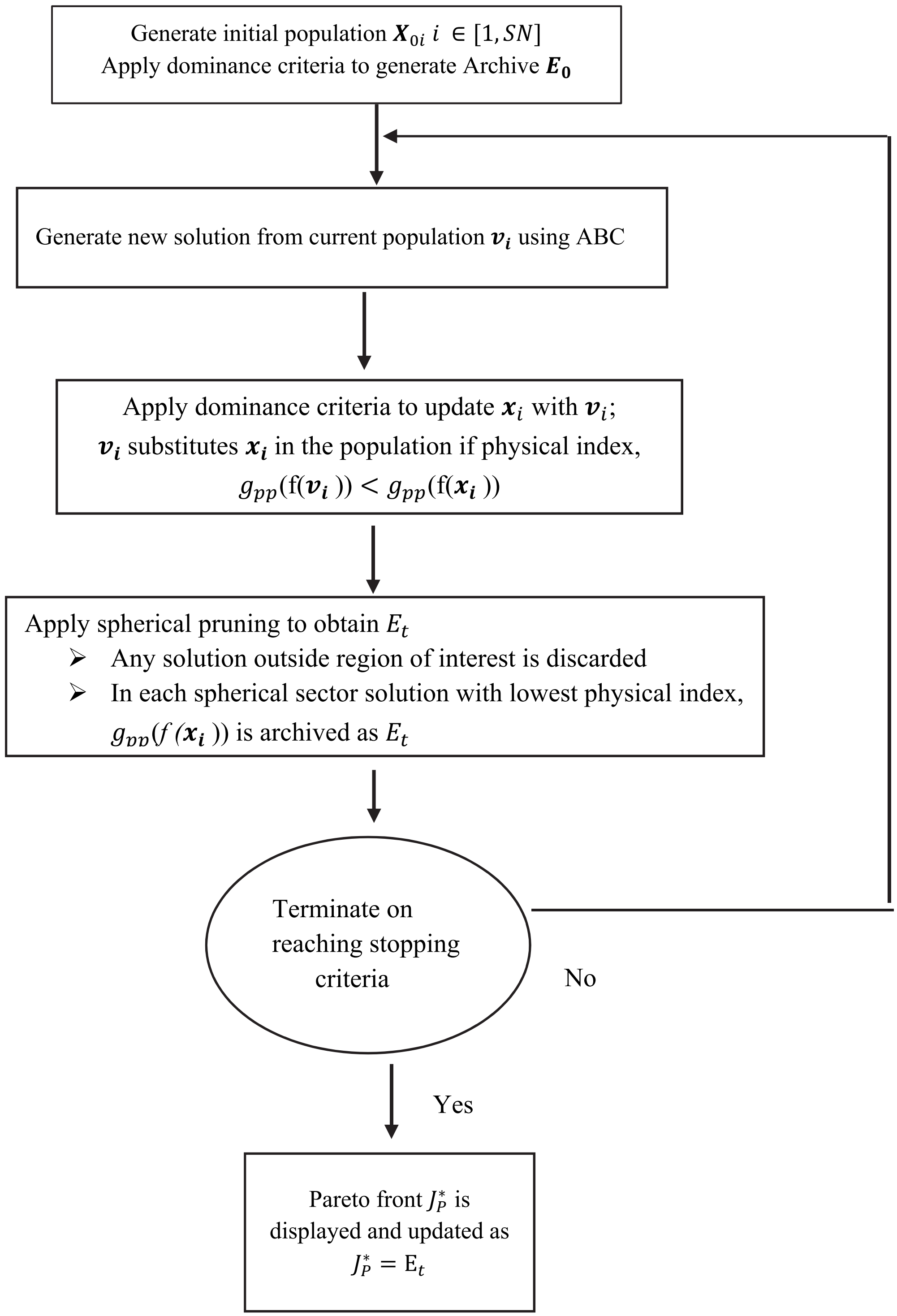


**Figure 6.2** Flowchart of Physical-Programming-based MOABC

### 6.4 Design Examples and Results

In this section, cascade-form IIR lowpass, highpass, bandpass digital filters are designed. An initialization filter should be chosen such that it satisfies at least one of the objective function specifications. In this design problem, the initial population $x_{ij}$, for $i = 1$ to $SN$ and $j = 1$ to $D$, is generated using elliptic filter which has the desired amplitude response but an arbitrary group delay response and the physical programming MOABC searches for a coefficient vector $\boldsymbol{c}$ that simultaneously minimize the objective function $e_{mag}(\mathbf{c})$ and $e_{gd}(\mathbf{c})$. For each food source in the solution space, stability is checked for all coefficient pairs $(a_{1n}, a_{2n})$ for $= 1$ to $N/2$ , and any solution that violates the stability criteria given in equation 6.3 is reverted to the required range.

The parameters of the physical programming MOABC and IIR filter specifications are given in Table 6.2. The frequency grid for optimization is $F_0$ and for error calculation is $F_e$, and it should be noted that a dense grid is selected for error calculation, but a coarse frequency grid is used in optimization. All optimizations are performed using an intel core i7-4790, 3.60 GHz with 12GB RAM desktop computer.

To evaluate the performance of the designed filter, maximum passband attenuation $R_p$ and minimum stopband attenuation $A_s$ are calculated as,

$$R_p = 20\log\frac{1+\delta_p}{1-\delta_p}\ \text{dB} \tag{6.22}$$

$$A_s = -20\log\delta_s\ \text{dB} \tag{6.23}$$

where $\delta_p$ and $\delta_s$ are the passband ripple and the stopband ripple respectively.

The quality of group delay characteristic $\tau$, of the filter is measured using maximum group delay deviation $Q_\tau$ which is defined by,

$$Q_\tau = \frac{100(\tau_{max} - \tau_{min})}{2 * \tau_{avg}} \tag{6.24}$$

where,

$$\tau_{ave} = \frac{\tau_{max} + \tau_{min}}{2} \tag{6.25}$$

$$\tau_{max} = \max_{w_i \in \Omega_p} \tau(\mathrm{c}, w_i) \tag{6.26}$$

$$\tau_{min} = \min_{w_i \in \Omega_p} \tau(\mathrm{c}, w_i) \tag{6.27}$$

where $\Omega_p$ represents the frequency region of interests in the passband.

**Table 6.2** MOABC Parameters and IIR Filter Specifications

| Symbol | Description | LP | HP | BP |
|---|---|---|---|---|
| $c_{num}^{[U]}$ | Upper bound of filter numerator coefficients | 8 | 8 | 8 |
| $c_{num}^{[L]}$ | Lower bound of filter numerator coefficients | -8 | -8 | 8 |
| $c_{a1n}^{[U]}$ | Upper bound of filter denominator coefficients | 2 | 2 | 2 |
| $c_{a1n}^{[L]}$ | Lower bound of filter denominator coefficients | -2 | -2 | -2 |
| $c_{a2n}^{[U]}$ | Upper bound of filter denominator coefficients | 1 | 1 | 1 |
| $c_{a2n}^{[L]}$ | Lower bound of filter denominator coefficients | -1 | -1 | -1 |
| $p$ | Least $p$th-order | 128 | 128 | 128 |
| $W(w_i)$ | Frequency weights for $0 \leq w_i \leq \pi$ | 1 | 1 | 1 |
| $M$ | Number of objective functions | 2 | 2 | 2 |
| $P$ | ABC population size | 100 | 100 | 100 |
| $Limit$ | Scout bee limit | 200 | 200 | 200 |
| $F_0$ | Optimization frequency grid | $[0: 0.005: 1]$ | | |
| $F_e$ | Peak error calculation frequency grid | $[0: 0.001: 1]$ | | |

The preference range selected for the objective functions are set using the design examples in [75] and are given in Table 6.3.

**Table 6.3** Preferences Range for IIR Filter Designs

| | Obj. fun. | $g_m^0$ | $g_m^1$ | $g_m^2$ | $g_m^3$ | $g_m^4$ | $g_m^5$ |
|---|---|---|---|---|---|---|---|
| LP | $e_{mag}$ | 0 | 0.015032 | 0.018038 | 0.021646 | 0.025975 | 0.031170 |
| | $e_{gd}$ | 0 | 0.019996 | 0.023995 | 0.028794 | 0.034553 | 0.041464 |
| HP | $e_{mag}$ | 0 | 0.005931 | 0.007117 | 0.008540 | 0.010248 | 0.012298 |
| | $e_{gd}$ | 0 | 0.004622 | 0.005546 | 0.006655 | 0.007986 | 0.007986 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| BP | $e_{mag}$ | 0 | 0.076478 | 0.091774 | 0.110129 | 0.132155 | 0.158586 |
| | $e_{gd}$ | 0 | 0.002745 | 0.003294 | 0.003953 | 0.004744 | 0.005693 |

### *6.4.1 IIR Lowpass Filter*

A 10th order IIR lowpass filter is designed, and filter specification is given in Table 6.4. In order to evaluate the performance of lowpass filter designed using the physical programming MOABC, its results are compared with example 6A-2 in [75], and design results are shown in Table 6.5. Plots of magnitude response, group delay response in passband of the designed IIR lowpass filter is shown in Figure 6.3 and the pole zero plot is given in Figure 6.4, red and blue dots indicate, poles and zeros, respectively. The pole zero plot shows that all 10 poles are inside the unit circle, which ensures that the designed IIR lowpass filter is stable.

**Table 6.4** IIR Lowpass Filter Design Specification

| Parameters | Values |
|---|---|
| Filter order $N$ | 10 |
| Distinct coefficients | 21 |
| Prescribed group delay in passband $\tau_d$ | 9.79 |
| Passband cutoff frequency $w_p$ | 0.4π |
| Stopband cutoff frequency $w_s$ | 0.56π |

**Table 6.5** Simulation Results of IIR Lowpass Filter

| Parameters | MOABC | Design 6A-2 [75] |
|---|---|---|
| Peak Error PB | 0.011850272132574 | 0.011870695865662 |
| Peak Error SB | 0.003161032888287 | 0.003162390333019 |
| Peak Group delay error PB | 0.019420941819440 | 0.019996828110962 |
| Max PB ripple (dB) | 0.205869948910281 | 0.206224795410127 |
| Min SB attenuation (dB) | 50.003419712266783 | 49.999690524800108 |
| $\tau_{avg}$ | 9.792987150078961 | 9.794423016597630 |
| $Q_\tau$ | 0.167811838090148 | 0.159006931668995 |
| Iteration number | 100000 | - |

The simulation results indicate that IIR lowpass filter designed using the physical-programming-based MOABC approach has lower passband peak error, stopband peak error and group delay error but larger maximum group delay deviation than the design example 6A-2 [75].

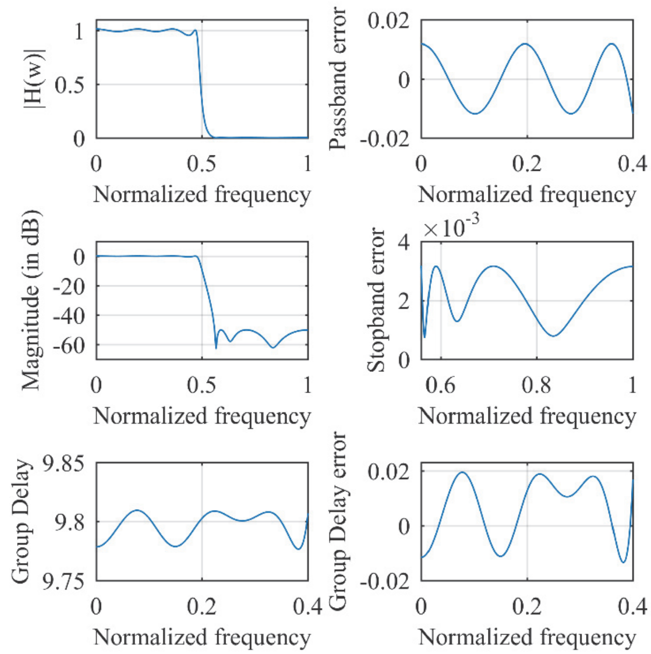


**Figure 6.3** Magnitude Response, Group Delay Response, Magnitude Errors and Group Delay Errors of IIR Lowpass Filter Designed Using MOABC

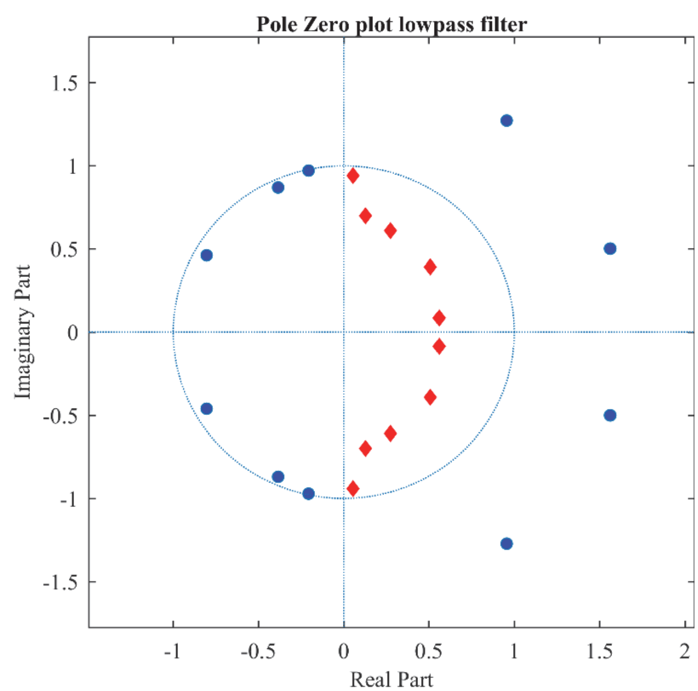


**Figure 6.4** Pole Zero Plot of IIR Lowpass Filter Designed Using MOABC

The pole zero values of IIR lowpass filter designed using the physical-programming-based MOABC algorithm and the example 6A-2 [75] is given in Table 6.6 and Table 6.7 respectively.

**Table 6.6** Poles and Zeros of IIR Lowpass Filter Designed Using MOABC

| Poles | Zeros |
|---|---|
| 0.558894210489161 + 0.084402856593976i | -0.385617081323501 + 0.869755263694404i |
| 0.558894210489161 - 0.084402856593976i | -0.385617081323501 - 0.869755263694404i |
| 0.508864518046386 + 0.392290143455185i | 0.955163900510168 + 1.272920488219513i |
| 0.508864518046386 - 0.392290143455185i | 0.955163900510168 - 1.272920488219513i |
| 0.127954755766791 + 0.698350504624347i | -0.803596542847903 + 0.461209386461021i |
| 0.127954755766791 - 0.698350504624347i | -0.803596542847903 - 0.461209386461021i |
| 0.273811070296788 + 0.610040811870065i | -0.206842243481173 + 0.971443865454256i |
| 0.273811070296788 - 0.610040811870065i | -0.206842243481173 - 0.971443865454256i |
| 0.054270897602955 + 0.941925241375190i | 1.562291434592744 + 0.501238247609318i |
| 0.054270897602955 - 0.941925241375190i | 1.562291434592744 - 0.501238247609318i |
| $g_0$ | 0.007869813458049 |

**Table 6.7** Poles and Zeros of IIR Lowpass Filter Example in 6A-2 [75]

| Poles | Zeros |
|---|---|
| 0.055899875082745 + 0.941595794139353i | 1.560532836267383 + 0.500037760769863i |
| 0.055899875082745 - 0.941595794139353i | 1.560532836267383 - 0.500037760769863i |
| 0.137189014239702 + 0.694579895414000i | 0.953953087535561 + 1.270175172428016i |
| 0.137189014239702 - 0.694579895414000i | 0.953953087535561 - 1.270175172428016i |
| 0.270300526450330 + 0.603032321027418i | -0.789354534288448 + 0.450991420752534i |
| 0.270300526450330 - 0.603032321027418i | -0.789354534288448 - 0.450991420752534i |
| 0.506904816181278 + 0.389758373464952i | -0.207363157634161 + 0.971912241425369i |
| 0.506904816181278 - 0.389758373464952i | -0.207363157634161 - 0.971912241425369i |
| 0.554307276978376 + 0.079692014072357i | -0.387023183479362 + 0.867177538670689i |
| 0.554307276978376 - 0.079692014072357i | -0.387023183479362 - 0.867177538670689i |
| $g_0$ | 0.008027005381132 |

The cascade-form representation of filter coefficients of IIR lowpass filter designed using the MOABC and in example 6A-2 is given in Table 6.8.

**Table 6.8** Filter Coefficients of IIR Lowpass Filter Using MOABC and 6A-2 [75]

| Section no: | Coefficients | MOABC | 6A-2 |
|---|---|---|---|
| Section 1 | $b_{11}$ | 0.413684486962345 | 1.578709068576896 |
| | $b_{21}$ | 0.986486897417032 | 0.826473842394121 |
| | $a_{11}$ | -0.547622140593576 | -1.108614553956733 |
| | $a_{21}$ | 0.447122294364160 | 0.313607374418078 |
| Section 2 | $b_{12}$ | 1.607193085695807 | -3.121065672534770 |
| | $b_{22}$ | 0.858481501836853 | 2.685300495264465 |
| | $a_{12}$ | -0.255909511533583 | -1.013809632362598 |
| | $a_{22}$ | 0.504065846832420 | 0.408864082353862 |
| Section 3 | $b_{13}$ | -3.124582869185487 | -1.907906175071123 |
| | $b_{23}$ | 2.691994307468313 | 2.523371461871180 |
| | $a_{13}$ | -0.108541795205910 | -0.540601052900625 |
| | $a_{23}$ | 0.890168490666340 | 0.436710354802987 |
| Section 4 | $b_{14}$ | 0.771234162647001 | 0.774046366958725 |
| | $b_{24}$ | 0.905174752132577 | 0.901783828125454 |
| | $a_{14}$ | -1.117788420978322 | -0.274378028479414 |
| | $a_{24}$ | 0.319486580719526 | 0.501262056741419 |
| Section 5 | $b_{15}$ | 1.910327801020336 | 0.412834654378671 |
| | $b_{25}$ | 2.532664646166801 | 0.987612884176496 |
| | $a_{15}$ | -1.017729036092773 | -0.111799750165491 |
| | $a_{25}$ | 0.412834654378671 | 0.889727435575182 |
| | $b_0$ | 0.007869813458049 | 0.008027005381132 |

### *6.4.2 IIR Highpass Filter*

A 14$^{th}$ order IIR highpass filter is designed, and filter specification is given in Table 6.9. In order to evaluate the performance of highpass filter designed using the physical programming MOABC, its results are compared with example 2A-2 in [75], and the simulation results are shown in Table 6.10. Plots of magnitude response, group delay response in passband of designed filter is shown in Figure 6.5. The pole-zero plot is given in Figure 6.6, red and blue dots indicates, poles and zeros, respectively.

**Table 6.9** IIR Highpass Filter Design Specification

| Parameters | Values |
|---|---|
| Filter order $N$ | 14 |
| Distinct coefficients | 28 |
| Prescribed group delay in passband $\tau_d$ | 18.026 |
| Passband cut off frequency $w_p$ | 0.6π |
| Stopband cut off frequency $w_s$ | 0.4π |

**Table 6.10** Simulation Results of IIR Highpass Filter

| Parameters | MOABC | Design 2A-2 [75] |
|---|---|---|
| Peak Error PB | 0.005716078250476 | 0.005726566029806 |
| Peak Error SB | 0.000205385627392 | 0.000211346697228 |
| Peak Group delay error | 0.005351591479776 | 0.004622731042623 |
| Max PB ripple (dB) | 0.099299531191199 | 0.099481728542779 |
| Min SB attenuation (dB) | 73.74859902031003 | 73.50009069236952 |
| $\tau_{avg}$ | 18.02573366738673 | 18.02299195551251 |
| $Q_\tau$ | 0.028368687584933 | 0.026673411553407 |
| Iteration number | 200000 | - |

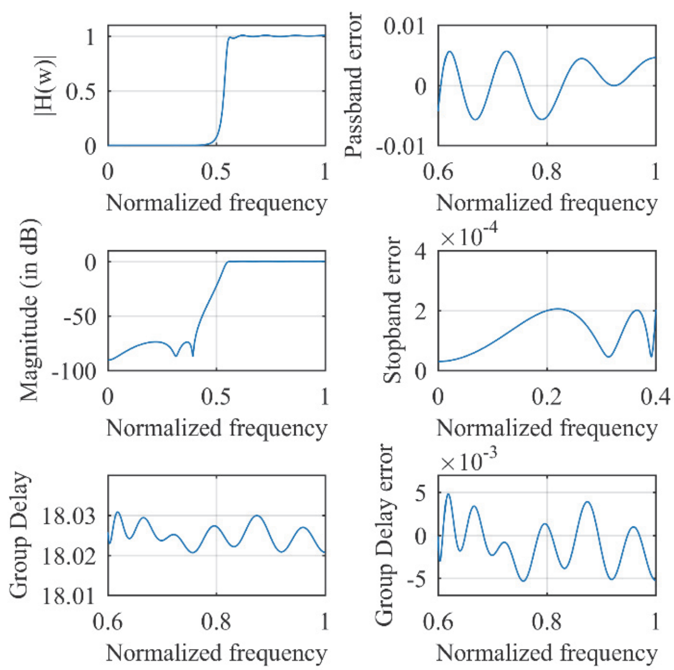


**Figure 6.5** Magnitude Response, Group Delay Response, Magnitude Errors and Group Delay Errors of IIR Highpass Filter Designed Using MOABC

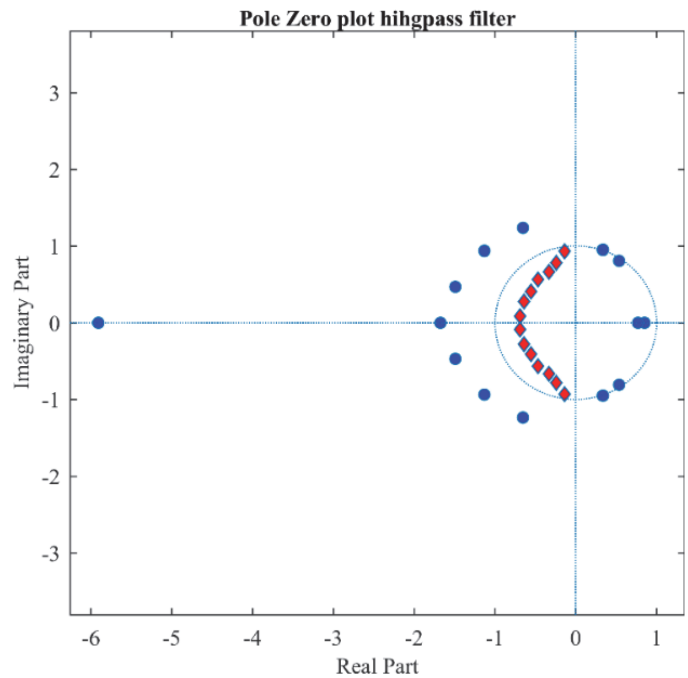


**Figure 6.6** Pole Zero Plot of IIR Highpass Filter Designed Using MOABC

The pole zero plot shows that all the 14 poles are inside the unit circle which ensures that the designed IIR highpass filter is stable.

The simulation results indicate that the IIR highpass filter designed using the physical-programming-based MOABC approach has lower passband peak error and stopband peak error but greater peak group delay error and maximum group delay deviation than the design example 2A-2 [75].

The pole-zero values of IIR highpass filter designed using the physical-programming-based MOABC algorithm and in the example 2A-2 [75] are given in Table 6.11 and Table 6.12 respectively. The cascade-form representation of the filter coefficients of IIR highpass filter designed using the MOABC and in example 2A-2 are given in Table 6.13.

**Table 6.11** Poles and Zeros of IIR Highpass Filter Using MOABC

| Poles | Zeros |
|---|---|
| -0.554670063556222 + 0.409554160702365i | -5.909928045952610 + 0.000000000000000i |
| -0.554670063556222 - 0.409554160702365i | 0.849292628142457 + 0.000000000000000i |
| -0.465498533338781 + 0.566507976580445i | -0.653199460008026 + 1.235828433557360i |
| -0.465498533338781 - 0.566507976580445i | -0.653199460008026 - 1.235828433557360i |
| -0.640180073099612 + 0.277547920370500i | 0.335396791617912 + 0.950264312284410i |
| -0.640180073099612 - 0.277547920370500i | 0.335396791617912 - 0.950264312284410i |
| -0.688147842803370 + 0.088479714553341i | -1.131884600820687 + 0.936708986098183i |
| -0.688147842803370 - 0.088479714553341i | -1.131884600820687 - 0.936708986098183i |
| -0.238546026334125 + 0.785066203456379i | 0.534685545678780 + 0.808784038013257i |
| -0.238546026334125 - 0.785066203456379i | 0.534685545678780 - 0.808784038013257i |
| -0.330136515849370 + 0.665451254702361i | -1.676361305137672 + 0.000000000000000i |
| -0.330136515849370 - 0.665451254702361i | 0.769029634344830 + 0.000000000000000i |
| -0.139084746832294 + 0.933226556234384i | -1.491469749070291 + 0.470065446556223i |
| -0.139084746832294 - 0.933226556234384i | -1.491469749070291 - 0.470065446556223i |
| $g_0$ | 1.389699269480474e-04 |

**Table 6.12** Poles and Zeros of IIR Highpass Filter in Example 2A-2 [75]

| Poles | Zeros |
|---|---|
| -0.129587030913823 + 0.935354095880280i | -1.109177147262147 + 1.275183686942845i |
| -0.129587030913823 - 0.935354095880280i | -1.109177147262147 - 1.275183686942845i |
| -0.228264708820005 + 0.780254340826646i | -1.493205529654857 + 0.285867750828198i |
| -0.228264708820005 - 0.780254340826646i | -1.493205529654857 - 0.285867750828198i |
| -0.324813730812861 + 0.665329580959911i | -1.322471208488332 + 0.805512989229631i |
| -0.324813730812861 - 0.665329580959911i | -1.322471208488332 - 0.805512989229631i |
| -0.452295969146698 + 0.548729032709602i | -0.678177338875631 + 1.315252474583515i |
| -0.452295969146698 - 0.548729032709602i | -0.678177338875631 - 1.315252474583515i |
| -0.545696283186623 + 0.422540657078167i | 0.914486674064672 + 0.415033644449005i |
| -0.545696283186623 - 0.422540657078167i | 0.914486674064672 - 0.415033644449005i |
| -0.638885665872786 + 0.259641928762650i | 0.331679107800718 + 0.942434670456086i |
| -0.638885665872786 - 0.259641928762650i | 0.331679107800718 - 0.942434670456086i |
| -0.660579026509467 + 0.068159724572040i | 0.530887153916813 + 0.841962916108677i |
| -0.660579026509467 - 0.068159724572040i | 0.530887153916813 - 0.841962916108677i |
| $g_0$ | 4.254845201569976e-004 |

**Table 6.13** Filter Coefficients of IIR Highpass Using MOABC and 2A-2 [75]

| Section | Coefficients | MOABC | 2A-2 |
|---|---|---|---|
| Section 1 | $b_{11}$ | 2.263769201641374 | -1.828973348129344 |
| | $b_{21}$ | 2.158586474212092 | 1.008538803066489 |
| | $a_{11}$ | 1.376295685606741 | 1.321158053022979 |
| | $a_{21}$ | 0.481376113442373 | 0.441010398319544 |
| Section 2 | $b_{12}$ | 0.907331670792841 | 2.218354294524278 |
| | $b_{22}$ | 1.289171521519846 | 2.856367379453924 |
| | $a_{12}$ | 0.660273031698741 | 1.277771331734585 |
| | $a_{22}$ | 0.551815491482108 | 0.475588825226353 |
| Section 3 | $b_{13}$ | 1.306398920016051 | 2.986411059309736 |
| | $b_{23}$ | 1.953941451743615 | 2.311383124755535 |
| | $a_{13}$ | 0.930997066677563 | 1.091392566392120 |
| | $a_{23}$ | 0.537620172069827 | 0.476325040370784 |
| Section 4 | $b_{14}$ | 5.060635417810152 | 2.644942416976690 |
| | $b_{24}$ | -5.019258322279907 | 2.397781273098042 |
| | $a_{14}$ | 1.109340127112445 | 0.904591938273504 |
| | $a_{24}$ | 0.475393489954082 | 0.505675195041505 |
| Section 5 | $b_{15}$ | -1.069371091357560 | -1.061774307833626 |
| | $b_{25}$ | 0.940020252902844 | 0.990742722296123 |
| | $a_{15}$ | 0.477092052668251 | 0.649627461636567 |
| | $a_{25}$ | 0.673233150489213 | 0.548167411027867 |
| Section 6 | $b_{16}$ | -0.670793583235823 | 1.356354677751240 |
| | $b_{26}$ | 1.015493271028951 | 2.189813574862431 |
| | $a_{16}$ | 1.280360146199224 | 0.456529417637149 |
| | $a_{26}$ | 0.486863374095814 | 0.660901613669359 |
| Section 7 | $b_{17}$ | 2.982939498140583 | -0.663358215601437 |
| | $b_{27}$ | 2.445443536437899 | 0.998194138629152 |
| | $a_{17}$ | 0.278169493664589 | 0.259174061827633 |
| | $a_{27}$ | 0.890256372062492 | 0.891680083261421 |
| | $b_0$ | 1.389699269474e-04 | 4.25484520156e-04 |

### *6.4.3 IIR Bandpass Filter*

A 14$^{th}$ order IIR bandpass filter is designed, and filter specification is given in Table 6.14. In order to evaluate the performance of bandpass filter designed using the physical programming MOABC, its results are compared with the example 3A-2 in [75], and the simulation results are shown in Table 6.15. Plots of magnitude response, group delay response in passband of the designed filter is shown in Figure 6.7. The pole-zero plot is given in Figure 6.8, red and blue dots indicates, poles and zeros, respectively. The pole-zero plot shows that all the 14 poles are inside the unit circle which ensures that the designed IIR bandpass filter is stable.

**Table 6.14** IIR Bandpass Filter Design Specification

| Parameters | Values |
|---|---|
| Filter order $N$ | 14 |
| Distinct coefficients | 28 |
| Prescribed group delay in passband $\tau_d$ | 25.54 |
| Stopband 1 cut off frequency $w_{s1}$ | 0.2π |
| Passband 1 cut off frequency $w_{p1}$ | 0.3π |
| Passband 2 cut off frequency $w_p$ | 0.5π |
| Stopband 2 cut off frequency $w_s$ | 0.7π |

**Table 6.15** Simulation Results of IIR Bandpass Filter

| Parameters | MOABC | Design 3A-2 [75] |
|---|---|---|
| Peak Error PB | 0.058314239202659 | 0.059398878299075 |
| Peak Error SB 1 | 0.008545505128267 | 0.008544664889970 |
| Peak Error SB 2 | 0.008541613067068 | 0.008544696148769 |
| Peak Group delay error | 0.002741178379249 | 0.002745356038197 |
| Max PB ripple (dB) | 1.014172718247246 | 1.033080328853921 |
| Min SB 1 attenuation | 41.365245217237678 | 41.366099300855993 |
| Min SB 2 attenuation | 41.369202117897160 | 41.366067525475515 |
| $\tau_{avg}$ | 25.542425092292156 | 25.542192048749879 |
| $Q_\tau$ | 0.001274583887433 | 0.00126449041859 |
| Iteration number | 200000 | - |

The simulation results indicate that IIR bandpass filter designed using the physical-programming-based MOABC approach has lower passband peak error, stopband 1 peak error and group delay error but greater peak stopband 2 peak error and maximum group delay deviation than the design example 3A-2 [75]. The pole-zero values of IIR highpass filter designed using the physical-programming-based MOABC algorithm and in the example 3A-2 [75] are given in Table 6.16 and Table 6.17 respectively. The cascade-form representation of filter coefficients of IIR bandpass filter designed using the MOABC and in example 3A-2 are given in Table 6.18.

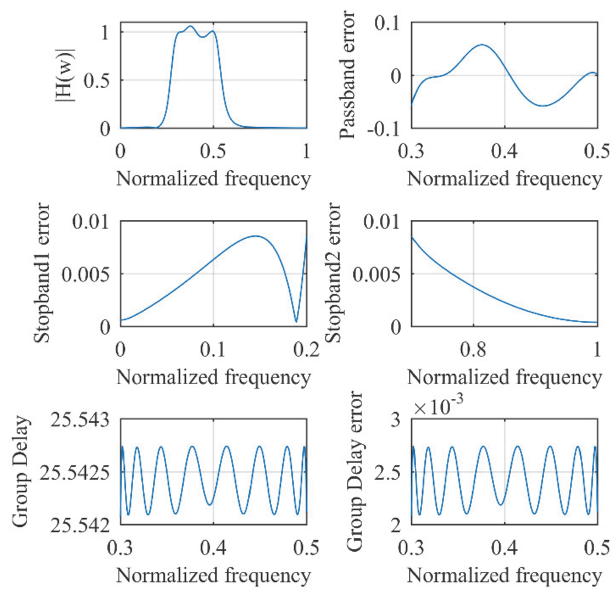


**Figure 6.7** Magnitude Response, Group Delay Response, Magnitude Errors and Group Delay Errors of IIR Bandpass Filter Designed Using MOABC

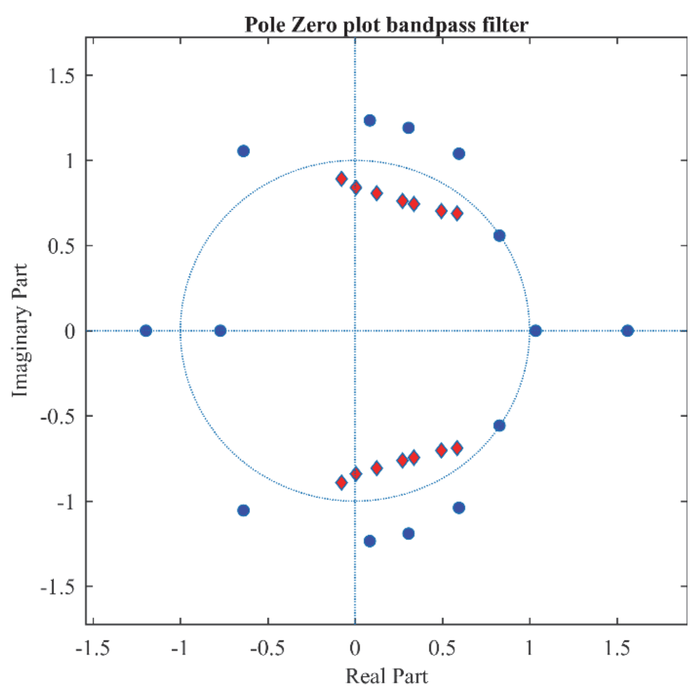


**Figure 6.8** Pole Zero Plot of IIR Bandpass Filter Designed Using MOABC

**Table 6.16** Poles and Zeros of IIR Bandpass Filter Using MOABC

| Poles | Zeros |
|---|---|
| 0.493894608581262 + 0.702518313262373i | 0.594972776318648 + 1.039563120258078i |
| 0.493894608581262 - 0.702518313262373i | 0.594972776318648 - 1.039563120258078i |
| 0.272178519664595 + 0.761216447502546i | -0.637641776079309 + 1.054933559385276i |
| 0.272178519664595 - 0.761216447502546i | -0.637641776079309 - 1.054933559385276i |
| 0.338827281370153 + 0.743658766655145i | 1.560937558290255 + 0.000000000000000i |
| 0.338827281370153 - 0.743658766655145i | 1.035267716101418 + 0.000000000000000i |
| 0.124170605143223 + 0.805942741602134i | 0.827351999477573 + 0.557693294518648i |
| 0.124170605143223 - 0.805942741602134i | 0.827351999477573 - 0.557693294518648i |
| -0.076643870392857 + 0.890878585508749i | 0.307609108908968 + 1.191403573851067i |
| -0.076643870392857 - 0.890878585508749i | 0.307609108908968 - 1.191403573851067i |
| 0.583831851732929 + 0.687760777910568i | 0.085275152227543 + 1.235273396048266i |
| 0.583831851732929 - 0.687760777910568i | 0.085275152227543 - 1.235273396048266i |
| 0.004906237972598 + 0.840010963297944i | -1.197694016499282 + 0.000000000000000i |
| 0.004906237972598 - 0.840010963297944i | -0.770605696420433 + 0.000000000000000i |
| $g_0$ | 0.002682636337256 |

**Table 6.17** Poles and Zeros of IIR Bandpass Filter in Example 3A-2 [75]

| Poles | Zeros |
|---|---|
| -0.076714645151686 + 0.890838519817448i | 1.561666032241332 |
| -0.076714645151686 - 0.890838519817448i | 1.034144738442276 |
| 0.004716638351661 + 0.839965105522936i | 0.828203881706730 + 0.558223703276560i |
| 0.004716638351661 - 0.839965105522936i | 0.828203881706730 - 0.558223703276560i |
| 0.583913983795629 + 0.687630980756487i | 0.595013662062041 + 1.039366803541748i |
| 0.583913983795629 - 0.687630980756487i | 0.595013662062041 - 1.039366803541748i |
| 0.494002366250057 + 0.702447038273360i | 0.307695507127380 + 1.191416490283998i |
| 0.494002366250057 - 0.702447038273360i | 0.307695507127380 - 1.191416490283998i |
| 0.124245297995792 + 0.805956367289334i | 0.085139886669244 + 1.234858171372786i |
| 0.124245297995792 - 0.805956367289334i | 0.085139886669244 - 1.234858171372786i |
| 0.338793525086818 + 0.743834836527864i | -0.638208918893237 + 1.054769047595507i |
| 0.338793525086818 - 0.743834836527864i | -0.638208918893237 - 1.054769047595507i |
| 0.272050461443023 + 0.761143525390037i | -1.199730733263683 |
| 0.272050461443023 - 0.761143525390037i | -0.769532635828309 |
| $g_0$ | 0.002680119463696 |

**Table 6.18** Filter Coefficients of IIR Bandpass Filter Designed Using MOABC and 3A-2 [75]

| Section no: | Coefficients | MOABC | 3A-2 |
|---|---|---|---|
| Section 1 | $b_{11}$ | -1.189945552637296 | -0.361935298977666 |
| | $b_{21}$ | 1.434684085561031 | -1.873578733973902 |
| | $a_{11}$ | -0.987789217162524 | -0.544100922888031 |
| | $a_{21}$ | 0.737463864854648 | 0.653350919812894 |
| Section 2 | $b_{12}$ | 1.275283552158618 | -0.264612102613937 |
| | $b_{22}$ | 1.519471849318864 | -0.795808126401440 |
| | $a_{12}$ | -0.544357039329189 | -0.248490595989497 |
| | $a_{22}$ | 0.653531626515206 | 0.665002560047421 |
| Section 3 | $b_{13}$ | -1.654703998955147 | 1.276417837786475 |
| | $b_{23}$ | 0.995533141790602 | 1.519848367920410 |
| | $a_{13}$ | -0.248341210286447 | -0.677587050174312 |
| | $a_{23}$ | 0.664962041922799 | 0.668071316676267 |
| Section 4 | $b_{14}$ | -0.170550304455085 | -0.615391014254765 |
| | $b_{24}$ | 1.533172214572047 | 1.514149778427029 |
| | $a_{14}$ | -1.167663703465858 | -0.009433276704000 |
| | $a_{24}$ | 0.813874518730050 | 0.705563625174430 |
| Section 5 | $b_{15}$ | 1.968299712919714 | -1.190027324124077 |
| | $b_{25}$ | 0.922949831683014 | 1.434324610345065 |
| | $a_{15}$ | -0.009812475945196 | -0.988004732497636 |
| | $a_{25}$ | 0.705642489631784 | 0.737470179437766 |
| Section 6 | $b_{16}$ | -2.596205274391673 | -0.170279773338488 |
| | $b_{26}$ | 1.615988260948077 | 1.532123503708188 |
| | $a_{16}$ | -0.677654562740306 | 0.15342929030334 |
| | $a_{26}$ | 0.667832287823740 | 0.799478405170950 |
| Section 7 | $b_{17}$ | -0.615218217817936 | -1.656407763413475 |
| | $b_{27}$ | 1.514065839668865 | 0.997535372573911 |
| | $a_{17}$ | 0.153287740785715 | -1.167827967592453 |
| | $a_{27}$ | 0.799538936986866 | 0.813791906168922 |
| | $b_0$ | 0.002682636337256 | 0.002680119463696 |

### *6.5 Conclusions*

In this chapter, the physical-programming-based multiobjective ABC algorithm is used to design cascade-form IIR filters. Three filter examples including IIR lowpass of order 10, IIR highpass filter of order 14 and IIR bandpass filter of order 14 are designed and the results are compared with the state-of-the-art design methods in [75]. The pole-zero plot of designed IIR filters have shown that all the poles are within the unit circle and the designed filters are stable. The physical-programming-based multiobjective ABC algorithm can be used to design IIR filter problems, which is a non-convex optimization problem requiring simultaneous optimization of both magnitude and phase responses. The proposed design method can achieve slightly better or comparable results in terms of peak errors in passband and stopband, and peak group delay error in passband when compared to other design methods in [75].

# CHAPTER 7

# CONCLUSIONS AND FUTURE DIRECTIONS

In this dissertation, the ability of the ABC algorithm in handling multimodal and nondifferentiable problems is utilized to design digital filters. Single-objective optimization has been extended into multiobjective space for simultaneous optimization of magnitude and/or phase characteristics. Different types of digital filters such as Types 3 and 4 linear phase FIR filters, and sparse Type 1 linear phase FIR filters are designed using the proposed ABC-AMR algorithm. In this chapter, conclusions of this dissertation and suggestions for future work are presented.

## *7.1 Conclusions*

In Chapter 3, an improved ABC algorithm called the ABC-AMR algorithm is proposed and used to design Types 3 and 4 linear phase FIR differentiators. The original ABC algorithm has certain shortcomings due to an insufficient balance between exploration and exploitation in the search equation, which in turn increases the convergence time in proportional to the number of parameters of a problem. In the ABC-AMR algorithm, instead of changing only one parameter in employed and onlooker bee phase, several new food locations are generated in every iteration. A self-adaptive control parameter known as the adaptive modification rate ($AMR$), is introduced which adaptively controls the number of parameters to be changed in each iteration. The $AMR$ ensures exploration in initial stages and exploitation in later stages of optimization. The ABC-AMR algorithm is used to design linear phase Type 3 and Type 4 linear phase FIR differentiators. Given the desired amplitude response $A_d(w)$ of a differentiator, the optimization process searches for an optimal coefficient vector $\boldsymbol{c}$ that minimizes the weighted minimax objective function given in equation 3.16. Since linear phase FIR filters have constant group delay, optimization is formulated as a single-objective optimization problem. Results are compared, in terms of minimax and peak errors in passband and stopband, iteration time and converged iteration number, with respect to the Parks-McClellan (PM) technique, the original ABC algorithm and its three variants namely the gbest-guided ABC, the best-so-far selection ABC and the global best ABC. Results indicate that differentiators can be designed using the proposed ABC-AMR algorithm to reach the lowest minimax errors and the lowest peak errors with reduced computational time. Unlike other variants of ABC algorithm, which directs search towards best solutions in the objective space, the ABC-AMR algorithm explores unknown regions of objective space as well as exploits neighborhood regions of best solution.

In Chapter 4, sparse FIR filters are designed using the constrained ABC-AMR algorithm. Compared to conventional FIR filters, design of sparse filters aims at reducing the number of nonzero coefficients and thereby decreasing the implementation cost by removing the multiplier units associated with zero coefficients. When using traditional $l_0$ norm-based optimization, obtaining a global optimum solution is difficult as $l_0$ norm is a highly non-convex problem. The ABC-AMR algorithm is combined with iterative shrinkage algorithm and $l_0$ norm to design sparse filters. The optimization algorithm aims for finding the positions of zero value coefficients and the filter is designed using constrained minimax objective function. To evaluate the performance of sparse filter design, the proposed algorithm is compared to other design methods such as minimum-increase method, smallest coefficient method and partial $l_1$ optimization. Results indicate that the proposed method can achieve better sparsity and lower peak errors.

In Chapter 5, the reference-based MOABC is used in the design of asymmetric FIR filters. Objective functions are formulated for magnitude responses in passband(s) and stopband(s) and for group delay response(s) in passband(s), and all objective functions are simultaneously optimized. Preferences are incorporated using the reference-point approach in which the solutions are ranked according to their normalized Euclidean distances from the reference point. Asymmetric lowpass, highpass, bandpass and bandstop FIR filters are designed, and the results are compared to the corresponding multiobjective differential evolution algorithm. The proposed method can result in lower peak errors in stopband and passband magnitudes as well as lower peak error(s) in passband group delay(s).

In Chapter 6, IIR filters are designed using the physical-programming-based MOABC algorithm. The design objective is to find an optimal coefficient vector $\boldsymbol{c}$, which minimizes magnitude error $e_{mag}(\boldsymbol{c})$ and group delay error $e_{gd}(\boldsymbol{c})$ such that the designed filter is the best approximation as per specifications. IIR filter design is a non-convex problem with many local optima on error surfaces, and hence multiobjective approach is utilized for optimization. The preferences of a decision maker are incorporated into optimization process *a priori* using the physical programming approach with different degrees of desirability. The size and diversity of the external archive are maintained using spherical pruning technique, which selects solutions with the lowest physical index from each spherical sector. Using the proposed method, IIR lowpass, highpass and bandpass filters are designed. The design results indicate better peak magnitude error at a small increase in peak group delay error can be obtained for each of lowpass, highpass and bandpass filters, in additional the bandpass filter also results in a small increase in stopband 1 peak magnitude error. Overall, the performance is slightly better or close to those of the state-of-the-art design methods in [75].

## *7.2 Suggestions for Future Work*

In this dissertation, improvements are applied to the original ABC algorithm for the design of various types of FIR and IIR filters and the results indicate that the proposed method can often achieve better results. Continue a future research along this direction would improve the performance of the ABC algorithm in digital filter design applications. A few topics for future study are briefly discussed below.

### *7.2.1 2-D Filter Design*

This dissertation work mainly focuses in the design of 1-D filters: Chapter 3 – 1-D linear phase FIR differentiator, Chapter 4 – 1-D sparse linear phase FIR filters, Chapter 5 -1-D asymmetric FIR filter, and Chapter 6 – 1-D IIR filters. These design methodologies can be extended to various designs of 2-D digital filters for image processing and other applications. Frequency response of 2-D FIR filter with impulse response $\mathrm{h}(n_1, n_2)$ can be expressed as [1],[112]-[114],

$$H(\omega_1, \omega_2) = \sum_{n_1=0}^{N_1-1} \sum_{n_2=0}^{N_2-1} \mathrm{h}(n_1, n_2). e^{-j(n_1\omega_1 + n_2\omega_2)}$$

$$M(\omega_1, \omega_2) = |H(\omega_1, \omega_2)| e^{j\theta(\omega_1, \omega_2)} \tag{7.1}$$

Minimax error is defined by,

$$\sum_{j=1}^{m_1} \sum_{k=1}^{m_2} \left[ \mathrm{W}(\omega_{1j}, \omega_{2k}) \left| |H(\boldsymbol{c}_t, \omega_{1j}, \omega_{2k})| - \mathrm{D}(\omega_{1j}, \omega_{2k}) \right|^p \right]^{\frac{1}{\mathrm{p}}} \tag{7.2}$$

where $D$ is the desired magnitude response and $\boldsymbol{c}_t$ is the filter coefficient vector.

$$\mathrm{D}(\omega_1, \omega_2) = \begin{cases} 1 & \sqrt{(\omega_{1k}^2 + \omega_{2k}^2)} \le \omega_p \\ 0 & \sqrt{(\omega_{1k}^2 + \omega_{2k}^2)} \ge \omega_s \end{cases} \tag{7.3}$$

When evolutionary algorithms, such as genetic algorithm, is applied to 2-D filter design applications, they suffer from premature convergence and get stuck at local optimum. As demonstrated for 1-D filter design, the ABC-AMR can be used to overcome these shortcomings.

### *7.2.2 Implicit Preference-Based Multiobjective ABC*

In multiobjective optimization, a set of optimal solutions known as the Pareto front is generated in the objective space instead of a single optimum solution. In practice, selecting a single optimum solution from the Pareto front containing many optimal solutions is difficult. Incorporating the preferences of a decision maker into optimization by providing some higher level information will guide the search towards a region of interest in the objective space. Preferences can be incorporated both explicitly and implicitly, and this dissertation focuses on two explicit preference-based approaches which can be extended to implicit methodology such as knee region and nadir point.

Knee regions are potential part of the Pareto front representing maximal trade-offs between the objectives [191]. Knee region corresponds to maximum bulge in convex and concave parts of the Pareto front of minimization and maximization multiobjective problems respectively. In minimization problems, knee points are defined as the farthest solution from the extreme line, where the value of objective function is minimum. In contrast to explicit methods such as the physical programming and the reference point technique, the reference point in the knee-based approach is picked from the first Pareto front. The decision maker has no *a priori* information regarding the number of knee regions in the Pareto optimal front, and in this case, preferences need to be set interactively [192]. The extreme line is defined using the extreme solutions in the Pareto front. Distance of each solution from this extreme line is calculated and then searches for the farthest solutions situated in the convex parts of the Pareto front. Distance from a given solution $P(x_p, y_p)$ to the extreme line $L: ax + by + c$ can be defined as,

$$d(P,L) = \begin{cases} \frac{|ax_P + by_P + c|}{\sqrt{a^2 + b^2}} & \text{if } ax_p + by_p + c < 0 \\ -\frac{|ax_P + by_P + c|}{\sqrt{a^2 + b^2}} & \text{otherwise} \end{cases} \tag{7.4}$$

In a knee-based multiobjective minimization problem, only convex regions are encouraged, and concave regions are discarded.

The nadir point $\boldsymbol{z}_m^N$ is a vector composed with the worst objective value over the Pareto optimal front *P,* for minimization problem,

$$\boldsymbol{z}_m^N = \max_{x \in P} f_m(\boldsymbol{x}), \quad m \in \{1,2, \dots \dots \dots \dots \dots M\} \tag{7.5}$$

where $M$ is the total number of objective functions [193].

The decision maker could use the nadir point as a form of implicit decision maker's preference. For constructing the nadir points, the Pareto optimal is first sorted from maximum to minimum based on each objective function value. Solutions closer to the extreme objective vector gets a higher rank compared to intermediate solutions. Solutions with the worst objective function value in each generation is defined as the reference points, which gets updated in every iteration [194]. The nadir point estimation has certain advantages such as maintaining objective space diversity, and ease of finding extreme points. These implicit methods can be incorporated into the ABC-AMR to set preferences for a future study.

## REFERENCES


[1] H. K. Kwan, Global Optimization Algorithms and Design Applications, dfisp.org, edition 1.2, Mar. 13, 2018 (403 pages). ISBN-13: 978-1-988-30704-6.

[2] A. Antoniou, Digital Filters: Analysis, Design, and Signal Processing Applications, Toronto, New York, London: McGraw-Hill, 2018 (952 pages). ISBN 978 0-07-184603-5.

[3] H. K. Kwan, Multimedia Signal Processing and Applications, dfisp.org, edition 1.3, May 19, 2018 (717 pages). ISBN-13: 978-1-988-30703-9.

[4] V. Cappellini, A. G. Constantinides, and P. Emiliani, Digital Filters and Their Applications, London, New York: Academic Press, 1978 (393 pages). ISBN 0121592502.

[5] W. S. Lu and A. Antoniou, Two-Dimensional Digital Filters. New York, Hong Kong: Marcel Dekker, 1992 (398 pages). ISBN 0-8247-8434-0.

[6] B. A. Shenoi, Magnitude and Delay Approximation of 1-D and 2-D Digital Filters, Berlin, Heidelberg, New York: Springer-Verlag, 1999 (250 pages), ISBN 3-540-64161-0.

[7] R. King, M. Ahmadi, R. Gorgui-Naguib, A. Kwabwe, and M. Azimi-Sadjadi, Digital Filtering in One and Two Dimensions, New York, London: Plenum Press, 1989 (466 pages). ISBN 0-306-42976-4.

[8] S. K. Mitra, Digital Signal Processing, New York: McGraw-Hill, 4th edition, Sep. 10, 2010 (992 pages). ISBN-13: 978-0-073-38049-0.

[9] P. S. R. Diniz, E. A. B. da Silva, and S. L. Netto, Digital Signal Processing: System Analysis and Design, London: Cambridge University Press, 2nd edition, Aug. 2010 (889 pages). ISBN-13: 978-0-521-88775-5.

[10] Y. C. Lim, H. K. Kwan, and W. C. Siu, Eds., Trends in Digital Signal Processing - A Festschrift in Honour of A. G. Constantinides (Pan Stanford Series on Digital Signal Processing - Volume 1), Singapore: Pan Stanford, Jul. 31, 2015 (Hardback, 577 pages). Print ISBN: 9789814669504, eBook ISBN: 9789814669511, DOI: 10.4032/9789814669511.

[11] J. McClellan, T. Parks, and L. Rabiner, "A computer program for designing optimum FIR linear phase digital filters," IEEE Transactions on Audio and Electroacoustics, vol. 21, no. 6, pp. 506-526, Dec 1973.

[12] C. A. Rahenkamp and B. Kumar, "Modifications to the McClellan, Parks, and Rabiner computer program for designing higher order differentiating FIR filters," IEEE Transactions on Acoustics, Speech and Signal processing, vol. 34, no. 6, pp. 1671-1674, Dec. 1986.

[13] S. C. Pei and J. J. Shyu, "Eigenfilter design of higher-order digital differentiators," IEEE Transactions on Acoustics, Speech and Signal processing, vol. 37, no. 4, pp. 505-511, Apr. 1989.

[14] S. Sunder and R. P. Ramachandran, "Least-squares design of higher order nonrecursive differentiators," IEEE Transactions on Signal Processing, vol. 42, no. 4, pp. 956-961, Apr. 1994.

[15] H. K. Kwan and R. Raju, "Minimax design of linear phase FIR differentiators using artificial bee colony algorithm," in Proc. of 8th International Conference on Wireless Communications and Signal Processing (WCSP 2016), Yangzhou, China, Oct. 13-15, 2016, pp. 1-4.

[16] H. K. Kwan and M. Zhang, "Minimax design of linear phase FIR Hilbert transformer using teaching-learning-based optimization," in Proc. of 8th International Conference on Wireless Communications and Signal Processing (WCSP 2016), Yangzhou, Jiangsu, China, Oct. 13-15, 2016, pp. 1-4.

[17] H. K. Kwan and J. Liang, "Design of linear phase FIR filters using cuckoo search algorithm," in Proc. of 8th International Conference on Wireless Communications and Signal Processing (WCSP 2016), Yangzhou, Jiangsu, China, Oct. 13-15, 2016, pp. 1-4.

[18] G. S. Mollova and R. Unbehauen, "Analytical design of higher-order differentiators using least-squares technique," Electronics Letters, vol. 37, no. 17, p. 1098–1099, Aug. 2001.

[19] M. A. Al-Alaoui, "Novel approach to designing digital differentiators," Electronics Letters, vol. 28, no. 15, pp. 1376-1378, Jul. 1992.

[20] M. Gupta, M. Jain, and B. Kumar, "Novel class of stable wideband recursive digital integrators and differentiators," IET signal processing, vol. 4, no. 5, pp. 560-566, Oct. 2010.

[21] M. Gupta, B. Relan, R. Yadav, and V. Aggarwal, "Wideband digital integrators and differentiators designed using particle swarm optimization," IET Signal Processing, vol. 8, no. 6, pp. 668-679, Aug. 2014.

[22] M. Jain, M. Gupta, and N. Jain, "Linear phase second order recursive digital integrators and differentiators," Radio Engineering, vol. 21, no. 2, pp. 712-717, Jun. 2012.

[23] A. Alaoui and M. Baydoun, "Novel wide band digital differentiators and integrators using different optimization techniques," in Proc. of International Symposium on Signals, Circuits and Systems ISSCS2013, Iasi, Romania, Jul. 11-12, 2013, pp. 1-4.

[24] T. Baran, D. Wei, and A. Oppenheim, "Linear programming algorithms for sparse filter design," IEEE Transactions on Signal Processing, vol. 58, no. 3, pp. 1605-1617, Mar. 2010.

[25] A. Jiang and H. K. Kwan, "Efficient design of sparse FIR filters in WLS sense," in Proc. of IEEE International Symposium on Circuits and Systems (ISCAS), Seoul, Korea, May 20-23, 2012, pp. 41-44.

[26] A. Jiang and H. K. Kwan, "WLS design of sparse FIR digital filters," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 60, no. 1, pp. 125-135, Jan. 2013.

[27] A. Jiang, H. K. Kwan, and Y. Zhu, "Peak-error-constrained sparse FIR filter design using iterative SOCP," IEEE Transactions on Signal Processing, vol. 60, no. 8, pp. 4035-4044, Aug. 2012.

[28] D. Wei, C. Sestok, and A.V. Oppenheim, "Sparse filter design under a quadratic constraint: Low-complexity algorithms," IEEE Transactions on Signal Processing, vol. 61, no. 4, pp. 857-870, Feb. 2013.

[29] D. Wei and A. Oppenheim, "A branch-and-bound algorithm for quadratically-constrained sparse filter design," IEEE Transactions on Signal Processing, vol. 61, no. 4, pp. 1006-1018, Feb. 2013.

[30] A. Jiang, H. K. Kwan, Y. Zhu, X. Liu, N. Xu, and Y. Tang, "Design of sparse FIR filters with joint optimization of sparsity and filter order," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 62, no. 1, pp. 195-204, Jan. 2015.

[31] A. Jiang and H. K. Kwan, "Recent advances in sparse FIR filter design using l0 and l1 optimization techniques," in "Trends in Digital Signal Processing - A Festschrift in Honour of A. G. Constantinides" (Pan Stanford Series on Digital Signal Processing - Vol. 1), Y. C. Lim, H. K. Kwan, W.-C. Siu, Eds., Singapore: Pan Stanford, Jul. 2015, ch. 5, pp. 145-176.

[32] W. Ye and Y. J. Yu, "Greedy algorithm for the design of linear-phase FIR filters with sparse coefficients," Circuits, Systems, and Signal Processing, vol. 35, no. 4, pp. 1427-1436, Apr. 2016.

[33] L. Zheng, A. Jiang, and H. K. Kwan, "Sparse FIR filter design via partial $L1$ optimization," in Proc. of IEEE International Symposium on Circuits and Systems (ISCAS), Baltimore, MD, USA, May 28-31, 2017, pp. 1-4.

[34] A. Jiang, H. K. Kwan, Y. Tang, and Y. Zhu, "Sparse FIR filter design via partial 1-norm optimization," IEEE Transactions on Circuits and Systems II: Express Briefs (Early Access), Aug. 26, 2019, DOI: 10.1109/TCSII.2019.2937343

[35] Z. Feng, K. Yiu, and S. Wu, "Design of sparse filters by a discrete filled function technique," Circuits, Systems, and Signal Processing, vol. 37, no. 10, p. 4279–4294, Oct. 2018.

[36] W. Chen, M. Huang, and X. Lou, "Design of sparse FIR filters with reduced effective length," IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 66, no. 4, pp. 1496-1506, Apr. 2019.

[37] R. Raju, H. K. Kwan, and A. Jiang, "Sparse FIR filter design using artificial bee colony algorithm," in Proc. of IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS 2018), Windsor, Ontario, Canada, Aug. 2018, pp. 956-959.

[38] H. K. Kwan, J. Liang, and A. Jiang, "Sparse FIR filter design using iterative MOCSA," in Proc. of IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS 2018), Windsor, Ontario, Canada, Aug. 2018, pp. 952-955.

[39] J. Liang, H. K. Kwan, and A. Jiang, "Sparse linear phase FIR filter design using iterative CSA," in Proc. of IEEE 23rd International Conference on Digital Signal Processing (DSP 2018), Shanghai, China, Nov. 2018, pp. 1-4.

[40] Y. Neuvo, D. C. Yu, and S. Mitra, "Interpolated finite impulse response filters," IEEE Transactions on Acoustics, Speech and Signal Processing, vol. 32, no. 3, pp. 563–570, Jun. 1984.

[41] T. Saramaki, Y. Neuvo, and S. K. Mitra, "Design of computationally efficient interpolated FIR filters," IEEE Transactions on Circuits and Systems, vol. 35, no. 1, pp. 70-88, Jan. 1988.

[42] Y. C. Lim and B. Liu, "Extrapolated impulse response FIR filters," IEEE Transactions on Circuits and Systems, vol. 37, no. 12, pp. 1548-1551, Dec. 1990.

[43] J. Liang and H. K. Kwan, “Sparse 2-D FIR filter design using iterative CSA with step-descendant coefficient thresholding,” in Proc. of 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP 2018), Shanghai, China, Nov. 2018, pp. 1-4.

[44] Y. C. Lim, “Frequency-response masking approach for the synthesis of sharp linear phase digital filters,” IEEE Transactions on Circuits and Systems, vol. 33, no. 4, pp. 357-364, Apr. 1986.

[45] W. S. Lu and T. Hinamoto, “Optimal design of frequency-response masking filters using semidefinite programming,” IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications, vol. 50, no. 4, pp. 557-568, Apr. 2003.

[46] H. Wang, L. Z. H. Wang, S. Fang, P. Liu and X. Du, “An optimal design of the extrapolated impulse response filter with analytical solution,” Signal Processing, vol. 92, no. 7, pp.1665-1672, Jul. 2012.

[47] W. Adams, “FIR digital filters with least-squares stopbands subject to peak-gain constraints,” IEEE Transactions on Circuits and Systems, vol. 38, no. 4, pp. 376-388, Apr. 1991.

[48] J. L. Sullivan and J. W. Adams, “A new nonlinear optimization algorithm for asymmetric FIR digital filters,” in Proc. of IEEE International Symposium on Circuits and Systems (ISCAS), London, May 30 – Jun. 21994, vol.2, pp. 541-544.

[49] M. Lang and J. Bamberger, “Nonlinear phase FIR filter design according to the L2 norm with constraints for the complex error,” Signal Processing, vol. 36, no. 1, pp. 31-40, Mar. 1994.

[50] R. Raju and H. K. Kwan, “FIR filter design using multiobjective artificial bee colony algorithm,” in Proc. of 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE 2017), Windsor, Ontario, Canada, Apr. 30-May 3, 2017, pp. 1-4.

[51] J. Liang and H. K. Kwan, "FIR filter design using multiobjective cuckoo search algorithm," in Proc. of 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE 2017), Windsor, Ontario, Canada, Apr. 30-May 3, 2017, pp. 1-4.

[52] M. Zhang and H. K. Kwan, "FIR filter design using multiobjective teaching-learning-based optimization," in Proc. of 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE 2017), Windsor, Ontario, Canada, Apr. 30-May 3, 2017, pp. 1-4.

[53] M. Lang and J. Bamberger, "Nonlinear phase FIR filter design with minimum LS error and additional constraints," in Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, Minneapolis, MN, USA, Apr. 27-30, 1993, pp. 57-60.

[54] S. S. Kidambi and R. P. Ramachandran, "Design of nonrecursive filters satisfying arbitrary magnitude and phase specifications using a least-squares approach," IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing, vol. 42, no. 11, pp. 711-716, Nov. 1995.

[55] M. C. Lang, "An iterative reweighted least squares algorithm for constrained design of nonlinear phase FIR filters," in Proc. of 1998 IEEE International Symposium on Circuits and Systems (ISCAS), Monterey, CA, May 31-Jun. 3, 1998, pp. 367-370, vol.5.

[56] S. U. Ahmad and A. Antoniou, "A multiobjective genetic algorithm for asymmetric FIR filters," in Proc. of IEEE International Symposium on Signal Processing and Information Technology, Giza, Egypt, Dec. 15-18, 2007, pp. 525-530.

[57] H. K. Kwan and A. Jiang, "Peak-constrained WLS strategy for FIR digital filter design," in Proc. of IEEE International Symposium on Circuits and Systems, New Orleans, Louisiana, U.S.A., May 27-30, 2007, pp. 2303-2306.

[58] H. K. Kwan and M. T. Tsim, "High speed 1-D FIR digital filtering architectures using polynomial convolution," in Proc. of 1987 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Dallas, Texas, USA, Apr. 6-9, 1987, pp. 1863-1866.

[59] H. K. Kwan, "Unconstrained asymmetric FIR filter design using harmony search algorithm," in Proc. of 2018 IEEE 31th Canadian Conference on Electrical and Computer Engineering (CCECE 2018), Quebec City, Quebec, Ontario, Canada, May 13-16, 2018, pp. 1-4.

[60] H. K. Kwan, "Asymmetric FIR filter design using evolutionary optimization," in Proc. of 2017 IEEE 30th Canadian Conference on Electrical and Computer Engineering (CCECE 2017), Windsor, Ontario, Canada, Apr. 30-May 3, 2017, pp. 1-4.

[61] H. K. Kwan, "On the problem of designing IIR digital filters with short coefficient word lengths," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 27, no. 6, pp. 620-624, Dec. 1979.

[62] H. K. Kwan, "Poles and zeros determination of arbitrary linear digital networks," IEEE Transactions on Circuits and Systems, vol. 30, no. 12, pp. 919-921, Dec. 983.

[63] H. K. Kwan, "Amplitude scaling in arbitrary linear digital networks," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 32, no. 6, pp. 1240-1242, Dec.1984.

[64] H. K. Kwan and A. G. Constantinides, "Roundoff noise in arbitrary linear digital networks," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 33, no. 3, pp. 734-737, Jun. 1985.

[65] H. Zhao, H. K. Kwan, L. Nie, and L. Wan, "A quick design of 1-D stable IIR filters using finite impulse response fitting," in Proc. of International Conference on Communications, Circuits and Systems, Guilin, China, Jun. 25-28, 2006, pp. 176-180.

[66] H. K. Kwan, "Linear phase equiripple IIR digital filter design," in Proc. of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2003), Hong Kong, China, vol. 6, Apr. 6-10, 2003, pp. 17-20.

[67] H. K. Kwan and C. L. Chan, “Design of linear phase circularly symmetric two-dimensional recursive digital filters,” IEEE Transactions on Circuits and Systems, vol. 36, no. 7, pp. 1023-1029, Jul. 1989.

[68] H. K. Kwan and A. Jiang, “Recent advances in FIR approximation by IIR digital filters,” in Proc. of International Conference on Communications, Circuits and Systems (ICCCAS 2006), Guilin, China, Jun. 25-28, 2006, pp. 185-190.

[69] H. K. Kwan and F. Wang, “State-space approaches for model reduction of FIR digital filters,” in Proc. of International Conference on Communications, Circuits and Systems, Guilin, China, Jun. 25- 28, 2006, pp. 216-221.

[70] F. Wang and H. K. Kwan, “IIR digital filter design via orthogonal projection of singular perturbational model reduction,” in Proc. of IEEE International Symposium on Circuits and Systems, Washington, Seattle, USA, May 18-21, 2008, pp. 1132-1135.

[71] A. Jiang and H. K. Kwan, “IIR digital filter design with new stability constraint based on argument principle,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 56, no. 3, pp. 583-593, Mar. 2009.

[72] A. Jiang and H. K. Kwan, “Minimax design of IIR digital filters using SDP relaxation technique,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 2, pp. 378-390, Feb. 2010.

[73] A. Jiang and H. K. Kwan, “Minimax design of IIR digital filters using iterative SOCP,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 57, no. 6, pp. 1326-1337, Jun. 2010.

[74] X. Lai, Z. Lin, and H. K. Kwan, “A sequential minimization procedure for minimax design of IIR filters based on second-order factor updates,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 58, no. 1, pp. 51-55, Jan. 2011.

[75] R. C. Nongpiur, D. J. Shpak, and A. Antoniou, “Improved design method for nearly linear-phase IIR filters using constrained optimization,” IEEE Transactions on Signal Processing, vol. 61, no. 4, pp. 895-906, Feb. 2013.

[76] A. Jiang, H. K. Kwan, Y. Zhu, N. Xu, and X. Liu, "Efficient WLS design of IIR digital filters using partial second-order factorization," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 63, no. 7, pp. 703-707, Jul. 2016.

[77] H. K. Kwan, "Group delay equalizer design using harmony search algorithm," in Proc. of IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS 2018), Windsor, Ontario, Canada, Aug. 2018, pp. 688-691.

[78] H. K. Kwan, "Allpass group delay equalizer design using harmony search algorithm," in Proc. of 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP 2018), Shanghai, China, Nov. 2018, pp. 1-4.

[79] Y. Wang, B. Li, and Z. Li, "Fixed-point digital IIR filter design using multi-objective optimization evolutionary algorithm," in IEEE Youth Conference on Information, Computing and Telecommunications, Beijing, China, Nov. 28-30, 2010, pp. 174-177.

[80] Y. Wang, B. Li, and Y. Chen, "Digital IIR filter design using multi-objective optimization evolutionary algorithm," Applied Soft Computing, vol. 11, no. 2, pp. 1851-1857, Mar. 2011

[81] R. Raju and H. K. Kwan, "IIR filter design using multiobjective artificial bee colony algorithm," in Proc. of 2018 IEEE 31th Canadian Conference on Electrical and Computer Engineering (CCECE 2018), Quebec City, Quebec, Ontario, Canada, May 13-16, 2018, pp. 1-4.

[82] H. K. Kwan and M. Zhang, "IIR filter design using multiobjective teaching learning-based optimization," in Proc. of 2018 IEEE 31th Canadian Conference on Electrical and Computer Engineering (CCECE 2018), Quebec City, Quebec, Ontario, Canada, May 13-16, 2018, pp. 1-4.

[83] J. Liang and H. K. Kwan, "IIR filter design using constrained multiobjective cuckoo search algorithm," in Proc. of 2018 IEEE 31th Canadian Conference on Electrical and Computer Engineering (CCECE 2018), Quebec City, Quebec, Ontario, Canada, May 13-16, 2018, pp. 1-4.

[84] K. M. Tsui, S. C. Chan, and H. K. Kwan, "A new method for designing causal stable IIR variable fractional delay digital filters," IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 54, no. 11, pp. 999-1003, Nov. 2007.

[85] S. C. Chan, K. M. Tsui, and H. K. Kwan, "A new method for designing constrained causal stable IIR variable digital filters," in Proc. of 2006 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), Singapore, Dec. 2006, pp. 65-68.

[86] H. K. Kwan, "Importance of feedback gain in adaptive IIR digital filtering," Electronics Letters, vol. 29, no. 12, pp. 1142-1144, Jun. 1993.

[87] H. K. Kwan, "Adaptive IIR digital filters with saturation outputs for noise and echo cancellation," Electronics Letters, vol. 38, no. 13, pp. 661-663, Jun. 20, 2002.

[88] F. Albu and H. K. Kwan, "Combined echo and noise cancellation based on Gauss-Seidel pseudo affine projection algorithm," in Proc. of 2004 IEEE International Symposium on Circuits and Systems (ISCAS), Vancouver, BC, Canada, May 23-26, 2004, vol. 3, pp. 505-508.

[89] F. Albu and H. K. Kwan, "Fast block exact Gauss-Seidel pseudo affine projection algorithm," Electronics Letters, vol. 40, no. 22, pp. 1451-1453, 28 Oct. 2004.

[90] F. Albu and H. K. Kwan, "A new block exact affine projection algorithm," in Proc. of 2005 IEEE International Symposium on Circuits and Systems (ISCAS), Kobe, Japan, May 23-26, 2005, pp. 4337-4340.

[91] J. Thevaril and H. K. Kwan, "Speech enhancement using adaptive neuro-fuzzy filtering," in Proc. of 2005 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Hong Kong, China, Dec. 13-16, 2005, pp. 753-756.

[92] F. Albu and H. K. Kwan, "Memory improved proportionate affine projection sign algorithm," Electronics Letters, vol. 48, issue 20, pp. 1279-1281, 27 Sept. 2012.

[93] F. Albu and H. K. Kwan, "New proportionate affine projection sign algorithms," in Proc. of 2013 IEEE International Symposiumon Circuits and Systems (ISCAS), Beijing, China, May 19-23, 2013, pp. 521-524.

[94] H. Zhao and H. K. Kwan, “Design of 1-D stable variable fractional delay IIR filters,” IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 54, no. 1, pp. 86-90, Jan. 2007.

[95] H. Zhao and H. K. Kwan, “Design of 1-D stable variable fractional delay IIR filters,” in Proc. of 2005 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS), Hong Kong, China, Dec. 2005, pp. 517-520.

[96] H. Zhao, H. K. Kwan, L. Wan, and L. Nie, “Design of 1-D stable variable fractional delay IIR filters using finite impulse response fitting,” in Proc. of International Conference on Communications, Circuits and Systems (ICCCAS), Guilin, China, Jun. 25-28, 2006, pp. 201-205.

[97] H. K. Kwan and A. Jiang, “FIR, allpass, IIR variable fractional delay digital filter design,” IEEE Transactions on Circuits and Systems I: Regular Papers, vol. 56, no. 9, pp. 2064-2074, Sep. 2009.

[98] H. K. Kwan and A. Jiang, “Integrated design of IIR variable fractional delay digital filters with variable and fixed denominators,” in “Digital Filters”, Fausto Pedro García Márquez, Ed., London: InTech, Apr. 2011, chapter 8, pp. 179-208, ISBN-13: 978-953-307190-9.

[99] H. K. Kwan, “Design of passive second-order digital filters,” in Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 1982), Paris, France, May 3-5, 1982, vol. 1, pp. 286-289.

[100] H. K. Kwan, “One-dimensional passive state-space digital filter design,” in Proc. of IEEE International Symposium on Circuits and Systems (ISCAS 1984), Montreal, Quebec, Canada, May 7-10, 1984, vol. 1, pp. 250-253.

[101] H. K. Kwan, “A multi-output first-order digital filter structure for VLSI implementation,” IEEE Transactions on Circuits and Systems, vol. 32, no. 9, pp. 973-974, Sept. 1985.

[102] H. K. Kwan, "A multi-output second-order digital filter structure for VLSI implementation," IEEE Transactions on Circuits and Systems, vol. 32, no. 1, pp. 108-109, Jan. 1985.

[103] H. K. Kwan, "A multi-output second-order digital filter without limit cycle oscillations," IEEE Transactions on Circuits and Systems, vol. 32, no. 9, pp. 974-975, Sept. 1985.

[104] H. K. Kwan, "A multi-output wave digital biquad using magnitude truncation instead of controlled rounding," IEEE Transactions on Circuits and Systems, vol. 32, no. 11, pp. 1185-1187, Nov. 1985.

[105] H. K. Kwan, "Tunable and variable passive digital filters for multimedia signal processing," in Proc. of IEEE International Symposium on Intelligent Multimedia, Video & Speech Processing (ISIMP 2001), Hong Kong, China, May 2-4, 2001, pp. 229-232.

[106] H. K. Kwan, "Variable multi-output passive digital filters," in Proc. 2005 IEEE International Symposium on Circuits and Systems (ISCAS), Kobe, Japan, May 23-26, 2005, pp. 3725-3728.

[107] H. K. Kwan, "Tunable bandpass passive digital filter design," in Proc. 2002 IEEE 45th International Midwest Symposium on Circuits and Systems (MWSCAS), Tulsa, Oklahoma, U.S.A., vol. 3, Aug. 4-7, 2002, pp. 481-484.

[108] H. K. Kwan, "High-order tunable passive digital filters," in Proc. of IEEE International Symposium on Circuits and Systems (ISCAS 2002), Scottsdale, Arizona, U.S.A, vol. II, May 26-29, 2002, pp. 700-703.

[109] H. K. Kwan, "Multi-output passive digital filters," in Proc. 2005 IEEE International Symposium on Circuits and Systems (ISCAS), Kobe, Japan, May 23-26, 2005, vol. 3, pp. 2595-2598.

[110] H. K. Kwan, "Two-dimensional passive state-space digital filter design," in Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing, San Diego, California, USA, Mar. 19-21, 1984, vol. 2, pp. 2071-2074.

[111] H. K. Kwan and C. L. Chan, “Circularly symmetric two-dimensional multiplierless FIR digital filter design using an enhanced McClellan transformation,” IEE Proc. Part G on Circuits, Devices and Systems, vol. 136, no. 3, pp. 129-134, Jun. 1989.

[112] H. K. Kwan and M. Zhang, “2-D linear phase FIR digital filter design using teaching-learning-based optimization,” in Proc. of 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP 2018), Shanghai, China, Nov. 2018, pp. 1-4.

[113] H. K. Kwan and M. Zhang, “2-D FIR digital filter design using teaching-learning-based optimization,” in Proc. of IEEE 61st International Midwest Symposium on Circuits and Systems (MWSCAS 2018), Windsor, Ontario, Canada, Aug. 2018, pp. 684-687.

[114] X. Lai and H. K. Kwan, “An online procedure for linear-phase 2-D FIR filters of smallest size with magnitude error constraint,” in Proc. of IEEE International Symposium on Circuits and Systems, New Orleans, Louisiana, USA, May 27-30, 2007, pp. 2970-2973.

[115] H. K. Kwan and C. L. Chan, “Multidimensional spherically symmetric recursive digital filter design satisfying prescribed magnitude and constant group delay responses,” IEE Proc. Part G on Electronic Circuits and Systems, vol. 134, no.4, pp. 187-193, Aug. 1987.

[116] H. K. Kwan and C. L. Chan, “Design of two-dimensional circularly symmetric IIR digital low-pass filters by spectral transformation,” in Proc. of IEEE Asian Electronics Conference (AEC 1987), Hong Kong, Sep. 1-6, 1987, pp. 29-33.

[117] C. L. Chan and H. K. Kwan, “Simple and computationally efficient design of two-dimensional circularly symmetric IIR digital filters,” International Journal of Electronics, vol. 64, no. 2, pp. 229-237, Feb. 1988.

[118] C. L. Chan and H. K. Kwan, “Simple design of 2-dimensional circularly symmetric recursive digital filters using 1-dimensional analogue filters,” International Journal of Electronics, vol. 67, no. 4, pp. 585-590, Oct. 1989.

[119] H. K. Kwan and C. L. Chan, “Design of multidimensional spherically symmetric and constant group delay recursive digital filters with sum of powers-of-two coefficients,” IEEE Transactions on Circuits and Systems, vol. CAS-37, no. 8, pp. 1027-1035, Aug. 1990.

[120] C. L. Chan and H. K. Kwan, “Multidimensional ellipsoidally and spherically symmetric recursive digital filter design with reduced number of optimised coefficients,” IEE Proc. Part G on Circuits, Devices and Systems, vol. 136, no. 6, pp. 301-305, Dec. 1989.

[121] C. L. Chan and H. K. Kwan, “Narrow transition width two-dimensional digital fan filters with sum of powers-of-two coefficients,” International Journal of Electronics, vol. 68, no. 3, pp. 391-398, Mar. 1990.

[122] H. K. Kwan and M. T. Tsim, “Efficient systolic high speed architectures for delayed multipath two-dimensional FIR and IIR digital filtering,” IEE Proceedings Part G on Circuits, Devices and Systems, vol. 137, no. 6, pp. 413-423, Dec. 1990.

[123] J. H. Holland, Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence, Cambridge: MIT Press, 1992. ISBN:0262082136.

[124] D. E. Goldberg, Genetic Algorithms in Search Optimization and Machine Learning, Boston, Massachusetts: Addison-Wesley Longman Publishing Co, 1st edition, 1989. ISBN:0201157675.

[125] J. Kennedy and R. Eberhart, “Particle swarm optimization,” in Proc. of ICNN'95 - International Conference on Neural Network, Perth, Western Australia, Nov. 27-Dec.1, 1995, pp. 1942-1948.

[126] M. Dorigo and T. Stutzle, Ant Colony Optimization, Cambridge, MA: MIT Press, Jun. 2004. ISBN: 9780262042192.

[127] K. V. Price, R. M. Storn, and J. A. Lampinen, Differential Evolution A Practical Approach to Global Optimization, Verlag, Berlin Heidelberg: Springer, 2005. ISBN 978-3-540-31306-9.

[128] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," Science, vol. 220, no. 4598, pp. 671-680, May 1983.

[129] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Erciyes Univ. Press, Erciyes , 2005.

[130] D. Karaboga, "Artificial bee colony algorithm," Scholarpedia, vol. 5, Mar. 2010.

[131] D. Karaboga and B. Basturk, "A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm," Journal of Global Optimization, vol. 39, no. 3, pp. 459-471, Nov. 2007.

[132] D. Karaboga and B. Basturk, "On the performance of artificial bee colony (ABC) algorithm," Applied Soft Computing, vol. 8, no. 1, pp. 687–697, Jan. 2008.

[133] D. Karaboga and B. Akay, "A comparative study of artificial bee colony algorithm," Applied Mathematics and Computation, vol. 214, no. 1, pp. 108-132, Aug. 2009.

[134] B. K. Verma and D. Kumar, "A review of artificial bee colony algorithm," International Journal of Engineering and Technology, vol. 2, no. 3, pp. 175-186, Jun. 2013.

[135] D. Karaboga and B. Akay, "A modified artificial bee colony (ABC) algorithm for constrained optimization problems," Applied Soft Computing, vol. 11, no. 3, pp. 3021-3031, Apr. 2011.

[136] G. Zhu and S. Kwong, "Gbest-guided artificial bee colony algorithm for numerical function optimization," Applied Mathematics and Computation, vol. 217, no.7, pp. 3166–3173, Dec. 2010.

[137] K. Diwold, A. Aderhold, A. Scheidler, and M. Middendorf, "Performance evaluation of artificial bee colony optimization and new selection schemes," Memetic Computing, vol. 3, no.149, pp. 149–162, Oct. 2011.

[138] A. Banharnsakun, T. Achalakul, and B. Sirinaovakul, "The best-so-far selection in artificial bee colony," Applied Soft Computing, vol. 11, no.2, pp. 2888–2901, Feb. 2011.

[139] B. Alatas, “Chaotic bee colony algorithms for global numerical optimization,” Expert Systems with Applications, vol. 37, no.8, pp. 5682–5687, Aug. 2010.

[140] W. Gao and S. Liu, “Improved artificial bee colony algorithm for global optimization,” Information Processing Letters, vol. 111, no.17, pp. 871–882, Sep. 2011.

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

[142] F. Kang, J. Li, and Z. Ma, “Rosenbrock artificial bee colony algorithm for accurate global optimization of numerical functions,” Information Sciences, vol. 181, no. 16, pp. 3508-3531, Aug. 2011.

[143] D. Aydin, T. Liao, M.A. Montes de Oca, and T. Stützle, “Improving performance via population growth and local search: the case of the artificial bee colony algorithm,” in: Artificial Evolution, J.K. Hao, P. Legrand, P. Collet, N. Monmarché, E. Lutton, M. Schoenauer, Eds. (Lecture Notes in Computer Science, vol 7401) Springer, Berlin, Heidelberg, 2011, pp 85-96. ISBN 978-3-642-35532-5.

[144] W. Gao, S. Liu, and L. Huang, “A novel artificial bee colony algorithm based on modified search equation and orthogonal learning,” IEEE Transactions on Cybernetics, vol. 43, no. 3, pp. 1011-1024, Jun. 2013.

[145] L. S. Coelho and P. Alotto, “Gaussian artificial bee colony algorithm approach applied to Loney's solenoid benchmark problem,” IEEE Transactions on Magnetics, vol. 47, no. 5, pp. 1326-1329, May 2011.

[146] W. Gao, L. Huang, S. Liu, and C. Dai, “Artificial bee colony algorithm based on information learning,” IEEE Transactions on Cybernetics, vol. 45, no. 12, pp. 2827-2839, Jan. 2015.

[147] I. Fister, I. J. Fister, J. Brest, and V. Zumer, “Memetic artificial bee colony algorithm for largescale global optimization,” in Proc. of 2012 IEEE Congress on evolutionary computation, Brisbane, QLD, Jun. 10-15, 2012, pp. 1-8.

[148] X. Yan, Y. Zhu, and W. Zou, “A hybrid artificial bee colony algorithm for numerical function optimization,” in Proc. of 11th International Conference on Hybrid Intelligent Systems (HIS), Melacca, Malaysia, Dec. 5-8, 2011, pp. 127-132.

[149] H. Ming, J. Baohui, and L. Xu, “An improved bee evolutionary genetic algorithm,” in Proc. of IEEE international conference on intelligent computing and intelligent systems, Xiamen, China, Oct. 29-31, 2010, pp. 372-374.

[150] M. El-Abd, “A hybrid ABC-SPSO algorithm for continuous function optimization,” in Proc. of IEEE Symposium on swarm intelligence, Paris, France, Apr. 11-15, 2011, pp.1-6.

[151] T. Sharma, M. Pant, and T. Bhardwaj, “PSO ingrained artificial bee colony algorithm for solving continuous optimization problems,” in Proc. of International conference on computer applications and industrial electronics (ICCAIE), Penang, Malaysia, Dec. 4-7, 2011, pp. 108-112.

[152] X. Shi, Y. Li, H. Li, R. Guan, L. Wang, and Y. Liang, “An integrated algorithm based on artificial bee colony and particle swarm optimization,” in Proc. of Sixth International Conference on Natural Computation, Yantai, China, Aug. 10-12, 2010, pp. 2586-2590.

[153] M. S. Kiran and M. Gunduz, “A recombination-based hybridization of particle swarm optimization and artificial bee colony algorithm for continuous optimization problems,” Applied Soft Computing, vol. 13, no. 4, pp. 2188–2203, Apr. 2013.

[154] Y. Zhong, J. Lin, J. Ning, and X. X. Lin, “Hybrid artificial bee colony algorithm with chemotaxis behavior of bacterial foraging optimization algorithm,” in Proc. of Seventh International conference on Natural Computation, Shanghai, China Jul. 26-28, 2011, pp. 1171-1174.

[155] A. Abraham, R. K. Jatoth, and R. Anguluri, “Hybrid differential artificial bee colony algorithm,” Journal of Computational and Theoretical Nanoscience, vol. 9, no. 2, pp. 249-257, Feb. 2012.

[156] B. Wu and S. Fan, “Improved artificial bee colony algorithm with chaos,” in Computer Science for Environmental Engineering and EcoInformatics, Springer Berlin, Heidelberg, 2011, pp. 51–56, ISBN 978-3-642-22694-6.

[157] T. K. Sharma and M. Pant, “Enhancing scout bee movements in artificial bee colony algorithm,” in Advances in Intelligent and Soft Computing, India, Springer, Dec. 2011, pp. 601-610.

[158] A. Rajasekhar, A. Abraham, and M. Pant, “Levy mutated artificial bee colony algorithm for global optimization,” in Proc. of IEEE International Conference on Systems, Man, and Cybernetics, Anchorage, AK, USA Oct. 9-12, 2011, pp.655- 662.

[159] H. Quan and X. Shi, “On the analysis of performance of the improved artificial-bee-colony algorithm,” in Proc. of Fourth International conference on Natural Computation, Jinan, China, Oct. 18-20, 2008, pp. 654-668.

[160] P. Tsai, J. Pan, B. Liao, and S. Chu, “Enhanced artificial bee colony optimization,” International Journal of Innovative Computing, Information and Control, vol. 5, no. 12, pp. 5081-5092, Dec. 2009.

[161] F. Kang, J. J. Li, and Q. Xu, “Structural inverse analysis by hybrid simplex artificial bee colony algorithms,” Journal Computers and Structures, vol. 87, no. 13-14, pp. 861–870, Jul. 2009.

[162] H. Zhao, Z. Pei, J. Jiang, R. Guan, C. Wang, and X. Shi, “A hybrid swarm intelligent method based on genetic algorithm and artificial bee colony,” in Advances in Swarm Intelligence ICSI 2010 Tan Y., Shi Y., Tan K.C. Eds., (Lecture Notes in Computer Science, vol 6145). Springer, Berlin, Heidelberg, 2010, pp. 558-565.

[163] H. B. Duan, C. F. Xu, and Z. H. Xing, “A hybrid artificial bee colony optimization and quantum evolutionary algorithm for continuous optimization problems,” International Journal of Neural Systems, vol. 20, no. 1, pp. 39-50, Feb. 2010.

[164] F. Kang, J. J. Li, and H. J. Li, “Artificial bee colony algorithm and pattern search hybridized for global optimization,” Applied Soft Computing, vol. 13, no. 4, pp. 1781–1791, Apr. 2013.

[165] W. F. Gao, S. Y. Liu, and L. L. Huang, “A novel artificial bee colony algorithm with Powell’s method,” Applied Soft Computing, vol. 13, no. 9, pp. 3763–3775, Sep. 2013.

[166] K. Deb, Multi-Objective Optimization Using Evolutionary Algorithms, New York, NY, USA: John Wiley & Sons, Inc, 2001, ISBN:047187339X.

[167] J. Knowles and D. Corne, “The pareto archived evolution strategy: A new baseline algorithm for pareto multiobjective optimisation,” in Proc. of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, Jul. 6-9,1999, vol.1, pp.98-105.

[168] J. D. Knowles and D. W. Corne, “Approximating the nondominated front using the Pareto archived evolution strategy,” Evolutionary Computation, vol. 8, no. 2, pp. 149-172, Jun. 2000.

[169] E. Zitzler and L. Thiele, “Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach,” IEEE Transactions on Evolutionary Computation, vol. 3, no. 4, pp. 257-271, Nov. 1999.

[170] E. Zitzler, M. Laumanns, and L. Thiele, “SPEA2: Improving the strength pareto evolutionary algorithm,” Computer Engineering and Networks Laboratory (TIK), Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, May 2001.

[171] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: NSGA-II,” IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197, Apr. 2002.

[172] R. Hedayatzadeh, B. Hasanizadeh, R. Akbari, and K. Ziarati, “A multi-objective artificial bee colony for optimizing multi-objective problems,” in Proc. of 3rd International Conference on Advanced Computer Theory and Engineering, Chengdu, China, Aug. 20-22, 2010, V5-277-V5-281.

[173] H. Zhang, Y. Zhu, W. Zou, and X. Yan, “A hybrid multi-objective artificial bee colony algorithm for burdening optimization of copper strip production,” Applied Mathematical Modelling, vol. 36, no. 6, pp. 2578-2591, Jun. 2012.

[174] S. Omkar, J. Senthilnath, R. Khandelwal, G. N. Naik, and S. Gopalakrishnan, “Artificial bee colony (ABC) for multi-objective design optimization of composite structures,” Applied Soft Computing, vol. 11, no. 1, pp. 489-499, Jan. 2011.

[175] A. Kishor, P. K. Singh, and J. Prakash, “NSABC: Non-dominated sorting based multi-objective artificial bee colony algorithm and its application in data clustering,” Neurocomputing, vol. 216, no. 2016, pp. 514-533, Dec. 2016.

[176] K. Deb, J. Sundar, U. B. Rao, and S. Chaudhuri, “Reference point based multiobjective optimization using evolutionary algorithms,” International Journal of Computational Intelligence Research, vol. 2, no. 3, pp. 273-286, 2006.

[177] C. Fonseca and P. Fleming, “Genetic algorithms for multiobjective optimization: formulation, discussion and generalization,” in Proc. of the 5th International Conference on Genetic Algorithms, San Francisco, CA, USA, Morgan Kaufmann Publishers Inc, 1993, pp. 416 - 423.

[178] M. Sakawa and K. Kato, “An interactive fuzzy satisficing method for general multiobjective 0-1 programming problems through genetic algorithms with double strings based on a reference solution,” Fuzzy Sets and Systems, vol. 125, no. 3, pp. 289-300, Feb. 2002.

[179] K. Deb, “Nonlinear goal programming using multi-objective genetic algorithm,” The Journal of Operational Research Society, vol. 52, no. 3, pp. 291- 302, Mar. 2001.

[180] J. Branke and K. Deb, “Integrating user preferences into evolutionary multi-objective optimization,” in: Knowledge Incorporation in Evolutionary Computation: Studies in Fuzziness and Soft Computing, vol 167, Y. Jin Eds., Springer, Berlin, Heidelberg, Springer, 2005, pp. 461-477, ISBN 978-3-540-44511-1.

[181] K. Deb and S. Chaudhuri, “I-EMO: An interactive evolutionary multi-objective optimization tool,” in Pattern Recognition and Machine Intelligence (Lecture Notes in Computer Science, vol 3776), S.K. Pal, S. Bandyopadhyay, S. Biswas Eds., Springer, Berlin, Heidelberg, 2005, pp. 690-695.

[182] H. Li and D. Landa-Silva, “Evolutionary multi-objective simulated annealing with adaptive and competitive search direction,” in IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, China, Jun. 1-6, 2008, pp. 3311-3318.

[183] J. Sanchis, M. Martínez, and X. Ferragud, “Integrated multiobjective optimization and a priori preferences using genetic algorithms,” Information Sciences, vol. 178, no. 4, pp. 931-951, Feb. 2008.

[184] K. Deb, A. Sinha, P. J. Korhonen, and J. Wallenius, “An interactive evolutionary multiobjective optimization method based on progressively approximated value functions,” IEEE Transactions on Evolutionary Computation, vol. 14, no. 5, pp. 723-738, Oct. 2010.

[185] B. Akay and D. Karaboga, “A modified artificial bee colony algorithm for real parameter optimization,” Information Sciences, vol. 192, pp. 120-142, 2010.

[186] A. P. Wierzbicki, “The use of reference objectives in multiobjective optimization,” in Multiple Criteria Decision Making Theory and Applications, Berlin, Germany, Springer, 1980, pp. 468-486.

[187] K. K. Dhaliwal and J. S. Dhillon, “Integrated cat swarm optimization and differential evolution algorithm for optimal IIR filter design in multi-objective framework,” Circuits Systems and Signal Processing, vol. 36, no. 1, pp. 270-296, Jan. 2017.

[188] G. Reynoso-Meza, J. Sanchis, X. Blasco, and S. G. Nieto, “Physical programming for preference driven evolutionary multiobjective optimization,” Applied Soft Computing, vol. 24, pp. 341-362, Nov. 2014.

[189] Y. Xu, F. Yao, S. Chai, and L. Sun, “Multiobjective optimization of extreme learning machine using physical programming,” in Proc. of 35th Chinese Control Conference, Chengdu, China, Jul. 27-29, 2016, pp. 3618-3623.

[190] M. A. Ilgin and S. M. Gupta, “Physical programming: A review of the state of the art,” Studies in Informatics and Control, vol. 21, no. 4, pp. 349-366, 2012.

[191] J. Branke, K. Deb, H. Dierolf, and M. Osswald, “Finding knees in multi-objective optimization," in: Parallel Problem Solving from Nature, Yao X. et al. Eds., (Lecture Notes in Computer Science, vol 3242), Springer, Berlin, Heidelberg, 2004, pp. 722-731.

[192] S. Bechikh, L. B. Said, and K. Ghédira, “Searching for knee regions of the Pareto front using mobile reference points,” in Soft Computing, vol 15, no.9, Springer-Verlag, Sept. 2011, pp. 1807–1823.

[193] H. Wang, S. He, and X. Yao, “Nadir Point estimation for many-objective optimization problems based on emphasized critical regions,” in Soft Computing, vol. 21, no. 9, Berlin Heidelberg, Springer, May 2017, pp. 2283–2295.

[194] S. Bechikh, L. B. Said, and K. Ghedira, “Estimating nadir point in multi-objective optimization using mobile reference points,” in Proc. of IEEE Congress on Evolutionary Computation, Barcelona, Spain, 18-23 Jul. 2010, pp. 1-9.

[195] A. G. Constantinides, “Spectral transformations for digital filters,” Proc. of the Institution of Electrical Engineers, vol. 117, no. 8, pp. 1585-1590.

[196] A. G. Constantinides, “Design of digital filters from LC ladder networks,” Proc. of the Institution of Electrical Engineers, vol. 123, no. 12, Dec. 1976, pp. 1307-1312.

[197] H. K. Kwan and Y. C. Lui, “Lattice implementation of two-dimensional recursive digital filters,” IEEE Transactions on Circuits and Systems, vol. 36, no. 3, pp. 383-386, Mar. 1989.

[198] A. Fettweis, “Wave digital filters: Theory and practice,” Proc. of IEEE, vol. 74, no. 2, Feb. 1986, pp. 270-327.

[199] H. K. Kwan, “Digital filter realization by generalized wave digital adaptors,” in Proc. of IEEE International Symposium on Circuits and Systems, Newport Beach, California, USA, May 2-4, 1983, vol. 1, pp. 432-436.

## APPENDIX A: IEEE PERMISSION TO REPRINT

# VITA AUCTORIS

Rija Raju was born in India, in 1988. She received her Bachelor degree in Electronics and Communication engineering from Cochin University of Science and Technology, Kerala, India in 2011, and her Master degree in Communication Systems from University of Kerala, Kerala, India in 2012. Now she is pursuing her PhD degree in Electrical Engineering with the University of Windsor, Windsor, Ontario, Canada.

During her master's program, she was qualified for the GATE scholarship from Ministry of HRD, Government of India and received tuition-waiver for the Master program. She was the university topper and gold medalist for her master's degree. She received her honorary membership in Golden Key International Society in August 2016 for her consistent hard work and academic excellence.

From 2015 to 2019, Rija Raju received Doctoral entrance Scholarship of University of Windsor. Furthermore, she was awarded with Canterbury College entrance Scholarship (2016), Clifford T. Terry residence Scholarship (2016), and Frederick Atkins graduate Scholarship (2017, 2018).

Her research interests include evolutionary algorithms, and its applications to digital signal processing and communication systems.