



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2014

Review of Various Optimization Techniques for FIR Filter Design

Swati Chowdhary¹, Prof.Mamta M Sarde²

PG Student [ECE], Dept. of ECE, ABHA Gaikwad Patil College of Engineering, Nagpur, India¹

Assistant professor, Dept. of ECE, ABHA Gaikwad Patil College of Engineering, Nagpur, India

ABSTRACT: The basic problem is to identify the means to achieve accurate coefficient and then to select the objective required in efficient manner and for this optimization is the best technique which is defined as selection of a best element from some set of available alternatives. An optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function. In this paper there is a study of different algorithms and their classification used in digital filter design to get the coefficient accurate.

Keywords: Optimization, Simulated Annealing Algorithm, Differential Evolution Algorithm, Genetic Algorithm.

I.INTRODUCTION

Optimization is simply choosing a design parameters to improve some objective. It is extracting some model parameters from data while minimizing some error measure. An optimization problem is the problem of finding the best solution from all feasible solutions. Optimization problems can be divided into two categories depending on whether the variables are continuous or discrete. An optimization problem with discrete variables is known as a combinatorial optimization problem. In a combinatorial optimization problem, we are looking for an object such as an integer, permutation or graph from a finite (or possibly countable infinite) set.

The standard form of a (continuous) optimization problem is

Minimize $f(x)$

subject to $g_i(x) \leq 0, i=1, \dots, m$ (1)

$H_i(x)=0, i=1, \dots, p$ (2)

where

$f(x): \mathbb{R}^n \rightarrow \mathbb{R}$ is the objective function to be minimized over the variable x ,

$g_i(x) \leq 0$ are called inequality constraints, and

$h_i(x) = 0$ are called equality constraints.

By convention, the standard form defines a minimization problem. A maximization problem can be treated by negating the objective function

Formally, a combinatorial optimization problem A is a quadruple (I, f, g, m) , where given an instance x and a feasible solution y of x , $m(x, y)$ denotes measure of y which is usually a positive real.



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2014

I is a set of instances, is the goal function, and is either min or max.

The goal is then to find for some instance x an *optimal solution*, that is, a feasible solution y with

$$m(x, y) = g \{m(x, y') | y' \in f(x)\} \quad (3)$$

For each combinatorial optimization problem, there is a corresponding decision problem that whether there is a solution for some particular measure m_0 . Hence optimization problem is based on computational time which is required to solve the problem. The classification of optimization is in next section.

A. Classification of Optimization Problem:

i) Classification of continuous optimization problems:

Optimization problems are classified as either convex or non-convex based on whether the domain and the cost function are both convex or not. The domain is convex if a straight line between any two points s_1 and s_2 in the domain is also part of the domain, and the cost function is convex if its value at any point along the straight line between any two points s_1 and s_2 in the domain has an upper bound in the chord through $(s_1; f(s_1))$ and $(s_2; f(s_2))$.

ii) Classification of combinatorial optimization problems:

The classification of combinatorial optimization problems, which is completely different from that of continuous problems, is based on the computational time needed to solve problems as the size grows. A distinction is made between problems which has a solution time, with respect to the best known algorithm, that is a polynomial function of the problem size and those which require a super polynomial, e.g. any function that grows faster than a polynomial, execution time in terms of their size. [1]

II. OPTIMIZATION ALGORITHMS

An optimization algorithm is related to the particular optimization problem at hand, but in general this algorithms applicable only to a subset of a problem to general algorithms tackling many problems without modification. The general algorithm is time computing time while an efficient general algorithm "fast enough" is in most cases. There are some algorithms which are used to compare complexity and executing time required are elaborated below.

B. Simulated Annealing (SA) algorithm:

The simulated annealing (SA) optimization algorithm is based on an analogy between the behaviors of a melted solid that is slowly cooled (annealed) into a "perfect" crystal. The algorithm, together with genetic and neural network algorithms, belongs to a group of algorithms sometimes referred to as "natural" algorithms since they rely on observations made in the study of natural systems such as the forming of crystals, the evolution of the species, and the workings of the human brain. [1]

The key algorithmic feature of simulated annealing is that it provides a means to escape local optima by allowing hill-climbing moves (i.e., moves which worsen the objective function value). As the temperature parameter is decreased to zero, hill climbing moves occur less frequently, and the solution distribution associated with the in homogeneous Markov chain that models the behavior of the algorithm converges to a form in which all the probability is concentrated on the set of globally optimal solutions [2]

C. FIR filter design with simulated annealing algorithm:

Boltzmann annealing (Ingber, 1989), which is defined by the following elements:

$$G(b_n) = [(2\pi)^{-N/2}] e^{-b^2/2T} \quad (4)$$



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2014

Where $g(b_n)$ is the probability density function (pdf) of the filter coefficients b_n deviation, $\Delta b_n = b_n^{(t+1)} - b_n$ is the deviation from state (filter) i to $i+1$, and T is a measure of the fluctuations of the Boltzmann distribution g in the $N-1$ $h(b_n) = 1/(1+e^{\Delta E/T})$ where $h(b_n)$ is the (PDF) probability for acceptance of new between the present and the previous values of the cost-function [2] i.e. filter error $T(k) = T_0 / \ln k$, where $T(k)$ is the schedule of annealing the temperature T in annealing-time steps. Simulated annealing is beneficial with arbitrary systems and cost functions for any given problem as well as it statistically guarantees finding an optimal solution for any optimization problem, besides this if annealing with a $(1/\log k)$ is very time consuming for complex cost function makes major drawback of SA. [3]

D. Differential Evolution(DE) Algorithm:

DE is very simple yet robust, powerful, stochastic, population based and easy to use optimization algorithm, which has been developed to optimize real parameter and real valued functions. The name of the algorithm is chosen as Differential Evolution (DE) to indicate a special type of differential operator in it that has been utilized to

Create offspring from the parent chromosomes without adopting crossover or mutation.
Differential Evolution algorithms can be divided into four steps, namely Initialization, Mutation, Recombination, and Selection.

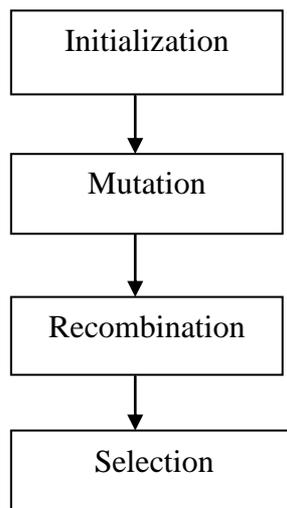


Fig1. Differential Evolution process

- **Initialization**

To obtain better result choose a suitable range of value which specify the search variables to optimize this technique. Thus if the j th gene of each chromosome has its lower and upper bound as x_j^L and x_j^U respectively, then the corresponding gene of i th chromosome can be initialized as

$$X_{ij}(1) = x_j^L + \text{rand}(0,1) \cdot (x_j^U - x_j^L) \tag{5}$$

where $i = 1, 2, 3, \dots, P$ and $\text{rand}(0, 1)$ is a uniformly generated random number in $[0, 1]$.



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2014

• Mutation

Initialization process only takes care of a entire search space. To find the optimum solution for a given problem there is a need to explore the entire search space which can be expanded in the process of mutation to generate a new population vector in which various parameter vectors are combined together.

DE technique is termed for different mutation strategies.

In DE/rand/1 scheme, donor vector of generation 'G+1' can be formed from the population vector of the generation 'G' as

$$V_i(G+1)_{rand/1} = X_{p1}(G) + F \cdot [X_{p2}(G) - X_{p3}(G)] \quad (6)$$

F is called the weighting factor with $i, p1, p2, p3 \in P$ where P is the set of population members of the evolutionary computation. The parameter

F is responsible for the amplification of the differential variation and generally lies within the range [0, 2].

• Recombination

This is the third step of DE, called cross-over or recombination. Recombination process plays an important role enhance the potential diversity of the population which is a scheme of crossover where donor vector exchanges its components. Two types of recombination exist in the literature, namely exponential and binomial cross-over. The trial vector

$U_i(G+1)$ case of exponential crossover has the form

$$U_i(G+1) = [U_{i,1}(G+1), U_{i,2}(G+1), \dots, U_{i,D}(G+1)] \quad (7)$$

$$U_{i,j}(G+1) = \begin{cases} v_{i,j}(G+1); & j = \text{mod}(n-k+1, D) \\ x_{i,j}(G); & \text{otherwise} \end{cases} \quad (8)$$

where $k=1, 2, \dots, L$ and L is another integer from the interval [1, D]. The parameter L signifies the number of components contributed by the donor vector to the target vector.

• Selection

The final decision is taken at the last step of DE, called selection process. The present parameter vector passes through the next generation. This can well be described by means of the following equation

$$X_i(G+1) = \begin{cases} U_i(G+1); & \mathfrak{F}(U_i(G+1)) < \mathfrak{F}(X_i(G+1)) \\ X_i(G); & \text{otherwise} \end{cases}$$

$$X_i(G+1) = \begin{cases} X_i(G); & \mathfrak{F}(X_i(G)) < \mathfrak{F}(X_i(G+1)) \\ U_i(G+1); & \text{otherwise} \end{cases} \quad (9)$$

Where $\mathfrak{F}(Y_i(G))$ identifies the cost functional value resulting from the i th vector $Y_i(G)$ at iteration 'G'. [5]

E. FIR filter design with Differential Evolution Algorithm:

For designing low pass FIR filter different mutation schemes are used. In optimization algorithm, the efficiency of DE is presented by a mathematical expression called the cost function. Here the fitness value and cost functional value may be regarded as two different functions of identical items as follows:

$$\text{Fitness value} = \Psi(\phi, M, R, \zeta) \quad (10)$$

$$\text{Cost functional value} = \sigma(\phi, M, R, \zeta) \quad (11)$$

The parameter ' ϕ ' in and reflects the formulation of the error function and for the design of a low-pass filter with a cut-off frequency ω takes the form as shown in the following equation:

$$\Phi(k, p_i) = \begin{cases} \min_i [\max_k \{1 - H_{pi}(k)\}] & \text{for } k \leq \omega N / 2\pi \\ \min_i [\max_k \{H_{pi}(k)\}] & \text{for } k \leq \omega N / 2\pi \end{cases} \quad (12)$$



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2014

The other parameters within the parenthesis in symbolize the set of mutation strategy, recombination strategy and control parameter and has been outlined [5] as follows:

$M = \{\text{rand}/1, \text{rand}/2, \text{best}/1, \text{best}/2, \text{rand to best}/1\}$

$R = \{\text{binomial}, \text{exponential}\}$

$\zeta = \{\text{weighting factor(s)}, \text{recombination probability}\}$ (13)

F. Genetic Algorithm

Genetic Algorithm (GA) are very flexible, non-problem specific, and robust. GA can explore multiple regions of the parameter space for solutions simultaneously. GA can discard suboptimal solutions in favour of more promising subsequent local solutions. They require a very large amount of computation. The GA consists four main stages: evaluation, selection, crossover and mutation. The evaluation procedure measures the fitness of each individual solution in the population and assigns to it a score. The selection procedure randomly selects individuals of the current population for development of the next generation. The crossover procedure takes two selected individuals and combines them about a crossover point thereby creating two new individuals. The mutation procedure randomly modifies the genes of an individual subject to a small mutation factor, introducing further randomness into the population. This process continues until one of the possible termination criteria is met: if a known optimal or acceptable solution level is attained; or if a maximum number of generations have been performed; or if a given number of generations without fitness improvement occur, (Goldberg 1989; Lewin 1994, 1996).

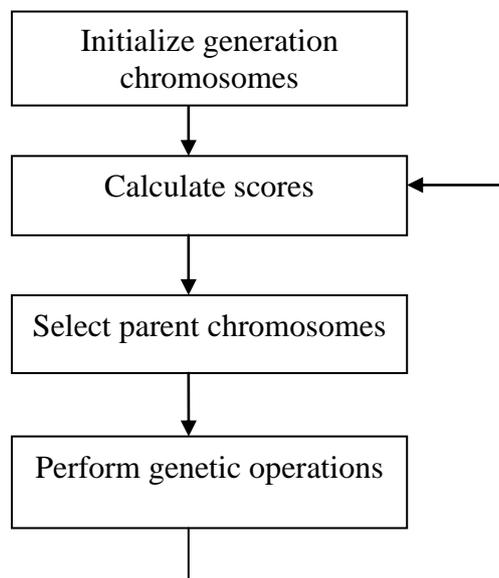


Fig 2. Genetic Algorithm Flowchart

G. FIR filter design with Genetic algorithm:

Consider a linear phase FIR filter, we want to find an N tap FIR filter that will approximate the frequency response of the ideal low pass filter as shown in figure . The transfer function of an FIR filter of order N is:

$$H(z) = \sum_{n=0}^N h(n)z^{-n} \quad (14)$$



International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 1, January 2014

The corresponding frequency response is given by:

$$H(e^{j\omega}) = \sum_{n=0}^N h(n) e^{-j\omega n} \quad (15)$$

In the case of linear phase, the transfer function will have symmetric coefficients that is $h(n)=h(N-n)$ and $H(e^{j\omega})=H(e^{j(\pi-\omega)})$ where

$$H(\omega) = h[N/2] + \sum_{n=1}^{N/2} (h[N/2-n][2\cos(n\omega)]) \quad (16)$$

Now we sample the frequency in $[0, \pi]$ with N points,

$$H_d(\omega) = [H_d(\omega_1), H_d(\omega_2), \dots, H_d(\omega_N)]^T \quad (17)$$

The error function (\mathcal{E}) now can be defined as

$$\mathcal{E}(\omega) = [H_d(\omega) - H(\omega)] \quad (18)$$

And the fitness function that has been used is the inverse of the absolute value of the error function (H):

$$\text{Fitness} = 1/|\mathcal{E}(\omega)| \quad (19)$$

Now the input to the GA will be as follow: Genetic Algorithm used to find the impulse response, and the matrix that will be [6]

$$x = [M, h[n-1/2], h[n-1/2+1], \dots, h[n-1]] \quad (20)$$

$M = N-1/2$

III.CONCLUSION

In order to get the coefficient accurate in less computational time an improved algorithm that is integer linear programming involves reducing of number of ripples with the filter specification within pass band and stop band. This algorithm also offers a fast execution. It also involves minimizing of non zero bits to get accuracy in the coefficient in particular filter specifications.

REFERENCES

- [1] Pearson P., "Annealing Based Optimization Methods for Signal Processing Applications", Blekinge Institute of Technology Dissertation Series 2003:01 ISSN 1650-2159.
- [2] Darrall Henderson, Sheldon H. Jacobson, Alan W. Johnson, "The Theory and Practice of Simulated Annealing".
- [3] Flávio C. A. Teixeira, "Optimum finite impulse response digital filter design using computational intelligence based optimization algorithms", IADIS International Conference Intelligent Systems and Agents 2007, ISBN: 978-972-8924-39-3.
- [4] Abhijit Chandra, Sudipta Chattopadhyay, "Novel Approach of Designing Multiplier-less Finite Impulse Response Filter using Differential Evolution Algorithm" I.J. Intelligent Systems and Applications, 2012, 4, 54-62 April 2012 in MECS.
- [5] Abhijit Chandra, Sudipta Chattopadhyay, "Role of Mutation Strategies of Differential Evolution Algorithm in Designing Hardware Efficient Multiplier-less Low-pass FIR Filter", JOURNAL OF MULTIMEDIA, VOL. 7, NO. 5, OCTOBER 2012
- [6] DE Ali Subhi Abbood, Raaed Faleh Hassan "Design of Finite Impulse Response Filters based on Genetic Algorithm". Diyala Journal of Engineering Sciences Vol. 06, No. 03, pp. 28-39, September 2013 ISSN 1999- 8716
- [7] Ajoy Kumar Dey, Susmita Saha, Avijit Saha, Shibani Ghosh "A Method of Genetic Algorithm (GA) for FIR Filter Construction: Design and Development with Newer Approaches in Neural Network Platform", (IJACSA), International Journal of Advanced Computer Science and Applications, Vol. 1, No. 6, December 2010
- [8] Juha Yli-Kaakinen, "Optimization of Digital Filters for Practical Implementations" Tampere University of Technology Publications 373 Tampere 2002