

# FIR FILTER DESIGN USING MULTIOBJECTIVE PARTICLE SWARM OPTIMIZATION

Major Project submitted in partial fulfilment of the requirements for the award of degree of

Master of Technology

in

Information Systems

Submitted By:

Pooja Upadhyay

(2K11/ISY/16)

Under the Guidance of:

Prof. O. P. Verma

(HOD, IT Department)



Department of Information Technology  
Delhi Technological University  
Bawana Road, Delhi – 110042 (2011-2013)

## CERTIFICATE

This is to certify that Ms. Pooja Upadhyay (2K11/ISY/16) has carried out the major project titled “FIR filter design using multiobjective particle swarm optimization” as a partial requirement for the award of Master of Technology degree in Information Systems by Delhi Technological University.

The major project is a bona fide piece of work carried out and completed under my supervision and guidance during the academic session 2011-2013. The matter contained in this report has not been submitted elsewhere for the award of any other degree.

(Project Guide)

Prof. O. P. Verma

Head of Department

Department of Information Technology

Delhi Technological University

Bawana Road, Delhi-110042

## ACKNOWLEDGEMENT

I express my gratitude to my major project guide Prof. O. P. Verma, HOD, IT Dept., Delhi Technological University, for the valuable support and guidance he provided in making this major project. It is my pleasure to record my sincere thanks to my respected guide for his constructive criticism and insight without which the project would not have shaped as it has.

I humbly extend my words of gratitude to other faculty members of this department for providing their valuable help and time whenever it was required.

Pooja Upadhyay

Roll No. 2K11/ISY/16

M.Tech (Information Systems)

E-mail: pu90\_upadhyay@yahoo.co.in

The proposed method designs a FIR filter with arbitrary frequency response and reduced delay. The coefficients of this FIR filter are real and the delay is less than half of the filter coefficients. The FIR filter in this approach has non linear group delay. The filter is designed using multi-objective approach minimizing certain objective functions that are responsible for generating a filter of lower delay and magnitude response of unity in the passband. The multiobjective constraints are tailored by incorporating an evolutionary algorithm with multiobjective approach. In this proposed approach Particle Swarm Optimization with multiobjective optimization(MOPSO) produces a set of non-dominated solutions called pareto optimals. The MOPSO takes a set of real coefficients of the FIR as the population and using multiobjective error formulation of amplitude response and group delay gives optimal FIR filters. The error formulation for magnitude response is to have a response of 1 in passband and 0 in stopband, and for the group delay, it must be as close to linearity as possible in passband. These solutions can tailor all types of requirements of the decision maker. The proposed approach has been compared with weighed least square method and the experimental results have shown that the magnitude response and delay characteristic using proposed approach are better than those achieved by weighed least square approach.

## List of figures

1: A low pass filter with all the specifications .....	4
2: An Ideal filters Lowpass, Highpass, Bandstop and Bandpass .....	5
3: Tradeoff between two objective functions in multiobjective optimization.....	14
4: Mapping from decision variable space to objective function space .....	15
5: Concept of modification of a searching point by PSO .....	17
6: Flowchart of multi-objective particle swarm optimization .....	19
7(a): The behavior of archive controller, when new solution is added.....	23
7(b): Adjustment of new solution that belongs within the grid boundaries.....	24
7(c): New solution lies outside the grid boundaries .....	24
8: Outputs for lowpass of objective functions, magnitude response and group delay .....	30
9: Outputs for highpass of objective functions, magnitude response and group delay .....	31
10: Outputs for bandpass of objective functions, magnitude response and group delay .....	32
11: Output for bandpass filter of magnitude response by WLS and proposed approach .....	34
12: Output of group delay for bandpass filter using WLS method and proposed method.....	35
13: Output for lowpass filter of magnitude response by WLS and proposed approach.....	36
14: Output of group delay of a lowpass filter using WLS method and proposed method .....	37
15: Output for highpass filter of magnitude response for WLS and proposed approach.....	38

16: Output of group delay of a highpass filter using WLS method and proposed method .... 39

### List of tables

1: frequency response of filters with coefficient symmetry ..... 4

2: Values for lowpass filter of Length 28 ..... 40

3: Values for bandpass filter of Length 28 ..... 40

4: Values for highpass filter of Length 28 ..... 40

## Table of Contents

<b>Certificate</b> .....	<b>i</b>
<b>Acknowledgement</b> .....	<b>ii</b>
<b>Abstract</b> .....	<b>iii</b>
<b>List of figures</b> .....	<b>iv</b>
<b>Chapter 1: Digital Filters</b> .....	<b>1</b>
1.1 Introduction .....	1
1.2 FIR filter .....	2
1.2.1 Filter specification .....	4
1.2.2 FIR Filter design .....	6
1.3 Previous approaches of FIR design .....	9
<b>Chapter 2: Multiobjective Optimization</b> .....	<b>11</b>
2.1 Introduction .....	11
2.2 Multiobjective optimization .....	11
2.2.1 Pareto terminology.....	12
2.2.2 Multiobjective problem solving approaches.....	15
2.3 MOPSO .....	16
2.3.1 Main Algorithm .....	19
2.3.2 External repository .....	22
<b>Chapter 3: Proposed Approach</b> .....	<b>26</b>
3.1 Objective function.....	27
3.2 Parameter values .....	28
3.3 Proposed method description .....	28
<b>Chapter 4: Results</b> .....	<b>29</b>
4.1 Results of the proposed method .....	29
4.2 Comparison of proposed method with WLS approach .....	33
<b>Chapter 5: Conclusion</b> .....	<b>41</b>
<b>References</b> .....	<b>42</b>

## 1.1 Introduction

Filters are the most essential elements in signal processing and telecommunication systems. These have various applications in control systems, systems for audio and video processing, audio graphic equalizer, and communication systems, noise reduction filters in Dolby systems, systems for medical applications, image deblurring, and image edge emphasis. Digital filters can be implemented in both software as well as hardware. Some user-friendly programming languages such as Matlab, helps to easily design all types of filters. A digital filter is a system, which modifies a sampled discrete time signal by enhancing or reducing certain aspects of the signal, using various mathematical operations.

Today, digital filters have almost replaced analog filters in many applications. Although digital filters are far more expensive than analog filters, but the digital filters have made some of the design possible, which were impractical by analog filters. Furthermore, digital filters can process real time signals, are more adaptive and stable, can design finite impulse response filters, have better signal to noise ratio as compared to analog filters.

Some of the functions of filters are listed below:

- a) To limit the signal within a particular frequency range.
- b) To decompose a signal into various sub-bands.
- c) To alter the frequency spectrum of a signal.
- d) To model the input-output relationship of a system.



A filter can be modeled to limit the signal, based on the application. The various forms are low-pass, high-pass, band-pass and band-stop. Changing the frequency range changes the form of the filter. A low-pass filter allows the low frequency signal to pass through it. Similarly, high pass allows a high frequency signal, and band-pass and band-stop allows certain frequency range signals.

Digital filter are mainly grouped into two types, FIR filters and IIR filters. This dissertation focuses on FIR filters. The next section describes the FIR filters in details.

## **1.2 FIR Filters**

FIR filters are non- recursive digital filters with finite impulse response. The non- recursive nature of these filters is due to lack of feedback circuitry. The filter is called finite impulse response, as the impulse response is finite, i.e., it reaches zero in finite time.

Each filter has predefined order  $N$  and for a filter with order  $N$ , the impulse response lasts for  $N+1$ , samples and then reduces to zero.

FIR filters have following attributes:

- Linear phase due to symmetry of coefficients
- high filter order (more complex circuits)
- stability as poles exist only at origin
- No feedback required

Although with FIR filters the drawback is it has higher filter order, thus large number of coefficients required and increasing the complexity and cost. As a result these filters are used only when linear phase is desired. The order of the filter is defined by the number of delay lines, and corresponding number of input samples must be saved for computing the output. FIR filters are characterized by given equation:

$$y(n) = \sum_{k=0}^{N-1} (h(k)x(n-k)) \quad (1)$$

Where,  $x(n)$  is the input signal,  $h(n)$  the impulse response and  $y(n)$  is the output signal obtained by summation of the convolution of  $x(n)$  and  $h(n)$  over filter order  $N$ .

The same can also be expressed in z-transform,

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n} \quad (2)$$

FIR filters have linear phase due to symmetry of coefficients. Based on different types of symmetry and number of filter coefficients FIR filters can be grouped as:

- **Symmetric with even order:** has impulse response  $h[n] = h[N-n-1]$  and the number of coefficients in this filter are even.
- **Symmetric with odd order:** has impulse response  $h[n] = h[N-n-1]$  and the number of coefficients in this filter are odd.
- **Asymmetric with even order:** has impulse response  $h[n] = -h[N-n-1]$  and the number of coefficients in this filter are even.
- **Asymmetric with odd order:** has impulse response  $h[n] = -h[N-n-1]$  and the number of coefficients in this filter are odd.

Impulse response	# coefs	$H(\omega)$	Type
$h(n) = h(M-1-n)$	Odd	$e^{-j\omega(M-1)/2} \left( h\left(\frac{M-1}{2}\right) + 2 \sum_{k=1}^{(M-3)/2} h\left(\frac{M-1}{2} - k\right) \cos(\omega k) \right)$	1
$h(n) = h(M-1-n)$	Even	$e^{-j\omega(M-1)/2} 2 \sum_{k=1}^{(M-3)/2} h\left(\frac{M}{2} - k\right) \cos\left(\omega\left(k - \frac{1}{2}\right)\right)$	2
$h(n) = -h(M-1-n)$	Odd	$e^{-j[\omega(M-1)/2 - \pi/2]} \left( 2 \sum_{k=1}^{(M-1)/2} h\left(\frac{M-1}{2} - k\right) \sin(\omega k) \right)$	3
$h(n) = -h(M-1-n)$	Even	$e^{-j[\omega(M-1)/2 - \pi/2]} 2 \sum_{k=1}^{(M-1)/2} h\left(\frac{M}{2} - k\right) \sin\left(\omega\left(k - \frac{1}{2}\right)\right)$	4

Table 1: frequency response of filters with coefficient symmetry

The table 1 above shows the frequency response for all types of FIR filter. Some of the observations derived for above four types of filters are:

- Type 1 are most versatile of all others.
- Type 2 has frequency response which is always 0 at  $\omega=\pi$  and is not suitable as a high-pass.
- Type 3 and 4 introduces a  $\pi/2$  phase shift and the frequency response is always 0 at  $\omega=0$  and not suitable as a high-pass.

### 1.2.1 Filter Specifications

Almost all FIR filters are based on ideal approximation of the filter. Increasing the order of the filter tends to more ideal approximation. FIR filter in figure 1 below is a low-pass filter.

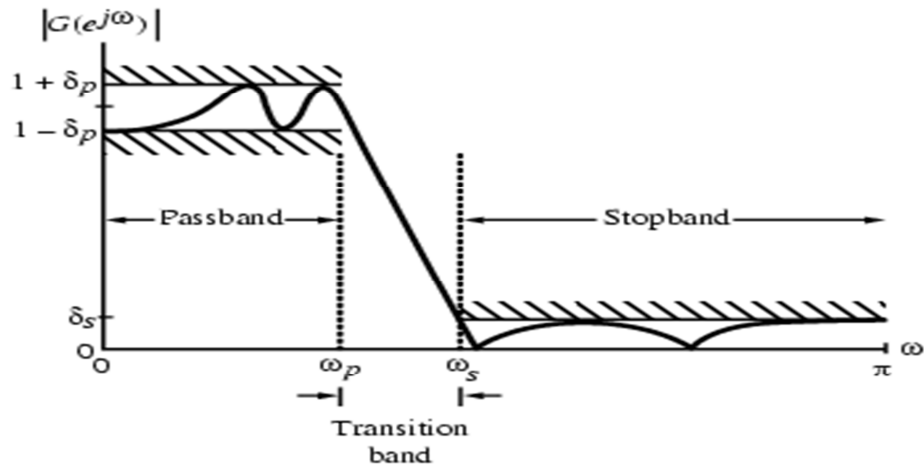


Fig 1: A low pass filter with all the specifications

$\omega_p$  – normalized cut-off frequency in the passband       $\delta_s$  – maximum ripple in the stopband

$\omega_s$  – normalized cut-off frequency in the stopband       $f_s$  –sampling frequency

$\omega$  – normalized frequency       $\delta_p$  – maximum ripples in the passband

In the pass-band  $0 \leq \omega \leq \omega_p$  we require that  $|G(e^{j\omega})| \cong 1$  with a deviation  $\pm \delta_p$

$$1 - \delta_p \leq |G(e^{j\omega})| \leq 1 + \delta_p, \quad |\omega| \leq \omega_p \quad (3)$$

In the stop-band  $\omega_s \leq \omega \leq \pi$  we require that  $|G(e^{j\omega})| \cong 0$  with a deviation  $\delta_s$

$$|G(e^{j\omega})| \leq \delta_s, \quad \omega_s \leq |\omega| \leq \pi \quad (4)$$

Each filter is characterized by magnitude response, phase response, phase delay and group delay.

The magnitude response denoted as  $|H(e^{j\omega})|$  is modulus of frequency filter response or the transfer function of the filter  $H(z)$ . The figure 2 below shows the ideal low-pass, band-pass, band-stop, high-pass filters.

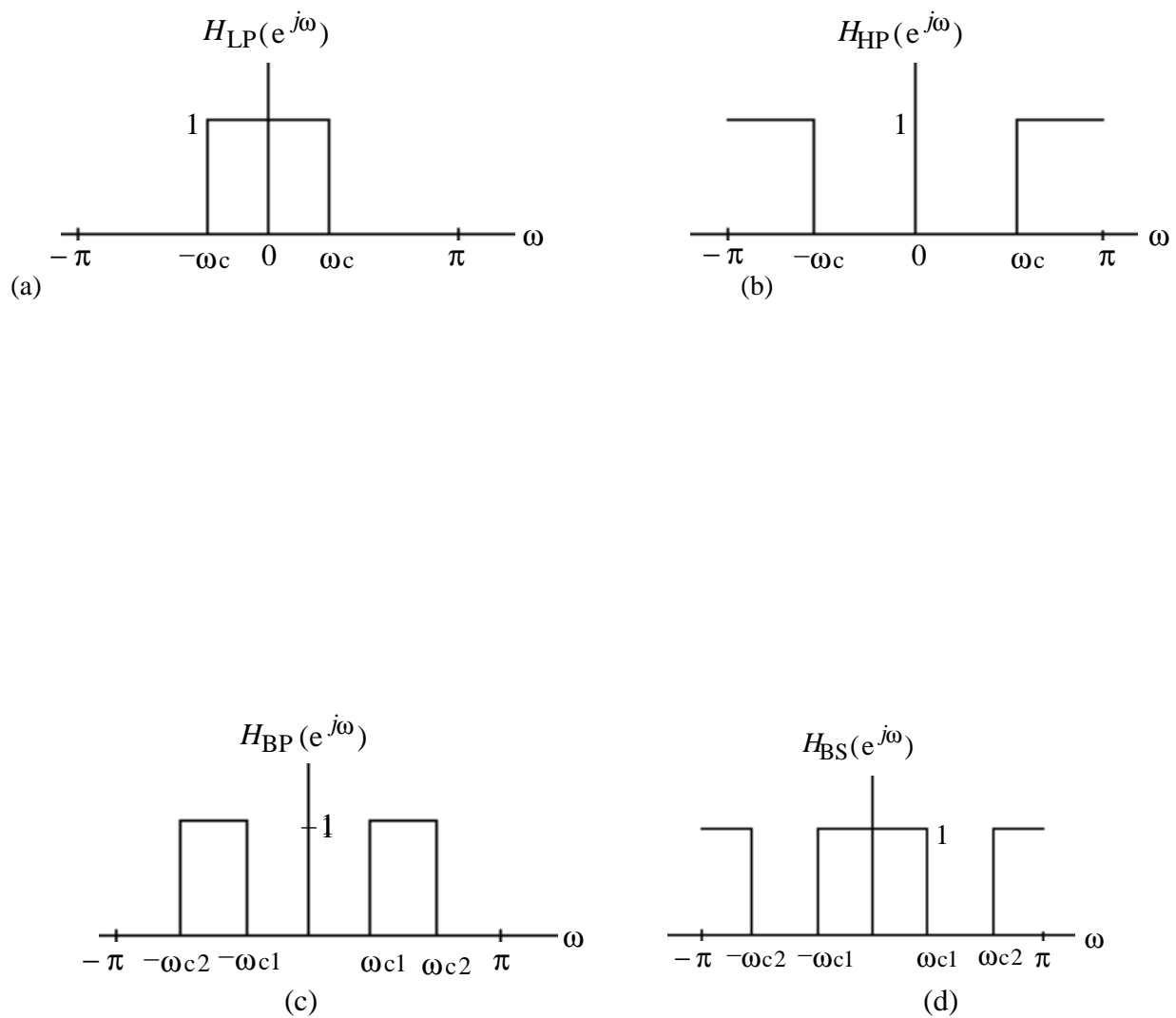


Figure 2: Ideal filters; a) Lowpass b) Highpass c) Bandstop d) Bandpass

### 1.2.2 FIR Filter Design

Most filters are designed to approximate the ideal response. An ideal response characteristic is impossible to achieve, although various methods have been devised to attain almost ideal response.

These are the four most common approaches used for filter design:

1. Least square error minimization
2. Windowing technique
3. Frequency sampling method
4. Computer based optimization approach

In this dissertation we have used least square error minimization method and computer based optimization method.

- **Least square error minimization:** The least square error minimization works by approximating the magnitude response to the desired magnitude response of the filter. This method requires specifying the desired magnitude response, given as:

$$H_d(e^{j\omega}) \stackrel{DTFT}{=} \sum_n h_{d,n} e^{-j\omega n} \quad (5)$$

And the filter coefficients can be obtained using equation 6, given below:

$$h_{d,n} = \frac{1}{2\pi} \int_{-\pi}^{\pi} H_d(e^{j\omega}) e^{j\omega n} d\omega \quad (6)$$

Although, the filter obtained from above equation is non-casual and of infinite length. So to make the filter finite, truncation is required at some point. As a criterion for truncation, the difference between desired response and truncated response is minimized. Thus, aiming to find

such an impulse response which is of finite duration and the DTFT of such impulse response approximates the desired frequency response.

The magnitude of the frequency response of the truncated filter is given as:

$$H_t(e^{j\omega}) = \sum_{n=L}^U h_{t,n} e^{-j\omega n} \quad (7)$$

Where, L and U are the lower and upper point or the boundaries at which the pulse response is truncated to get a pulse with finite response.

- **Frequency sampling method:** The frequency sampling method samples the frequency response producing N samples. The sampling is done at equally spaced frequencies. Frequency response  $H(\omega)$  is continuous in nature, thus sampling of frequency response gives the DFT of  $H(\omega)$ . Performing the IDFT gives the filter coefficients:

$$h(n) = \frac{1}{N} \sum_{k=0}^{N-1} \left( H(k) e^{j \frac{2\pi n k}{N}} \right) \quad (8)$$

This method can be used for any magnitude response value, unlike the window method. Also, the frequency response only at sampled points is equal to desired response.

- **Window method:**

Window method uses a finite weighing sequence  $w(n)$  called window. In this method, the infinite impulse response is multiplied by window  $w(n)$ , giving the Fourier coefficients. This method converts an infinite response to finite response, with the help of these window functions. This method is very simple and easy to implement, although has a drawback like frequency sampling method as it cannot be used for any magnitude response value.

The most popular and widely used window functions are; Rectangular window, Hanning window, hamming window and Kaiser window.

Some of the common windows are given below in equations 9-12:

i. Hanning window function

$$w(n) = \begin{cases} 0.5 - 0.5 * \cos\left(\frac{2\pi n}{N-1}\right) & 0 \leq n \leq N-1 \\ 0 & \textit{otherwise} \end{cases} \quad (9)$$

ii. Hamming window function

$$w(n) = \begin{cases} 0.54 - 0.46 * \cos\left(\frac{2\pi n}{N-1}\right) & 0 \leq n \leq N-1 \\ 0 & \textit{otherwise} \end{cases} \quad (10)$$

iii. A Generalized Hamming window function

$$w(n) = \begin{cases} \alpha - (1 - \alpha) * \cos\left(\frac{2\pi n}{N-1}\right) & 0 \leq n \leq N-1 \\ 0 & \textit{otherwise} \end{cases} \quad (11)$$

iv. Rectangular window

$$w(n) = \begin{cases} 1 & 0 \leq n \leq N-1 \\ 0 & \textit{otherwise} \end{cases} \quad (12)$$

- **Computer based optimization approach:** This is one the most recent methods of FIR filter design. Increasing use of optimization algorithm has made this method more popular over other methods. Also, it produces more efficient results by reducing the error as in frequency sampling method. In this method, a set of coefficient or coefficient vectors used to characterize an FIR filter is obtained by iteratively reducing the error (such as magnitude response). Although, this method is very complex but is highly efficient. Whereas, the FIR filter using

window method can be easily designed but are not very efficient. It is based on the designer to choose accuracy of approximation or ease of design. Under computer based optimization, one of the most common filter designed is arbitrary response filters.

### 1.3 Literature Survey

Some of the design methods most commonly used for filter design have been reported in literature. Some of the earliest methods were using chebyshev approximation to design analog filters. Various such attempts were made during 1960's to design such filter using this approximation. Very few were successful in designing such filters. One of them was the design of equiripple filter with restricted band edges by Otto Herrmann. Later, Ed Hofstetter using the design developed a FIR filter with as many ripples. Then in 1972 **Parks and McClellan** together proposed a new approach of designing FIR filter using chebyshev approximation iteratively [1]. It designs an optimal FIR filter which was efficient compared to other FIR filter designs. The algorithm gives optimal design by reducing the errors in both stopband as well as passband. Later various other techniques were proposed for the design of FIR. Some of the optimization methods were incorporated along with Parks-Mc Clellan algorithm. Some of them are Hui Zhao and Juebang Yu design neural network-based digital filter. V. Ralph Algazi design Finite duration filters using least-square method. Some of the design methods were proposed using single objective evolutionary algorithms [2-6]. Later, other design methods using multi objective evolutionary algorithm were given [7]. Also, a least square approach for non-recursive filters, with arbitrary magnitude and phase was proposed by S. K. Kidambi and R. P. Ramachandran [8]. M. Lang gave Algorithms for the Constrained Design of Digital Filters with Arbitrary Magnitude and Phase Responses [9].



## Chapter 2: Multiobjective Optimization

---

### 2.1 Introduction

In current real world scenario, almost all optimization problems are multi-objective in nature. Satisfying more than one objective at the same time, the multi-objective optimization technique produces a set of optimal solutions (pareto-optimal solutions). The decision maker has to select from the set depending on the application. Although, it is incorrect to say that a particular pareto-optimal solutions is better than other set. Thus, for a particular problem the user must run the optimization problem many times and get as many pareto-optimal solutions as possible. The ability to generate pareto-optimal solutions is the reason to extend multi-objective optimization to various evolutionary algorithms. Multi-objective optimization has been applied in various fields such as science, logistics, economics and finance.

#### **Single Objective Optimization:**

An optimization problem which consists of single objective function and the problem of finding an optimal solution is called single-objective optimization.

### 2.2 Multiobjective Optimization

The Multiobjective Optimization Problem (also called multi-criteria optimization, multi-performance or vector optimization problem) can then be defined (in words) as the problem of finding [24]:

“A vector of decision variables which satisfies constraints and optimizes a vector function whose elements represent the objective functions. These functions form a mathematical description of performance criteria which are usually in conflict with each other. Hence, the term, “optimize”

means finding such a solution which would give the values of all the objective functions acceptable to the decision maker.”[24]

It involves optimizing k-objective functions simultaneously and generating a set of solutions (pareto-optimal solutions) instead of a unique solution. The decision maker selects the solution from the solution set. The selection involves compromising one complete solution over another solution. The optimization problem may involve either minimization or maximization of k objective function or the combination of maximization or minimization of k such functions.

Mathematically, it is defined as:

$$\begin{array}{ll}
 \text{Minimizing (or Maximizing)} & F(x) = (f_1(x), \dots, f_k(x)) \\
 \text{subject to} & g_i(x) \leq 0, i = \{1, \dots, m\} \\
 \text{and} & h_j(x) = 0, j = \{1, \dots, p\}
 \end{array} \quad (13)$$

Here,  $F(x)$  is a vector of k-objective function,  $x$  which may be continuous or discrete is n-dimensional decision variable vector  $x = \{x_1 \dots x_n\}$ . Vector  $F(x)$  components are minimized or maximized based on the constraints  $g_i(x) \leq 0$  and  $h_j(x) = 0$ . The k-objective function may be linear or non-linear.

### 2.2.1 Pareto Terminology

With multi-objective optimization the notion of finding an optimum solution changes, as it becomes finding tradeoff rather than single solution.

- **Non-dominated Set:** Of the solution set  $P$ , the non dominated set is a solution set  $P'$  whose any member is not dominated by other member of set  $P$ .
- **Global Pareto Optimality Set:** The solution set of non-dominated solution within the feasible search space  $S$  is the globally Pareto-optimal set.

- **Pareto-optimality:** A solution  $x \in \Omega$  is said to be Pareto optimal with respect to universe  $\Omega$  if and only if there is no  $x' \in \Omega$  for which  $v = F(x') = (f_1(x'), \dots, f_k(x'))$  dominates  $u = F(x) = (f_1(x), \dots, f_k(x))$ . The above definition explains that, if  $x^*$  is pareto-optimal then there is no feasible vector  $x$  which may increase in some criteria without causing a simultaneous decrease in at least one another criterion (assuming maximization). The pareto optimal is with respect to decision variable space.

- **Pareto-dominance:** A vector  $u = (u_1, \dots, u_k)$  is said to dominate another vector  $v = (v_1, \dots, v_k)$  (denoted by  $u \preceq v$ ) if and only if  $u$  is partially less than  $v$ , i.e.,

$$\begin{aligned} \forall i \in \{1, \dots, k\} : u_i &\leq v_i \\ \wedge \exists i \in \{1, \dots, k\} : u_i &< v_i \end{aligned} \quad (14)$$

- **Pareto Optimal set:** For a given MOP,  $F(x)$ , the Pareto Optimal Set,  $P^*$ , is defined as:

$$P^* := \{x \in \Omega \mid \neg \exists x' \in F(x') \leq F(x)\} \quad (15)$$

Pareto optimal solutions are solutions belonging to set of solution. The set is represented as  $P^*$ . Within this set all solutions have the best possible objective values and cannot be simultaneously improved further for all objective values. These solutions are efficient solutions and are non-inferior. Therefore these solutions are termed as non-dominated solutions. Although, these non-dominated solutions belong to same set, but have no relationship to each other.

These pareto optimal solutions are plotted against various objective function along different dimensions. A curve as shown in Fig 3 is obtained. Each point on the curve represents a solution obtained in the set. These solutions can work for different applications, based on the requirement. In Fig 3, for a two objective function, a decision maker can also use solutions for application requiring better objective value for any one objective function. The leftmost points on the curve,

serve for low cost and high efficiency application. Rightmost solutions can be used for high cost and low efficiency applications. The centre of the curve contains solution having moderate cost and efficiency.

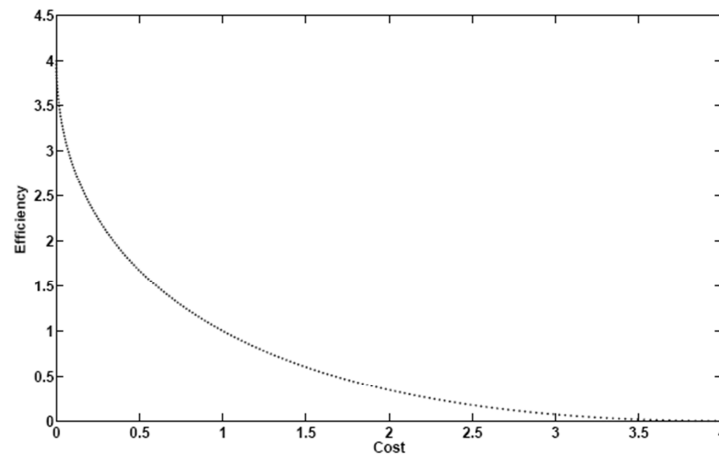


Fig 3: tradeoff between two objective functions

- **Pareto Front:** For a given MOP,  $F(x)$ , and Pareto Optimal set,  $P^*$ , the Pareto Front  $PF^*$  defined as:

$$PF^* = \{u = F(x) \mid x \in P^*\} \quad (16)$$

The pareto front  $PF^*$  contains the vector for each solution in pareto optimal set  $P^*$ . The vector components are evaluated objective values for non-dominated solutions. To generate a pareto-front, the points in the universe are evaluated for the objective functions. Then a set of non-dominated solutions are determined, producing a pareto front. The mapping from decision space to objective function space is shown in Fig 4. These points on the curve in objective function space give the pareto-front.

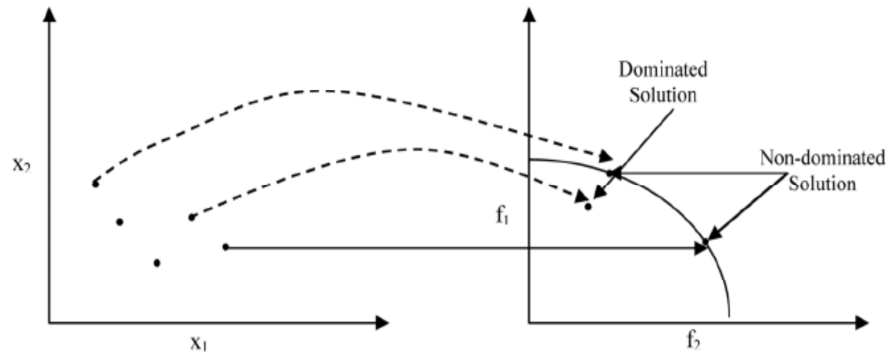


Fig 4: Mapping from decision variable space to objective function space

### 2.2.2 Multiobjective Problem Solving Approaches

These approaches define various ways in which multiobjective problems can be solved. These are the basic 3 techniques used:

- 1) **Weighted Sum Approach:** in this approach weights are assigned to each objective function. The sum of weights is always equal to one. It performs the sum of the product of the weights with corresponding objective functions
- 2) **Lexicography Approach:** In this approach it prioritizes the objective function. Each objective function is assigned a priority and based on based the priorities objectives are optimized.
- 3) **Pareto Approach:** Only one Pareto optimal solution can be expected to be found in one simulation run. All algorithms require some problem knowledge, such as suitable weights, epsilon, or target values, etc. Some of the pareto approach in the domain of evolutionary algorithms
  - VEGA (Vector Evaluated Genetic Algorithms) contributed by David Schaffer in 1984.
  - VOES (Vector Optimized Evolution Strategy) contributed by Frank Kursawe in 1990.
  - MOGA (Multi-objective GA) introduced by Fonseca and Fleming in 1993.
  - NSGA (Non-dominated Sorting GA) introduced by Srinivas and Deb in 1994.

### **2.3 Multiobjective Particle Swarm Optimization**

The first multiobjective evolutionary algorithm was implemented by David Schauffer in mid 1980's [25]. After his work, lot of effort has been made in this field, now referred to as Multiobjective evolutionary algorithm (MOEA). In past, a number of multiobjective algorithm have been proposed [10-13]. Various evolutionary algorithms such as genetic algorithm[23,24] has been enhanced to solve multiobjective problems. Multi-objective particle swarm optimization (MOPSO) [14] is multi-objective problem with enhanced problem solving capability of evolutionary algorithm (PSO). It incorporates pareto-dominance to particle swarm optimization giving pareto-optimal solution storing the non-dominated solution, rather than unique solution as in PSO.

Particle swarm optimization (PSO) is a population based stochastic optimization technique developed by Dr. Eberhart (electrical engineer) and Dr. Kennedy (social-psychologist) [15] in 1995, inspired by social behaviour of bird flocking or fish schooling. The algorithm starts by initializing the population with a set of random solutions. These particles or potential solutions move in search space on the basis of current optimal solutions. After performing fixed number of iteration (number of iterations equal to number of generations) it reaches to a unique optimal solution at the end. This requires no evolution operators like crossover or mutation. The updation is done based on the inertia of the particle moving in search space. In this each particle is treated as a point in an N-dimensional space. These particles adjust its "flying" according to its own flying experience as well as the flying experience of other particles.

Each particle has memory, storing its best position so far achieved by the particle, this position is called the pbest (personal best). These stored positions of the particle are obtained from the coordinates in the solution space. The PSO keeps track of another position, the position achieved

so far by any particle in solution space, this position is referred to as the gbest (global best). The PSO objective is to accelerate each particle towards these best positions. Fig 5 depicts the complete concept of how PSO works.

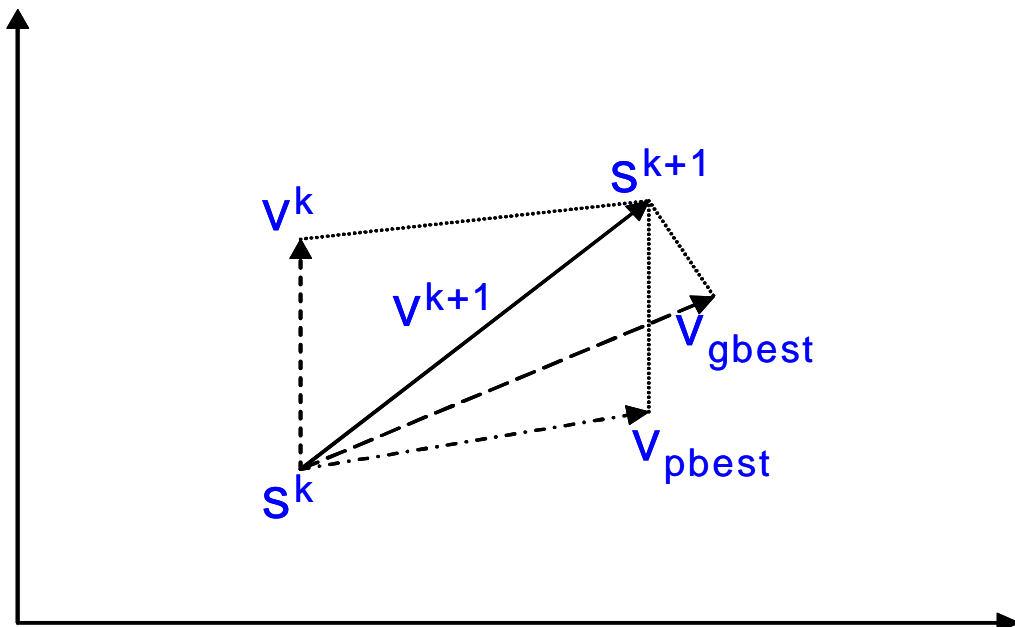


Fig.5: Concept of modification of a searching point by PSO

$S^k$ : current searching point

$S^{k+1}$ : modified searching point

$V^k$ : current velocity

$V^{k+1}$ : modified velocity

$V_{pbest}$ : velocity based on pbest

$V_{gbest}$ : velocity based on gbest

In PSO potential solutions fly through the hyperspace, thus increasing the convergence towards global best solution. This high speed convergence of PSO for single objective optimization, thus makes it best suitable for multiobjective optimization [16]. Also in PSO the individuals are benefited from their past experience and have very few parameters that need to be adjusted. The objective function in PSO can be both linear and non-linear. Also, PSO can handle both discrete and continuous type functions.

**Multiobjective particle swarm optimization description:** The multiobjective optimization can be best handled by use of pareto ranking scheme. The archive or the repository records contain the non dominated solutions and the PSO with global attraction mechanism directs the solution towards the global best non-dominated solutions.

The flowchart is shown below in fig. 6.

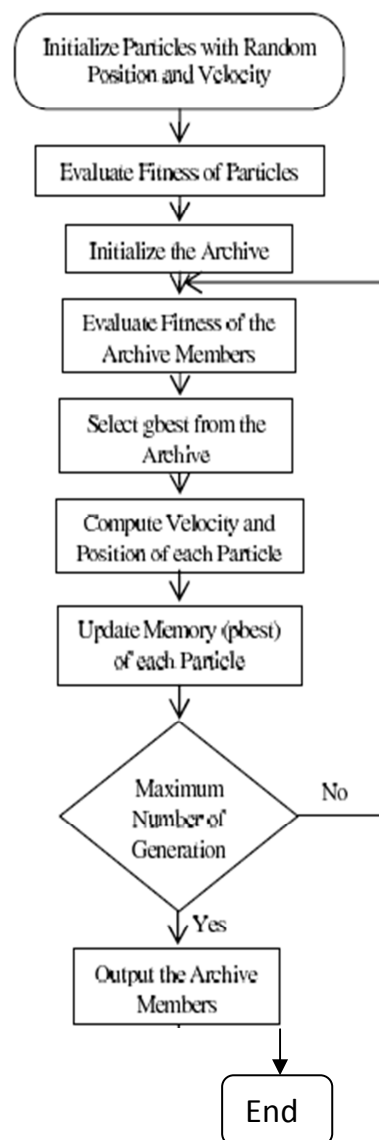


Fig 6: flowchart of multi-objective particle swarm optimization



### 2.3.1 Main Algorithm

The algorithm of MOPSO is the following.

- 1) Initialize the population, POP :
  - a) For counter from 1 to MAX ; where MAX is number of particles
  - b) Initialize POP[counter]
- 2) Initialize the speed of each particle, VEL:
  - a) For counter from 1 TO MAX
  - b) VEL[counter]=0
- 3) Evaluate objective functions for every particles in population POP.
- 4) From the repository REP, the position of the particle representing non-dominated vectors are stored.
- 5) Next, a hypercubes is generated for the search space analyzed till now, and using the values of the objective functions these particles are located in this hypercube. These hypercubes are coordinate systems with objective functions along various dimensions.
- 6) Memory (past) stored in repository is initialized for each particle and these particles moving in the search space are guided using them:
  - a) For counter from 1 TO MAX
  - b) Set PBEST[i]=POP[i]
- 7) WHILE maximum number of cycles has not been reached DO
  - a) Compute the speed of each particle using the following expression:

$$VEL[i]=W*VEL[i]+R1*(PBEST[i]-POP[i])+R2*(REP[h]-POP[i]) \quad (17)$$

Where W (inertia weight) takes a value of 0.4; R1 and R2 are random numbers in the range [0...1] ; PBEST[i] is the best position that the particle i has had; REP[h] is a value that is taken from the repository;

Selecting index  $h$  is done as follows: the selection is based on the fitness value. For each hypercube the fitness value is obtained by dividing a number  $x$  (where  $x$  is any number greater than 1, i.e.,  $x > 1$ ) by the number of particles contained in that hypercube. Thus hypercubes with large number of particles tend to have lower fitness. Next, a selection method called roulette-wheel selection, selects the hypercubes and then a particles corresponding to those hypercubes are selected randomly.

b) New position of particle is computed as follows

$$POP[i]=POP[i]+VEL[i] \quad (18)$$

The speed obtained in the previous step is added to current position of the particle, giving the updated position of the particle.

c) The particles in the search space should remain within the boundaries. All the particles should lie within valid search space. In case, the decision variable does not lie within boundary, then following must be done: 1) the decision variable takes the value of its corresponding boundary (either the lower or the upper boundary) and 2) its velocity is multiplied by (-1) so that it searches in the opposite direction.

d) Each of the particle in population POP is evaluated.

e) Once each of the particle has been evaluated, the hypercube is updated along with the repository. the updation takes place by finding out non-dominated particles in the new population. These new particles are updated and inserted into the repository. The position (i.e., their geographical location) of the new particle in the hypercube is updated. Since the repository is of limited size, the dominated particles need to be constantly removed from the repository. Once, the repository is full the particles with less close neighbours in the objective space are given priority over others placed in high population space.

f) Whenever better current position of a particle compared to the position stored in the memory, then the position of the particle is updated as:

$$PBESTS[i]=POP[i] \quad (19)$$

The decision of updating the particle position is done based on the pareto-dominance concept. In which, if the current position dominates the position stored in memory, then particle position in memory is replaced with the current position; else the position in the memory is retained; if none dominates the other, then one of them is randomly selected.

g) Increment the loop counter; counter = counter + 1

8) END WHILE

### 2.3.2 External Repository

There are historical records corresponding to each non dominated vector. The external repository store these records for each non dominated vector found during the search process. It has two parts: the archive controller and the adaptive grid. [17]

**Archive controller:** In this method, the archive controller acts as a decision maker, taking a decision regarding the acceptance of new solution in the archive. An archive is a vector containing non dominated solutions. The decision process is as follows:

Initially if the repository or the external archive is empty, then current non-dominated vector found is simply added to the archive. If some solutions already exist in the archive then the current nondominated vector found at each iteration, is compared with the contents of the archive. During comparison following cases may occur:

Case1: New solution is dominated by the contents in the archive; discard the new solution.

Case2: New solution is not dominated by any solution in the archive; new solution is added to the archive

Case3: New solution dominates some of the solution in the archive; replace them with the new solution.

Case 4: if external population reaches its maximum limit, then adaptive grid procedure is adopted.

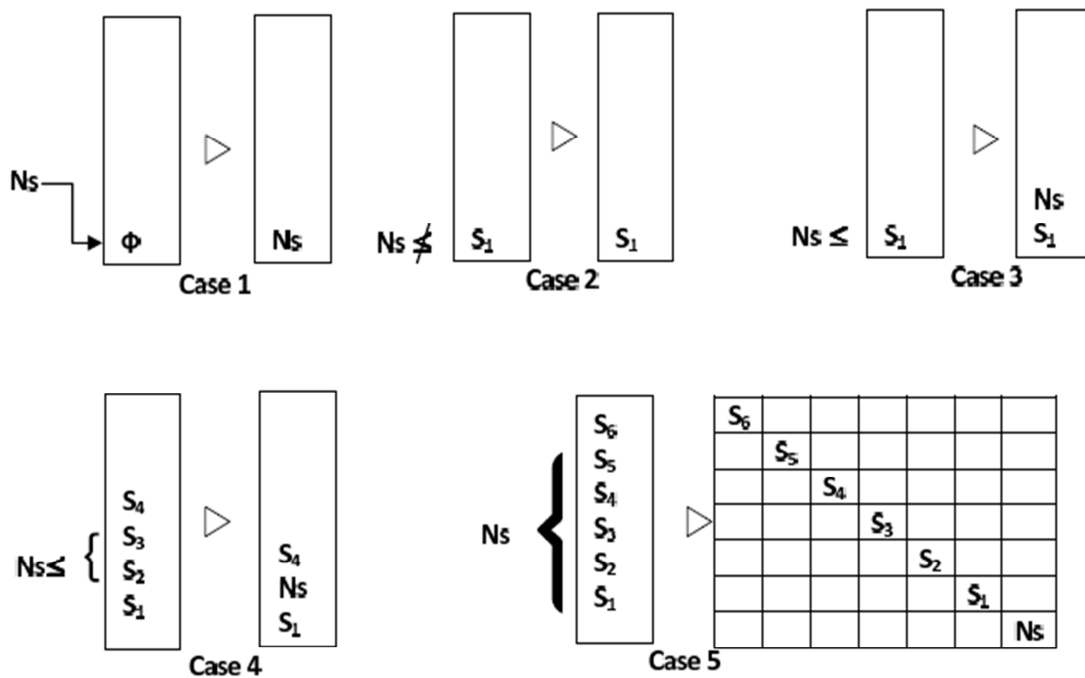


Fig. 7(a): the behavior of archive controller, when new solution is added.

**Adaptive grid:** An adaptive grid is formed from various hypercubes placed adjacent to each other forming a grid. Each hypercube has various components, equal to the number of objective functions. Now if new individual inserted lies outside the grid bounds, then being an adaptive grid, the complete grid is recalculated and each individual within the grid space is relocated.

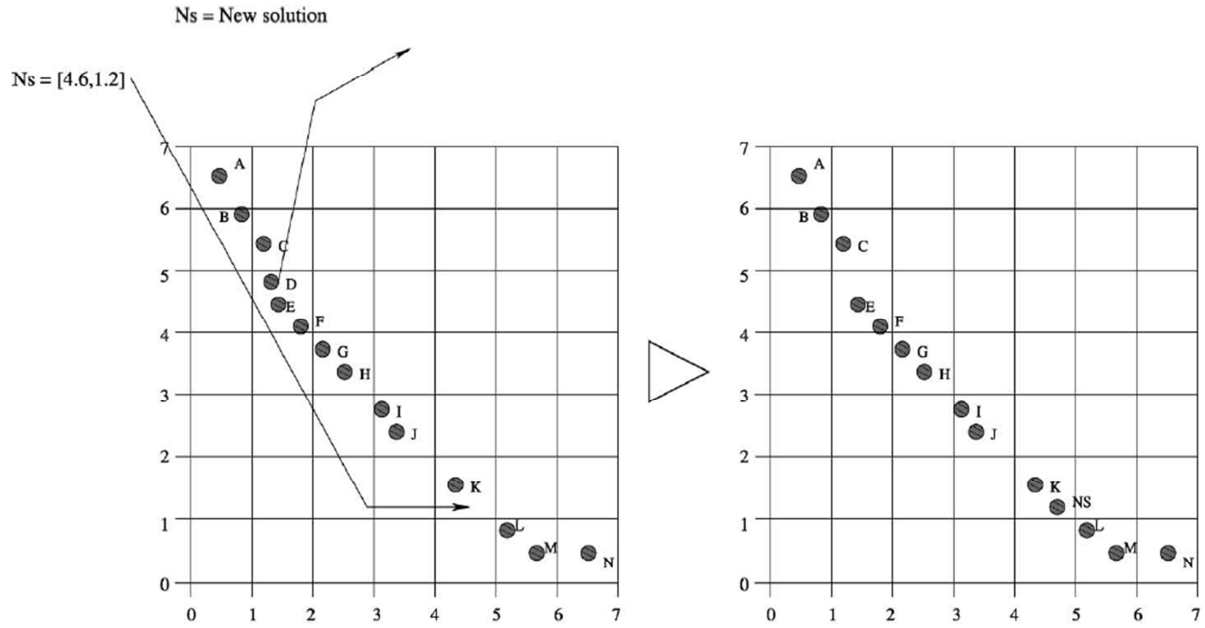


Fig 7(b): New solution belongs within the grid boundaries

Here, as the new solution arrives, it is positioned into the hypercube based on its optimization values, a solution belonging to hypercube with least fitness is removed. Thus maintaining the number of solution same before and after adding a solution within the adaptive cube. As shown in fig. 7(b).

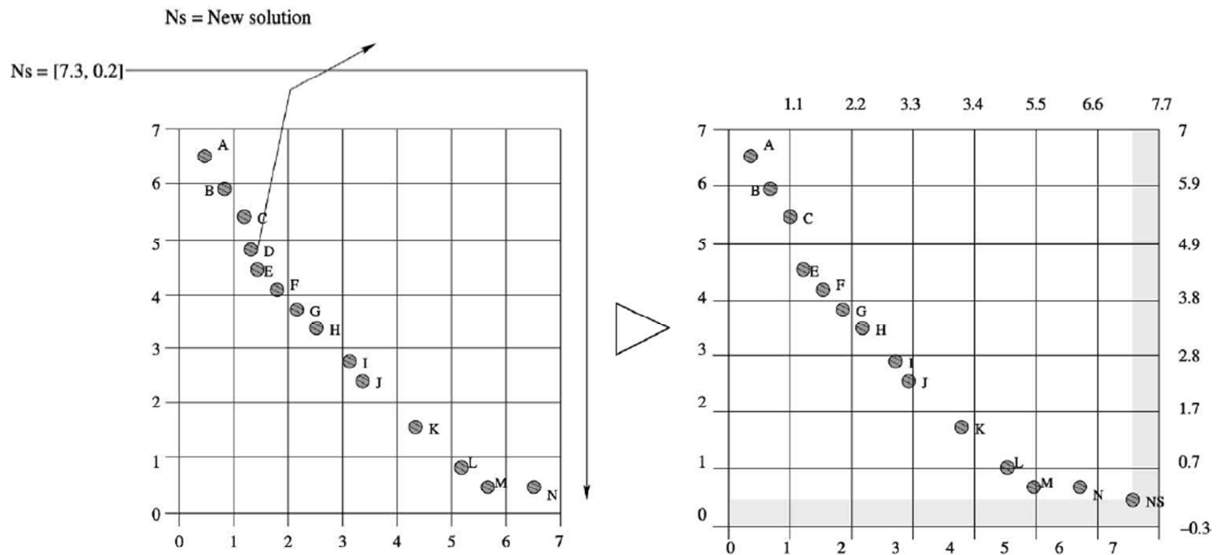


Fig 7(c): New solution lies outside the grid boundaries

In Fig 7(c) the solution does not lie within the boundaries of the hypercube. Thus, the grid being adaptive changes the size of each hypercube, maintaining same number of hypercubes within the grid, so that the new solution can be accommodated within the grid.

## Chapter 3: Proposed Approach

---

The proposed approach for the design of real coefficient filter with arbitrary frequency response is explained below:

FIR filters generally have linear phase and linear group delay. The linearity occurs due to symmetry of coefficients [20, 21]. The proposed approach generates a filter with reduced non-linear group delay and uses optimization functions to make the group delay approximately linear. The lack of symmetry reduces the group delay. [18]

In this approach, the optimization algorithm gives an optimized FIR filter coefficients, by minimizing a set of predefined objective functions. The objective function minimizes the amplitude response error in passband and stopband.

A FIR filter of length  $N$  in  $z$ -transform is represented as:

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n} \quad (20)$$

and the frequency response is represented as:

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n} \quad (21)$$

The frequency response can be expressed as a combination of real and imaginary parts making it a complex frequency response as:

$$\begin{aligned} H(e^{j\omega}) &= \sum_{n=0}^{N-1} h(n) \cos n\omega + j \sum_{n=0}^{N-1} h(n) \sin n\omega \\ &= h^T c(\omega) + j h^T s(\omega) \end{aligned} \quad (22)$$

where,  $h$ ,  $c$  and  $s$  are vectors of coefficients, cosine and sine respectively. These vectors are given as:

$$h = [h(0)h(1)\dots\dots\dots h(N-1)]^T$$

$$c = [1 \cos \omega \dots\dots\dots \cos(N-1)\omega]^T$$

$$s = [0 \sin \omega \dots\dots\dots \sin(N-1)\omega]^T$$

### 3.1 Objective Functions

The objective functions used for optimization using PSO are as follows:

#### 1) The Magnitude Response Error

For passband it is given as:

$$F_p = \max \left| 1 - \left| H(e^{j\omega}) \right| \right| \quad \text{for } 0 \leq \omega \leq \omega_p \quad (23)$$

For stopband error is given as:

$$F_a = \sum_{k=1}^{k_a} \left| H(e^{j\omega}) \right|^2 \quad \text{for } \omega_a \leq \omega \leq \omega_s / 2 \quad (24)$$

Under the constraint that  $A_\omega \leq \delta_a$  for  $\omega_a \leq \omega \leq \omega_s / 2$

#### 2) Group Delay:

$$G = \frac{\max(\text{group delay}) - \min(\text{group delay})}{\max(\text{group delay}) + \min(\text{group delay})}$$

If  $g_1$  is max group delay and  $g_2$  is min group delay, then the equation for  $G$  becomes:

$$G = \frac{g_1 - g_2}{g_1 + g_2} \quad (25)$$

Using above three objective functions, the MOPSO optimizes the coefficients and produces an almost linear group delay.

### 3.2 Parameters Values



- a. The initial values of the parameters of MOPSO are set as follows:
- b. The number of swarms,  $swarm\_size = 50$
- c. Maximum number of iterations,  $itr = 100$
- d. Repository size,  $rep = 70$
- e. Inertia weight,  $w = 0.72$
- f. Grid Inflation Parameter,  $alpha = 0.01$
- g. Number of Grids per each Dimension,  $ngrid = 10$
- h. Leader Selection Pressure Parameter,  $beta = 4$
- i. Extra (to be deleted) Repository Member Selection Pressure,  $gamma = 2$
- j. Inertia Weight Damping Ratio,  $wdamp = 1$
- k. Personal Learning Coefficient,  $c1 = 1.5$
- l. Global Learning Coefficient,  $c2 = 1.5$

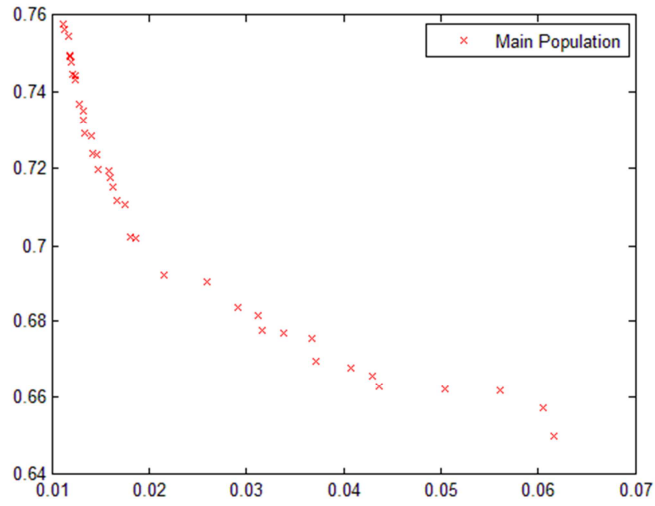
### **3.3 Proposed Method Description**

First, initialize the position and velocity of the particle and the external repository. The position values represent the coefficients. The three objective functions have been explained previously. With the pareto dominance concept, non dominated particles are separated from the universe and stored in the repository. in the hypercube, the value of the objective functions of non dominated particles are located. Initialize the iteration and repeat the following till max iteration is reached: The fittest particle called the leader particle is located from the repository. Update the velocity and position of the particle. Update the repository again with new set of particles.

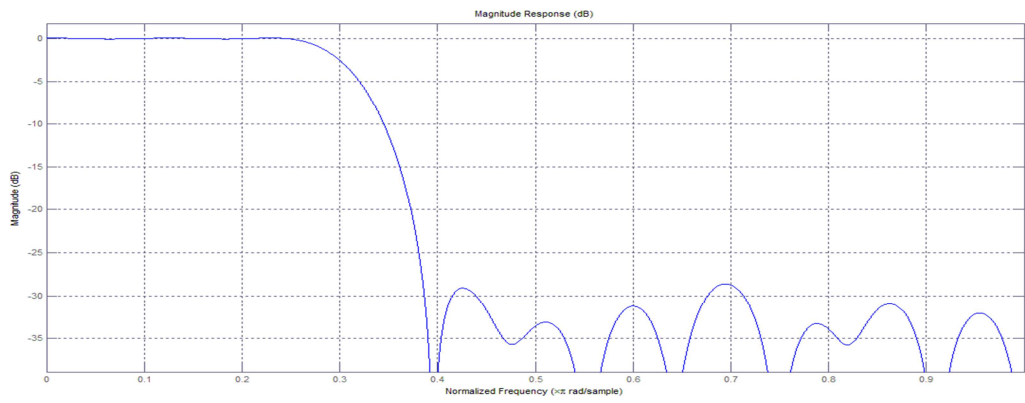
### 4.1 Proposed Method Results

The proposed method is used for designing an optimal FIR filter whose magnitude response approximates an ideal response. The approach has been implemented in MATLAB. Performance of the algorithm depends on the parameters: popsize, iterations, number of coefficients, cutoff frequency, size of repository. For different types of FIR filters, value of certain parameters need to be changed. These parameters are: cutoff frequency, passband frequency, stopband frequency. The approach generates output for all types depending on the cutoff frequency. Also, the approach produces better results when the population size is large and the number of iteration is more. The response of FIR filter depends on the number of coefficients used for FIR filter design. Increasing the number of coefficients improves the magnitude response and also the phase linearity. The outputs shown are magnitude response, group delay and plot showing multiobjective functions values. These outputs are shown in figures 8-10 for lowpass, highpass and bandpass.

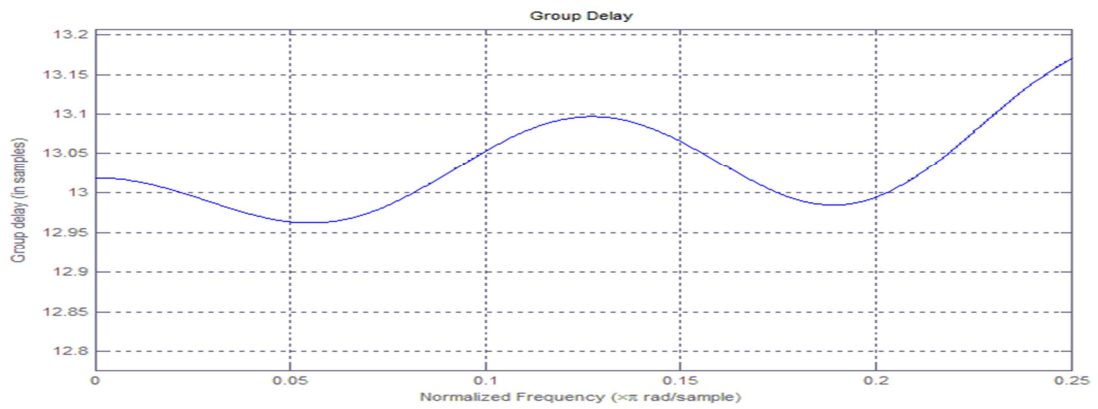
Fig 8 shows the output for a lowpass filter. Fig 8(a) shows the position of particles in objective function space, showing the particles take the position as required in multiobjective algorithms. Fig 8(b) shows the magnitude response, which is approximately 0(dB) in passband and a very low attenuation in stopband. The values for these are given in table 2. Fig 8(c) shows the group delay, which approximates to linearity in passband. A comparison of these results with a previous approach, the WLS approach, has been illustrated in next section. Similarly, fig 9 and 10 shows the output for highpass and bandpass.



(a)

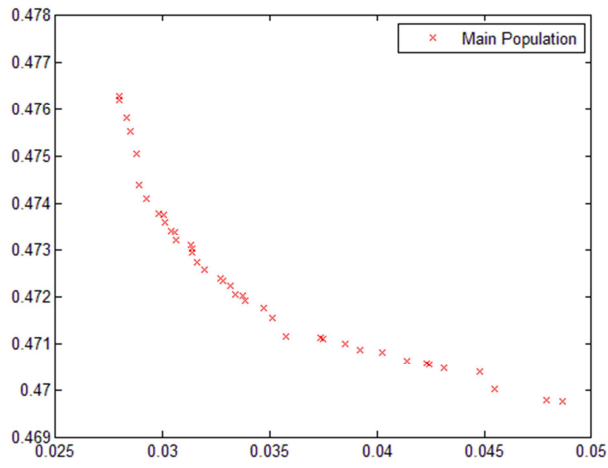


(b)

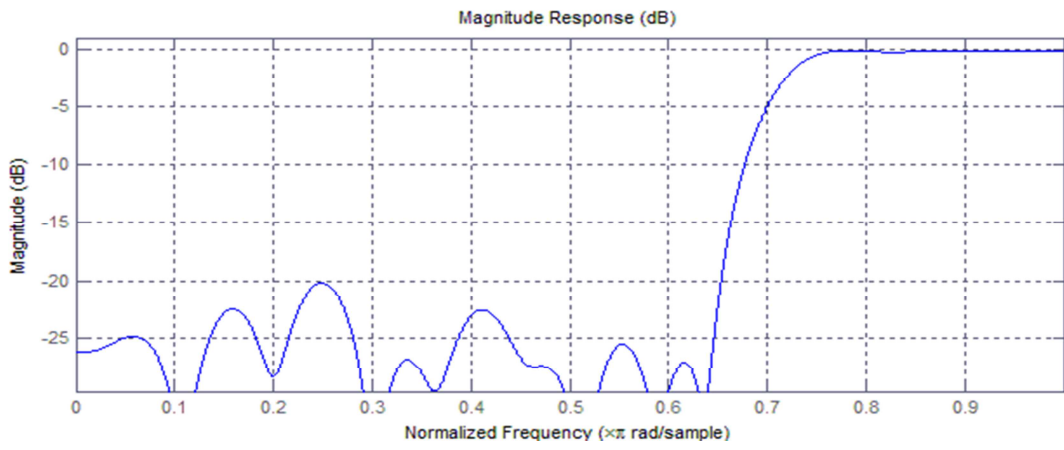


(c)

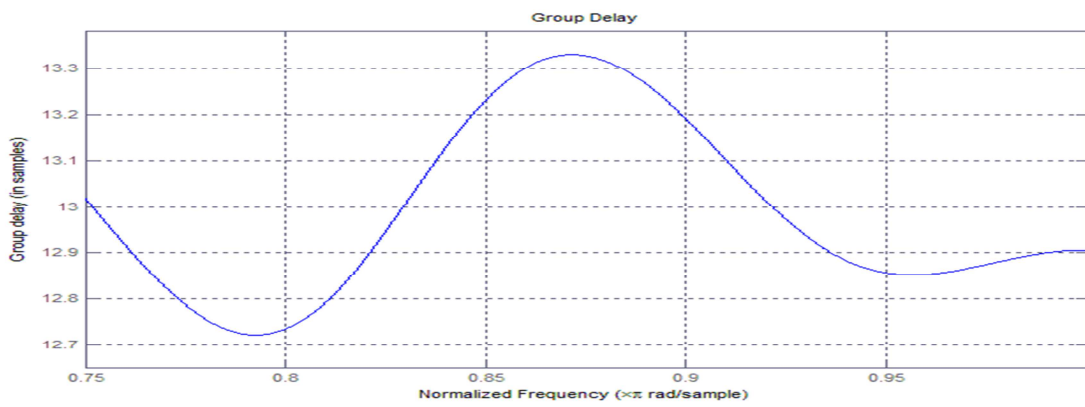
Fig 8: outputs of low pass filter. a) plot of filters objective functions b) magnitude response c) group delay



(a)

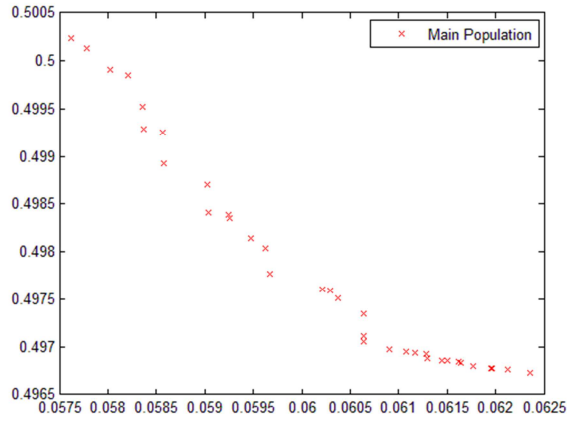


(b)

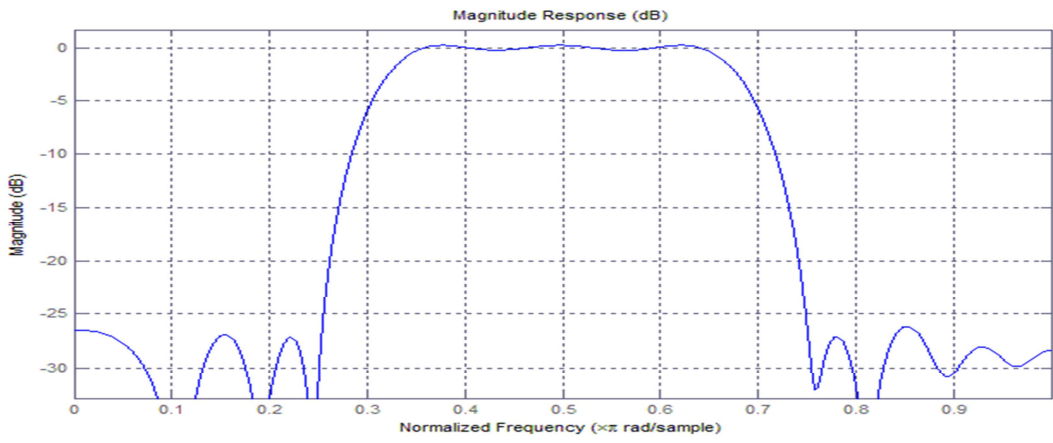


(c)

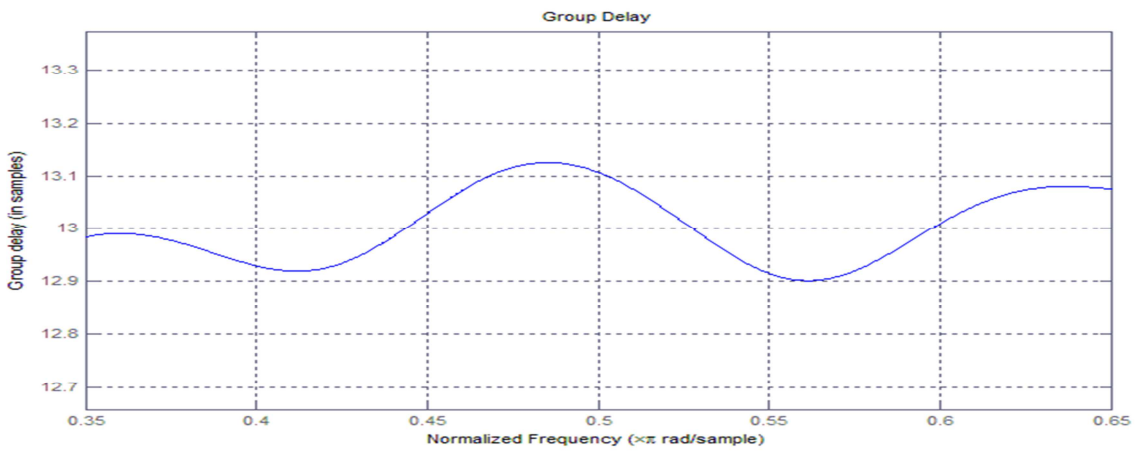
Fig 9: output of a highpass filter.a) plot of filters objective functions b) magnitude response c)group delay



(a)



(b)



(c)

Fig 10: output of a bandpass filter.a) plot of filters objective functions b) magnitude response c)group delay

#### 4.2 Comparison Between the Proposed Approach and WLS Approach

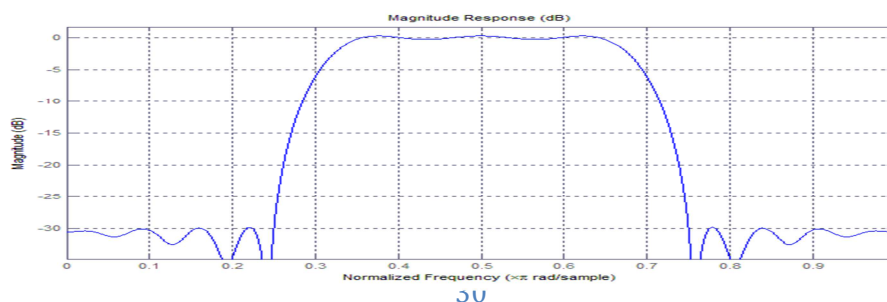
A comparison of the proposed approach with a previous approach (WLS) [8,19] approach has been discussed in this section, giving a theoretical description of how the proposed method outperforms the WLS method. The amplitude response of both the methods along with the group delay is shown. The comparison for both methods has been shown for filter length of 28, and the following passband and stopband edges.

For lowpass passband edge: 0.25 and stopband edge 0.40

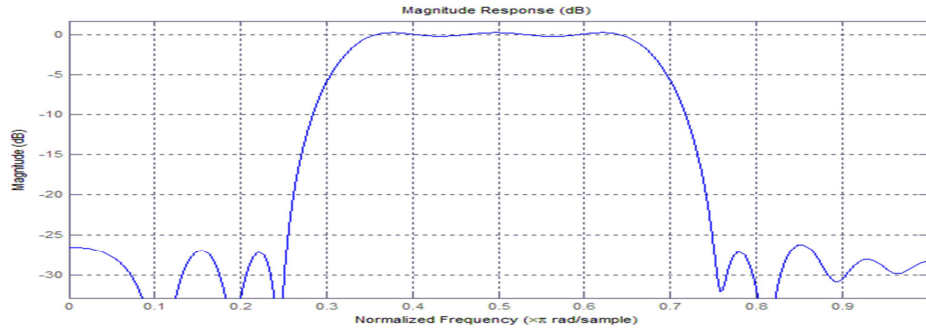
For highpass passband edge: 0.75 and stopband edge 0.65

For bandpass passband edge: 0.35 and 0.65, and stopband edge 0.40

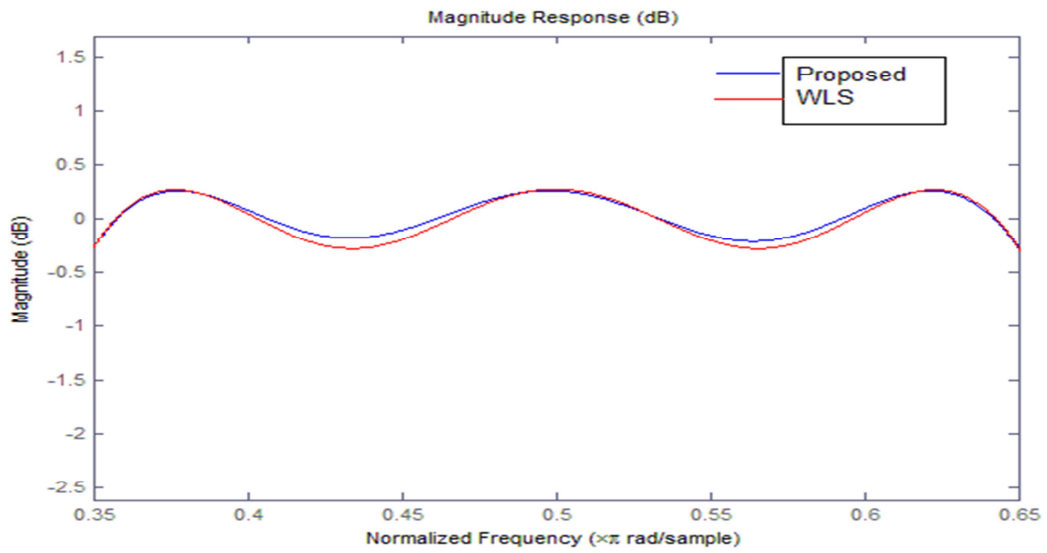
Fig 11-16 show the comparison for the results obtained by WLS and proposed method. The comparison has been made on the basis of the three objective values viz. passband ripple, stopband attenuation and group delay.



(a)

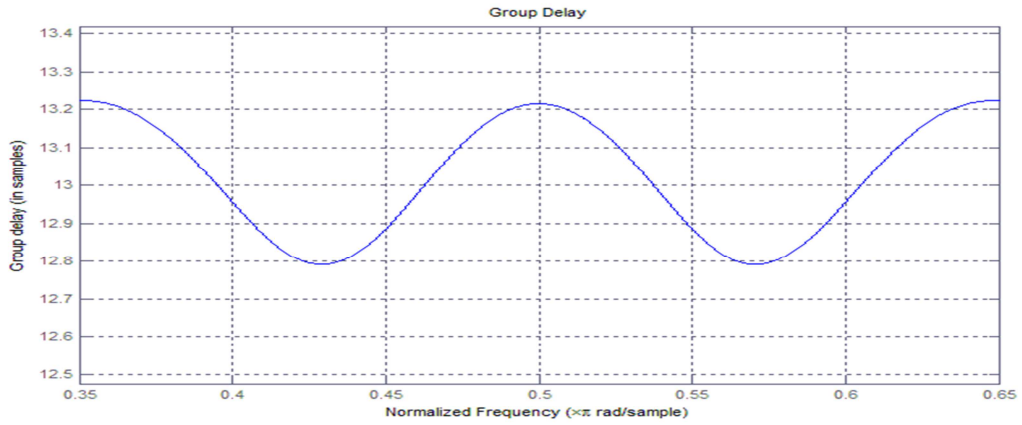


(c)

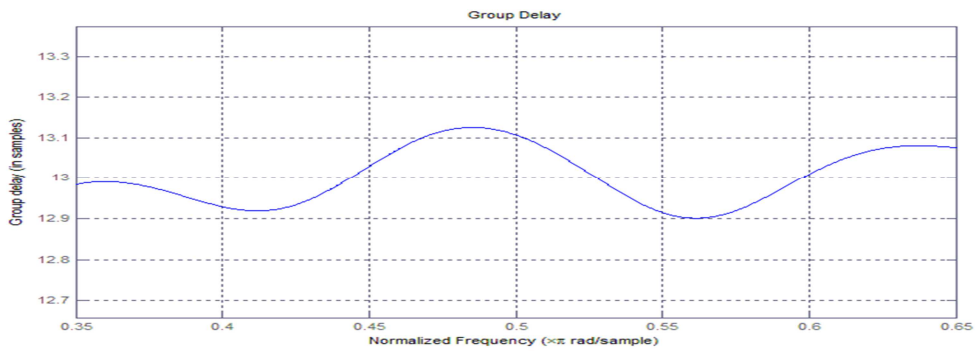


(c)

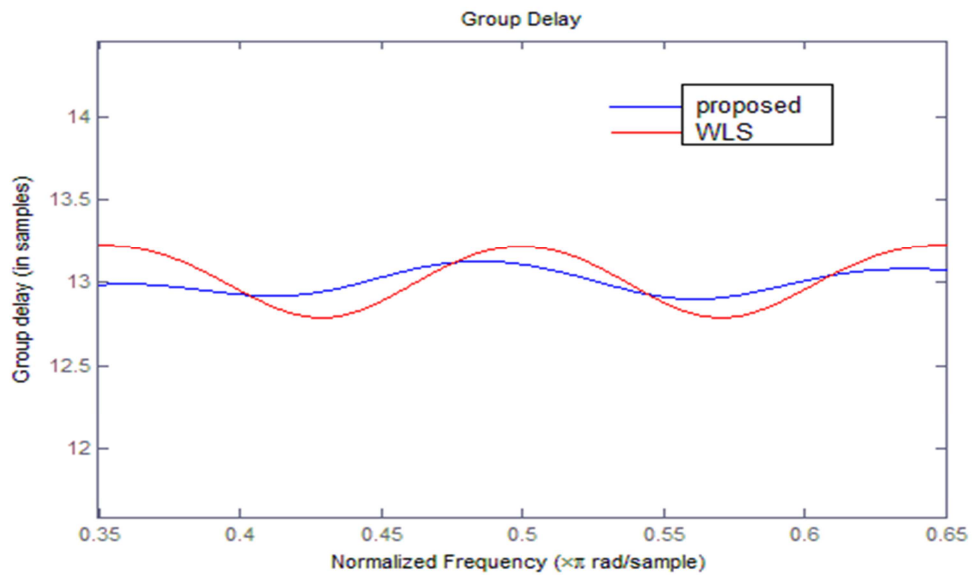
Fig 11: bandpass filter.a) magnitude response for WLS approach b) magnitude response for proposed approach c) comparison of both approaches in passband



(a)



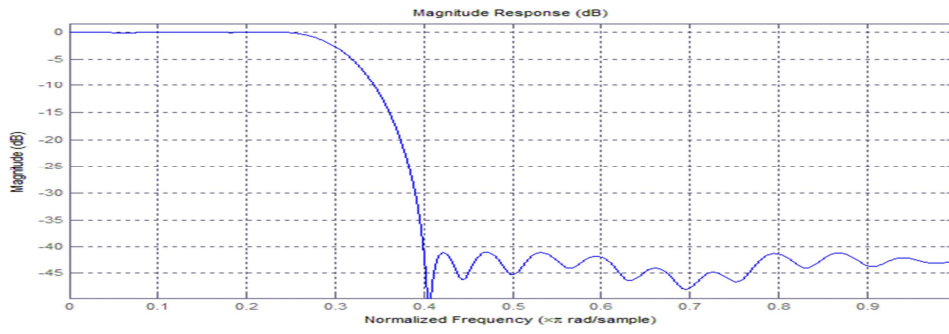
(b)



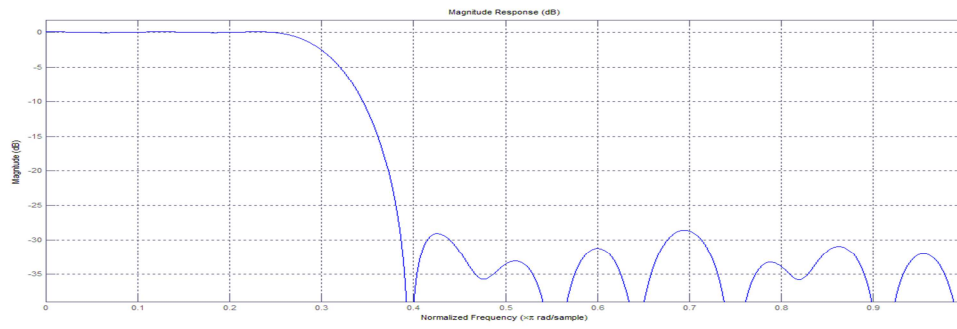
(c)

Fig 12: Group delay of a bandpass filter a) using WLS method b) using proposed method c) showing comparison of both approaches

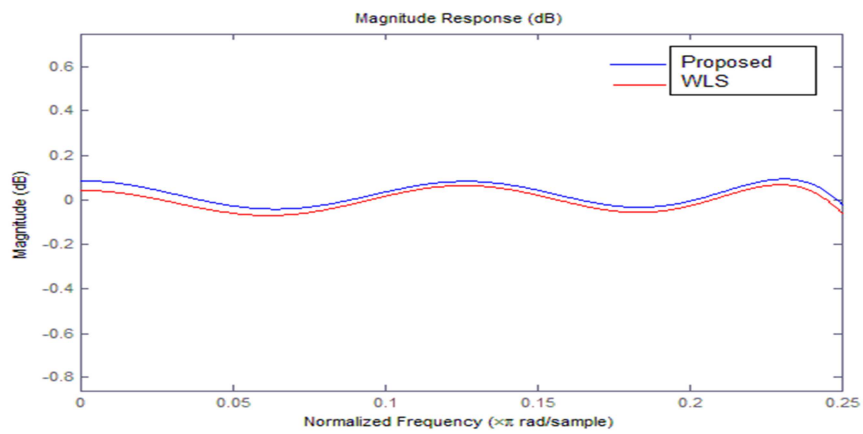




(a)

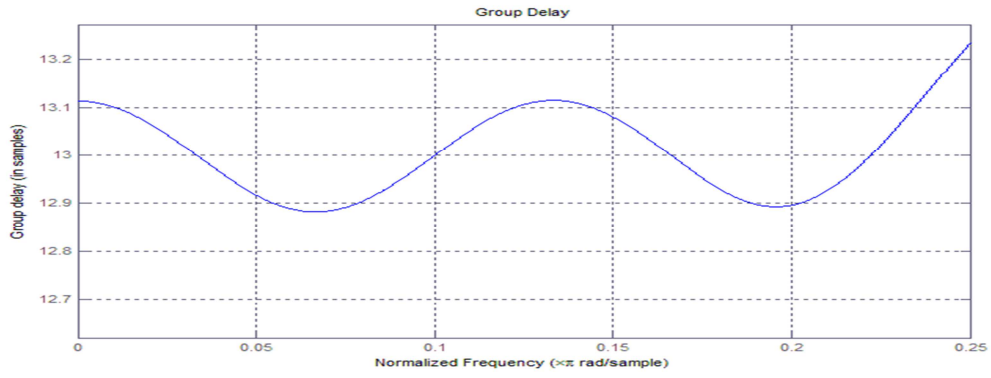


(b)

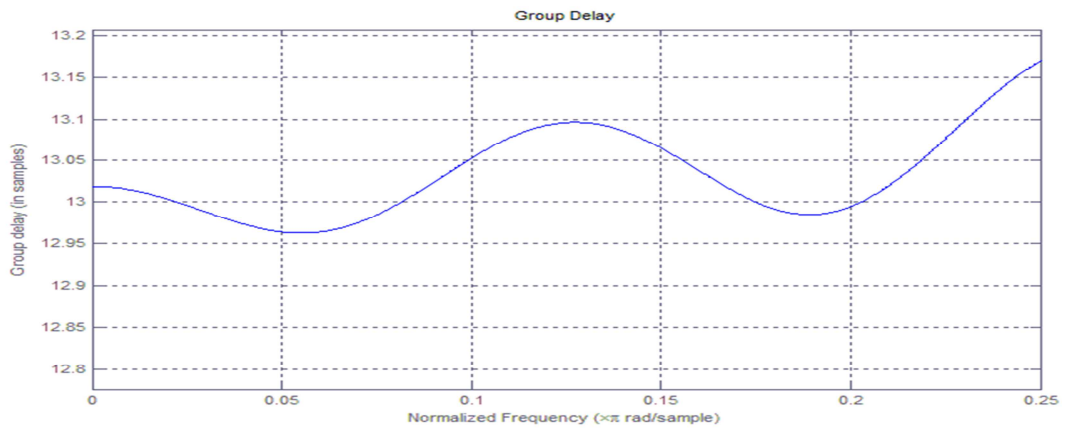


(c)

Fig 13: lowpass filter a) magnitude response for WLS approach b) magnitude response for proposed approach c) comparison of both approaches in passband



(a)



(b)

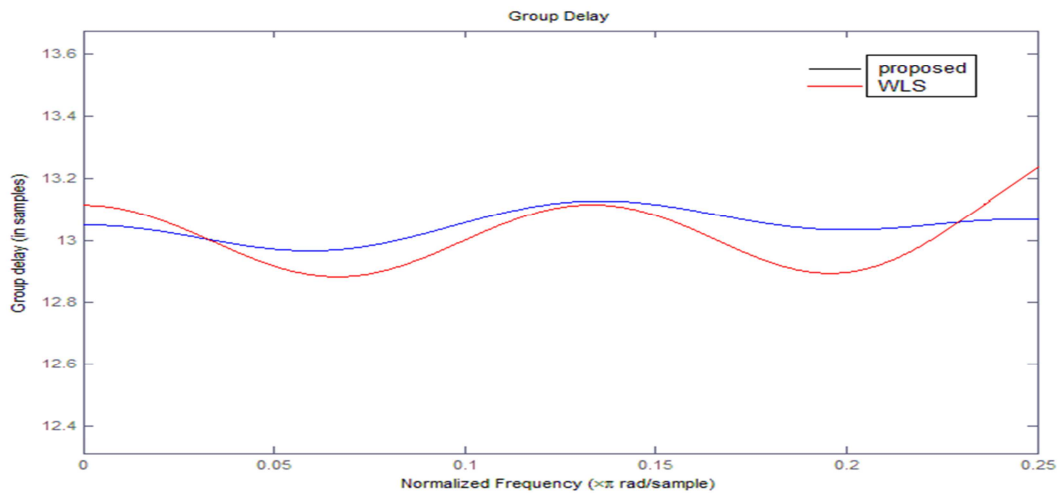
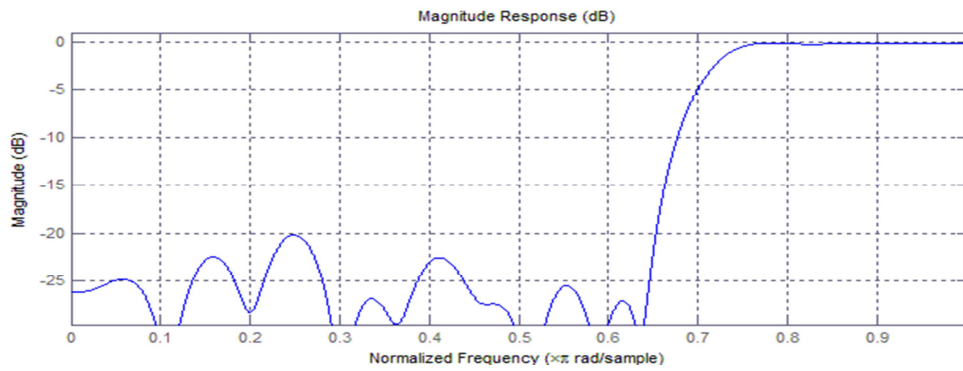
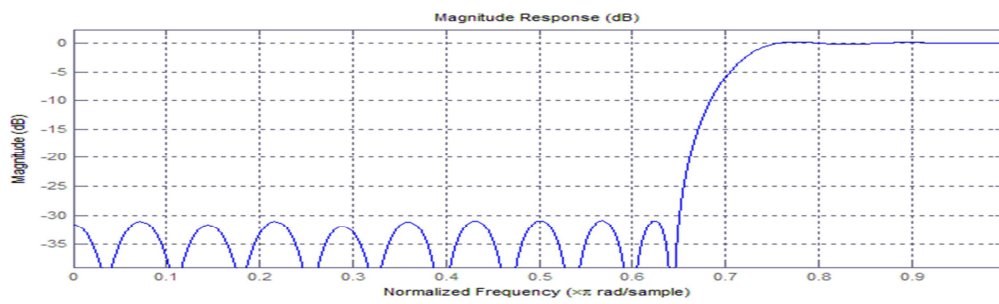


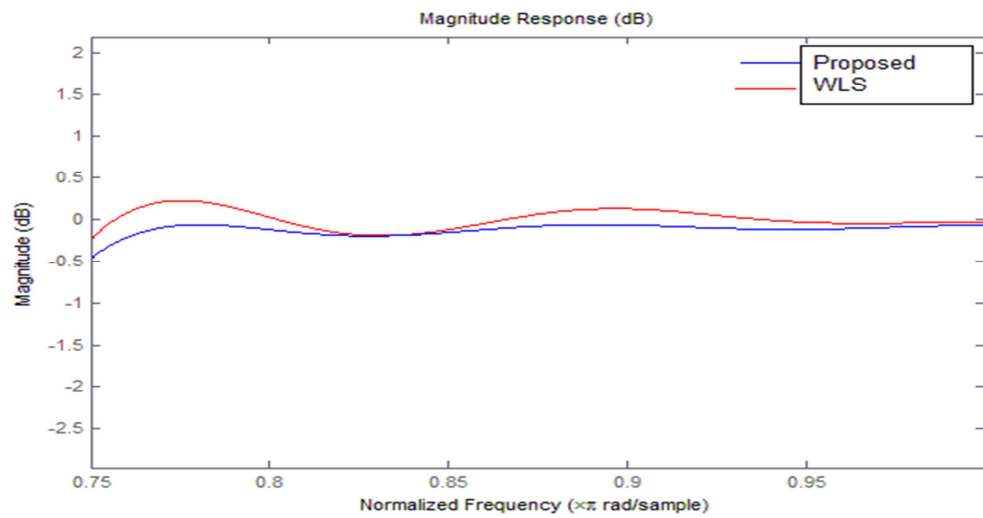
Fig 14: group delay of a lowpass filter a) using WLS method b) using Proposed method c) showing the comparison using both methods in passband



(a)

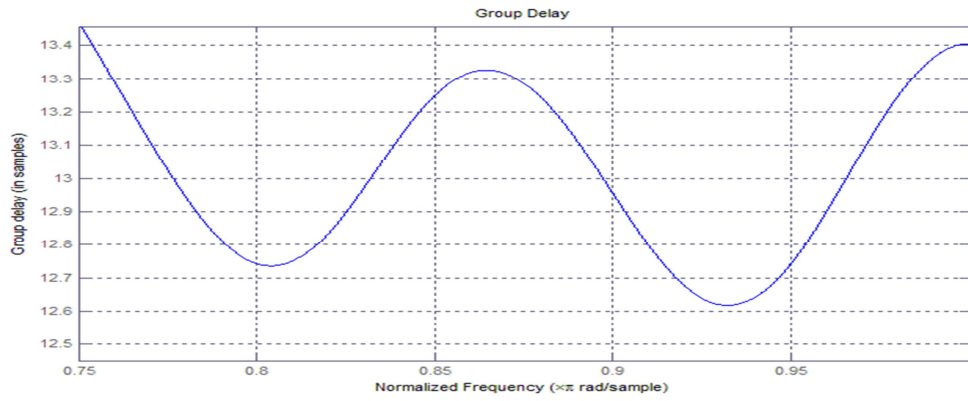


(b)

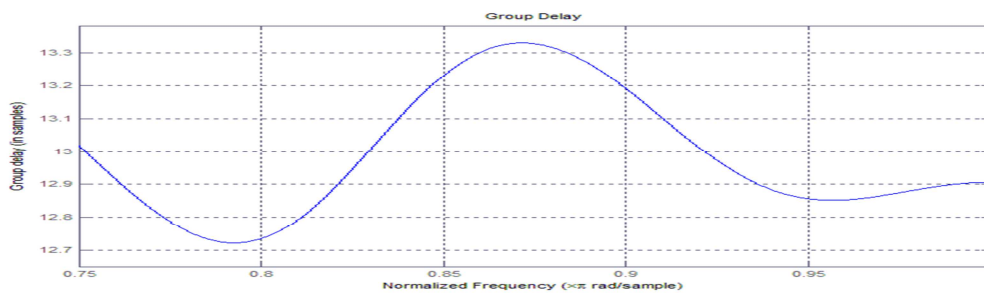


(c)

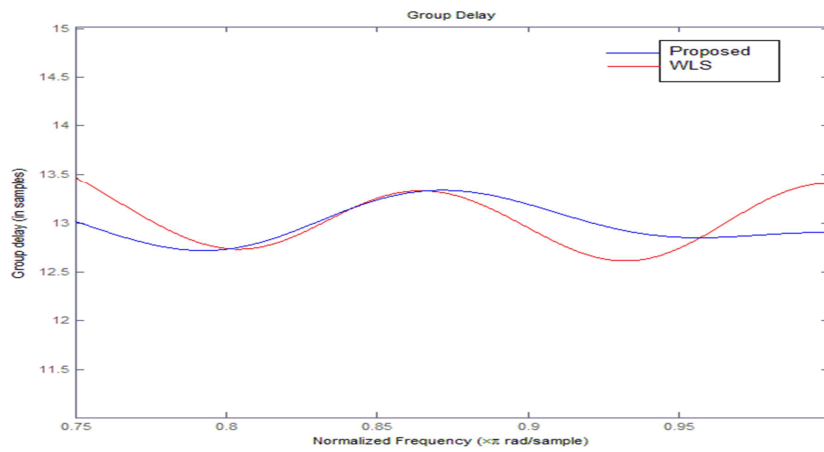
Fig 15: output of a highpass filter a) magnitude response for proposed approach b) magnitude response for WLS approach c) comparison of two approaches in passband



(a)



(b)



(c)

Fig 16: group delay of a highpass filter a) using WLS method b) using proposed approach c) showing comparison of both approaches

Table 2-4 below shows the value of passband ripple, stopband attenuation and group delay for lowpass, bandpass and highpass filters of length 28 using WLS and proposed approach, showing the comparison of the two approaches.

Table 2: Values for lowpass filter of : Passband edge: 0.25; Stopband edge: 0.40; Length: 28

	WLS	Proposed method
Passband Ripple	0.686	0.669
Stopband attenuation	44.01	43.17
Group delay in passband	1.2	0.8

Table 3: Values for bandpass filter of : Passband edge: 0.35 and 0.65; Stopband edge : 0.25 and 0.75; Length : 28

	WLS	Proposed method
Passband Ripple	0.598	0.45
Stopband attenuation	43.5	43.07
Group delay in passband	1.31	0.9

Table 4: Values for highpass filter of : Passband edge: 0.75; Stopband edge : 0.65; Length : 28

	WLS	Proposed method
Passband Ripple	0.265	0.243
Stopband attenuation	49.04	48.39
Group delay in passband	0.987	0.86

The comparison of the proposed approach with the WLS method shows that the amplitude response in the passband has less ripples for the proposed method. Also the magnitude response has steep curve at the passband edge frequency. The group delay is much more linear in the passband for the proposed approach as compared to stopband. The value shown in table 2, 3, 4, show the amplitude response and group delay values for both the methods, from those values it can be seen that the proposed approach outperforms the WLS method.

## Chapter 5: Conclusion

---

The proposed work focuses on reducing the delay while reducing the ripple in passband of the amplitude response. It uses multiobjective particle swarm optimization to minimize the error functions for amplitude response and group delay. The error function for amplitude response is, to reduce the ripple in passband and attenuation in stopband. The group delay error function reduces the non linearity in group delay. Also, comparison with a previous approach shows, reduced ripple in passband of the amplitude response and linearity in group delay.

The performance of the approach depends on various parameters  $\alpha$ ,  $\beta$ ,  $\gamma$ ,  $c1$ ,  $c2$ ,  $w_{damp}$ ,  $\text{rep}$  and  $w$ , contributing to the success of the proposed method.

The future scope of the work is to obtain optimum filters using the other evolutionary algorithms with multiobjective problem solving capability. Also, to extend the proposed method for multiband pass filters and IIR filters.

## References

---

- [1] T.W. Parks, J.H. McClellan, "Chebyshev approximation for non-recursive digital filters with linear phase", *IEEE Trans. Circuits Theory CT-19* (1972), pp. 189–194.
- [2] Rajib Kar, Durbadal Mandal, Dibbendu Roy, Sakti Prasad Ghoshal, "FIR Filter Design using Particle Swarm Optimization with Constriction Factor and Inertia Weight Approach," *ACEEE Int. J. on Electrical and Power Engineering*, Vol. 02, No. 02, August 2011.
- [3] Hung-Ching Lu, Shian-Tang Tzeng, "Design of arbitrary FIR log filters by genetic algorithm approach, *Signal Processing*", 2000, 80, pp. 497-505.
- [4] D. Karaboga, D.H. Horrocks, N. Karaboga, A. Kalinli, Designing digital FIR filters using Tabu search algorithm, *IEEE International Symposium on Circuits and Systems*, 1997, vol.4, pp.2236-2239.
- [5] S. Mandal, S. P. Ghoshal, R. Kar, D. Mandal, "Optimal Linear Phase FIR Band Pass Filter Design using Crazyness Based Particle Swarm Optimization Algorithm", *Journal of Shanghai Jiaotong University (Science)*, Springer. Vol. 16, No. 6, pp. 696-703, 2011.
- [6] J. I. Ababneh, M.H. Bataineh. "Linear phase FIR filter design using particle swarm optimization and genetic algorithm". Elsevier, *Digital Signal Processing*, 2006.
- [7] Ahmad, S.U., [Antonio A.](#), "A Multiobjective Genetic Algorithm for Asymmetric FIR Filters", *IEEE International Symposium on signal processing and information technology*, pp: 525 – 530, 2007.
- [8] S. K. Kidambi and R. P. Ramachandran, "Design of nonrecursive filters satisfying arbitrary magnitude and phase specifications using a least-squares approach," *IEEE Trans. Circuits Syst. - II: Analogue and Digital Signal Proc.*, vol. 2, no. 11, pp. 711-716, Nov. 1995.

- [9] M. Lang, "Algorithms for the Constrained Design of Digital Filters with Arbitrary Magnitude and Phase Responses," Vienna University of Technology, Institute for Communications and Radio-Frequency Engineering, 1999.
- [10] Deb, K., Pratap. A, Agarwal, S., and Meyarivan, T., "A fast and elitist multi-objective genetic algorithm: NSGA-II", *IEEE Transaction on Evolutionary Computation*, pp. 181-197, 2002
- [11] Deb. K., "An efficient constraint-handling method for genetic algorithms", *Comput. Methods Appl. Mech. Eng.*, vol. 186, no. 2-4, pp. 311-338, 2000.
- [12] J. Horn, N. Nafplotis, and D. E. Goldberg, "A niched Pareto genetic algorithm for multiobjective optimization", *IEEE Conference on Evolutionary Computation*, Z. Michalewicz, Ed. Piscataway, NJ: IEEE Press, pp. 82-87, 1994.
- [13] J. Knowles and D. Corne, "The Pareto archived evolution strategy: A new baseline algorithm for multiobjective optimization", *Proceedings of Congress on Evolutionary Computation. Piscataway, NJ: IEEE Press*, pp. 98-105, 1999.
- [14] Carlos A. Coello Coello, Gregorio Toscano Pulido, and Maximino Salazar Lechuga, "Handling multiple objectives with particle swarm optimization", *Evol. Comput.*, vol. 8, pp. 256-279, NO. 3, June 2004.
- [15] Kennedy, R. Eberhart, "Particle Swarm Optimization," *IEEE int. Conf. On Neural Network*, 1995.
- [16] A. Coello Coello and M. S. Lechuga, "MOPSO: A proposal for multiple objective particle swarm optimization", *Proc. Congr. Evolutionary Computation (CEC'2002)*, vol. 1, Honolulu, HI, pp. 1051-1056, May 2002.



- [17] Carlos A. Coello Coello, Gregorio Toscano Pulido, Maximino Salazar Lechuga, "Handling Multiple Objectives With Particle Swarm Optimization", *IEEE transactions on evolutionary computation*, Vol. 8, No. 3, June 2004.
- [18] M. C. Lang, "An iterative reweighted least-squares algorithm for constraint design on nonlinear phase FIR filters," *Proc. IEEE Int. Symp., Circuits and Systems*, vol. 5, pp. 367-370, 1995.
- [19] M. Lang, "Algorithms for the Constrained Design of Digital Filters with Arbitrary Magnitude and, Phase Reponse", Ph.D. Thesis, Jun. 1999.
- [20] A. Antoniou, *Digital Signal Processing: Signals, Systems, and Filters*. New Jersey: McGraw-Hill, 2005.
- [21] T. W. Parks and C. S. Burrus, *Digital Filter Design*. New York: Wiley, 1987.
- [22] K. D. Jong, "Hybrid methods using genetic algorithms for global optimization," *IEEE Trans. Systems Man Cybernetics*, vol. 10, no. 9, pp. 566 {574, September 1980.
- [23] M. Mitchell, "An Introduction to Genetic Algorithms", Cambridge, MA: MIT Press, 1996.
- [24] Deb. K. *Optimization for Engineering Design: Algorithms and Examples*. New Delhi: Prentice-Hall, Feb 2004.
- [25] Schaffer, J. D. "Some Experiments in Machine Learning Using Vector Evaluated Genetic Algorithms". Ph. D. Thesis, Nashville, TN: Vanderbilt University, 1984.