Received June 2004
Revised April 2005
Accepted April 2005

A parallel tabu search algorithm for digital filter design

Adem Kalinli
Department of Electronics, Vocational High School, Erciyes University, Kayseri, Turkey

Nurhan Karaboga
Department of Electronic Engineering, Faculty of Engineering, Erciyes University, Kayseri, Turkey

Abstract

Purpose – The purpose of the paper is to present a novel design method for the optimal finite word length (FWL) finite impulse response (FIR) filters.

Design/methodology/approach – The design method is based on a parallel tabu search (TS) algorithm which uses the crossover operator of the genetic algorithm.

Findings – Three design examples have been presented to show that the proposed method can provide a good solution to the design problem of a FWL FIR filter. In order to show the validity of the proposed method, the performance of the suggested method has been compared to those of widely-used other methods. From the comparison results, it was concluded that the proposed method can be efficiently used for the optimal FWL FIR filter design.

Research limitations/implications – The number of examples can be increased and also the performance of the proposed method might be compared to other design methods, apart from those presented in this work, developed for the design of optimal FWL FIR filters.

Practical implications – The use of this method produces optimal digital FWL FIR filters with low complexity and therefore provides advantages in the terms of speed and cost.

Originality/value – The originality is the application of the parallel TS algorithm described by the authors to the FWL FIR filter design. The work presented in the paper is particularly important for the researchers studying on the design methods for FWL FIR filter design and the applications of these type filters.

Keywords Optimization techniques, Signal processing

Paper type Research paper

1. Introduction

Any digital signal processing (DSP) algorithm or processor can be reasonably described as a digital filter. Digital filters may be divided into recursive and non-recursive categories depending on their use of feedback. The response of non-recursive or finite impulse-response (FIR) filters is dependent only upon present and previous values of the input signal. Recursive or infinite impulse-response (IIR) filters, however, depend not only upon the input data but also upon one or more previous output values. The main advantage of IIR filters is that they can provide a much better performance than FIR filters having the same number of coefficients. However, IIR filters might have a multi-modal error surface and a further problem with them is the possibility of the filter becoming unstable during the adaptation process. Therefore, FIR filters are usually desired for real-time applications.

COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering
Vol. 24 No. 4, 2005
pp. 1284-1298
© Emerald Group Publishing Limited
0332-1649
DOI 10.1108/03321640510615616

Generally, the implementation of a digital filter is carried out with hardware which has an important practical design constraint. The constraint is that each theoretical infinite precision coefficient used to describe the filter must be represented by a finite number of bits. Rounding or truncating each coefficient to a finite word length (FWL) coefficient produces a filter which may have a sub-optimal set of coefficients. Several computer techniques have been developed for determining an optimal set of FWL coefficients from the set of infinite precision coefficients (Kodek, 1980; Diethorn and Munson, 1986; Xu and Daley, 1992; Karaboga *et al.*, 1995; Radecki *et al.*, 1995). These techniques employ search methods using probabilistic rules to produce FWL coefficients. However, as the number of FWL coefficients used to describe the filter increases, the computation time required to obtain an acceptable value of error between the desired frequency response and the response of the digital filter designed by the set of FWL coefficients increases significantly. For example, with the use of the general purpose integer programming algorithm, the computation time increases exponentially by increasing the number of FWL coefficients used to define the filter. In the case of the methods based on genetic algorithms (GAs) (Holland, 1975), the promising regions of the search space can be found very quickly because of the parallel nature of the algorithm. However, because the same solutions might be evaluated several times due to the probabilistic transition rules employed, determining an optimal solution after quickly finding the good search regions might take quite a long computation time (Karaboga *et al.*, 1995).

The tabu search (TS) algorithm which is based on the general tenets of intelligent problem solving is a modern heuristic developed particularly for combinatorial optimisation problems. Although it has been proposed for combinatorial problems, it has also produced efficient solutions for numeric problems, too (Glover, 1986; Osman, 1991; Ni Guangzhen and Yan, 1998; Fanni *et al.*, 1999; Traferro and Uncini, 2000; Machado *et al.*, 2001). A basic version of the algorithm employs a neighbourhood search mechanism and uses deterministic transition rules to improve solutions. However, with the use of the strategies such as *forbidding* and *freeing* the algorithm can get out of a local minima and find the optimal solutions for the problem in a reasonable computational time (Glover, 1986; Karaboga and Kaplan, 1995). The purpose of this paper is to present a design method based on a parallel TS (PTS) algorithm for optimal FWL FIR filters. The paper is structured as follows. In Section 2, the basic and the proposed TS algorithms are introduced. Section 3 presents the definition of the problem. Simulation results and discussion are given in Section 4 to demonstrate the validity of the proposed method.

2. TS algorithms

In the following, the basic and the proposed TS algorithms are described.

2.1 Basic TS algorithm

TS is a general heuristic search procedure devised for finding a global minimum of a function $f(x)$. The problem of searching the optimum value of x which makes $f(x)$ minimum is called the optimisation problem of $f(x)$ and can be mathematically expressed as:

$$\text{Minimise } f(x)$$
$$\text{Subject to } x \in \mathbf{X}$$

where x represents the decision variables which are also called design parameters and the condition $x \in \boldsymbol{X}$ describes the constraints on the solution x. A step of the TS starts with a present solution x_{now}. $x_{\text{now}} \in \boldsymbol{X}$, has an associated set of feasible solutions $\mathbf{Q}$ which can be obtained by applying a simple modification to x_{now}. This modification is called a move. In order to be able to get rid of a local minima, a move to the neighbour $x*$ is created even if $\boldsymbol{x*}$ is worse than x_{now}. This would cause the cycling of the search. In order to avoid the cycling problem, a tabu list T is introduced. The tabu list stores all the tabu moves that cannot be applied to the present solution x_{now}. The moves stored in the tabu list are those carried out most frequently and recently according to some criteria called tabu restrictions. The use of tabu list decreases the possibility of cycling because it prevents returning in a certain number of iterations to a solution visited recently. After a subset of feasible solutions $\mathbf{Q}^*$ is produced according to the tabu list and evaluated for $f(x)$, the next solution is selected from them. The highest evaluation solution is selected as the next solution x_{next}. This loop is repeated until a stopping criteria is satisfied.

A tabu restriction is activated when the reverse of a move recorded in the tabu list occurs within a predetermined number of iterations or with a certain frequency over longer range of iterations. The former produces a *recency-based restriction* and the latter a *frequency-based restriction*. Tabu restrictions might sometimes prevent the search to find the solutions which have not been visited yet or even cause all available moves to be classified as the tabu. Therefore, it should be possible to forget the tabu constraints when a freedom is required for the search. A criterion called aspiration criteria is employed to determine which moves should be freed in such cases.

The flowchart of a basic TS is shown in Figure 1. In the initialisation unit, a random feasible solution $x_{\text{initial}} \in \mathbf{X}$ for the problem is generated, and the tabu list and other parameters are initialised. In the neighbour production unit, a feasible set of solutions is produced from the present solution according to the tabu list and aspiration criteria. The evaluation unit evaluates each solution $x*$ produced from the present one x_{now}. After the next solution x_{next} is determined by the selection unit, in the last unit the history record of the search is modified. If the next solution determined is better than the best solution found so-far x_{best}, the next solution is replaced with the present best solution.

In this work, a string of binary numbers is used to represent a possible solution and TS employs the following two constraints which are based on recency and frequency memories as the tabu constraints:

$$\text{recency}(x*) \geq \text{restriction\_period}$$

or

$$\text{frequency\_ratio}(x*) \leq \text{frequency\_limit}$$

The recency of a move is the difference between the current iteration count and the last iteration count at which that move was created. The frequency measure is the frequency ratio whose numerator represents the count of the number of occurrences of

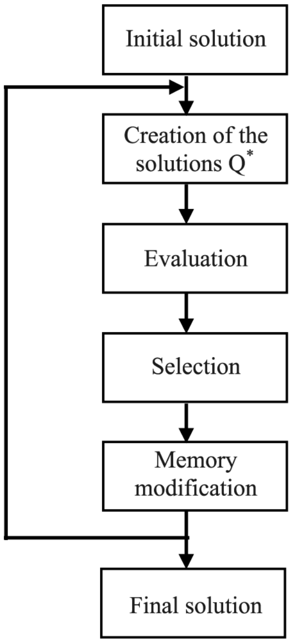


Figure 1.
Flowchart of a basic TS

a specific move and whose denominator represents the average numerator value over all possible moves.

In the algorithm used, the highest evaluation move is selected as the next solution. The aspiration by default is employed as the aspiration criteria. According to this criteria the least tabu solution is selected as the next solution. The least tabu solution is the solution which loses its tabu classification by the least increase in the value of the present iteration number.

2.2 Proposed PTS

There are mainly four sources of parallelism in TS algorithm (Crainic and Toulouse, 1997): Parallelism:

(1) in the cost function evaluation;

(2) in the neighbourhood examination;

(3) in the problem decomposition; and

(4) in the solution domain exploration by carrying out different searches.

In the last type parallelism, several independent TS algorithms are executed from different initial solutions. TS algorithms being executed in parallel have the ability of exchanging information. If the communication between the algorithms is carried out at the predetermined moments, then parallel algorithms are called synchronous. If this information exchange is realized at different moments, these type of algorithms are called asynchronous. The parallelism used in this work is of the last type.

The information exchange process between the basic TS algorithms executed in parallel is based on the crossover operation used in GAs.

2.3 Crossover operator used

The crossover operator is used to create two new solutions (children) from two existing solutions (parents) in the population formed by the selection operation. Depending on the representation way of the problem in the string form, a proper crossover operator must be chosen (Michalewicz, 1996; Goldberg, 1989). When the problem is represented in the binary string form, the simplest crossover operation can be applied as the following: Two solutions are randomly selected as parent solutions from the population and cut at a randomly selected point. The tails, which are the parts after the cutting point, are swapped and two new solutions are produced. A crossover operation can thus yield better solutions by combining the good features of parent solutions. An example of this simple crossover operator is given below:

PresentSolution1	**111100|001111**	→Tail
PresentSolution2	110001|111111	→Tail

NewSolution1	**111100|**111111
NewSolution2	110001**|001111**

2.4 Structure of the proposed model

The flowchart of the proposed model is depicted in Figure 2. Blocks from 1 to h are each identical to the flowchart shown in Figure 1. These blocks represent h TSs executing independently of one another. The initial solutions for TSs at the first level are created using random number generator with different seeds.

After a given number of iterations (*maxit1*) the execution of h TSs is stopped. *maxit1* is normally chosen to be sufficiently large to allow the search to complete the local searching. The h solutions found by these TSs at the first level represent the "preliminary" solutions. An operation is applied to the solutions found by h PTSs at the first level to create the initial solutions for TSs at the second level. Various methods could be used for this purpose; for example, the best solution found by the TSs at the first level can be always selected as one of the initial solutions for the next level. The others can be produced from the preliminary solutions by using a suitable crossover operator or all the initial solutions can be created using crossover operator. Another procedure to form initial solutions could be the following: the best preliminary solution is selected directly, some can be created by applying the crossover operator to the preliminary solutions and the rest can be generated by a random number generator. In this work, the best preliminary solution is directly selected and the rest of initial solutions of TSs are created by applying the crossover operation to the solutions obtained at the previous level.

At all levels, TSs are executed in the same way. The best solution found through the whole search process is taken as the optimal solution for the problem. Each TS might have different control parameter values.

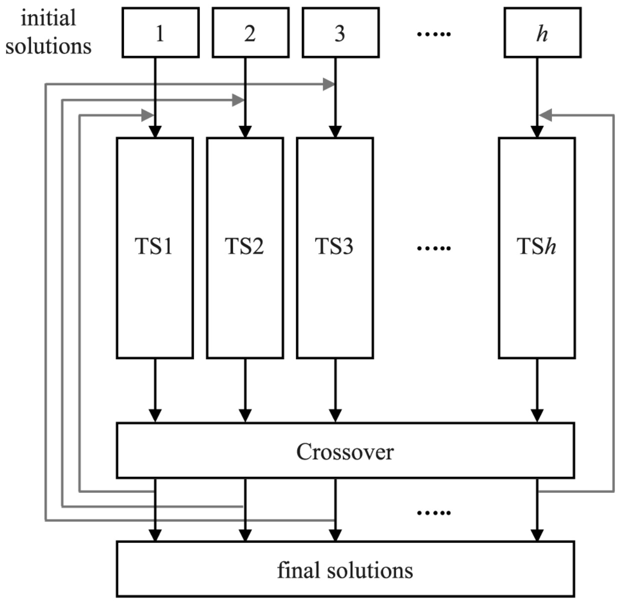


Figure 2. Flowchart of the proposed TS

3. Description of the problem

The normalised transfer function of a linear phase, symmetric N-order FIR filter with finite precision coefficients is defined as the following:

$$H(\mathrm{f}) = \left\{ \begin{array}{ll} a(0) + 2\sum_{n=1}^{\mathrm{M}} A(n)\cos(2\pi \mathrm{fn});\ \ \mathrm{M} = \dfrac{\mathrm{N}-1}{2} & \text{if N odd} \\ 2\sum_{n=1}^{\mathrm{M}} A(n)\cos(2\pi \mathrm{f(n}-1)/2);\ \ \mathrm{M} = \dfrac{\mathrm{N}}{2} & \text{if N even} \end{array} \right\} \quad (1)$$

where $A(n)$ are the impulse response coefficients of the filter and M is their number.

It is worth noting that the values of $a(n)$ are within the interval $[-1, 1]$, and they are not uniformly distributed. When minimising the amount of hardware is a concern, fixed-point representation of coefficients is usually used. In fixed-point representation a binary word can be interpreted as an unsigned or signed fixed-point rational. Since a signed fixed-point rational is a number in the form $A(n)/2^{\mathrm{b}}$, where $A(n)$ and b are integers, $-2^{L-1} \leq A(n) \leq 2^{L-1} - 1$, and L is the word length used for the coefficients, the estimate $a'(n)$ of the coefficient $A(n)$ by choosing a value for b and determining $A(n)$ is found as $a'(n) = A(n)/2^{b}$, where $A(n) = \mathrm{round}(A(n) * 2b)$. A value for b that will not overflow the largest magnitude coefficient. Hence, this maximum value for b is computed as $b = \lfloor \log_2((2^{L-1} - 1)/\max(|A(n)|)) \rfloor$, where $\lfloor * \rfloor$ denotes the greatest integer less than or equal to $*$. In general, $a'(n)$ is only an estimate of $a(n)$ because of the rounding operation. This approximation is referred to as coefficient quantization process, which produces discrete values. For these reasons, a discrete optimisation algorithm seems to be more suitable to approach this problem (Fanni *et al.*, 1998).

The problem of optimal design of digital filters can be formulated as a min-max optimisation problem in which the objective function is the maximum weighted ripple in a finite number N_{f} of normalised frequencies at the limits (sampling frequency equal to 1), and inside the stop-band and the pass-band regions. In other words, the filter coefficients are to be determined in such a way that the magnitude of the maximum deviation between the desired and the actually realised amplitude responses is minimised (Schlichthärle, 2000). Let $W(f)$ be a weighting function, $H_{\mathrm{d}}(f)$ the desired value of the frequency response, and $H(f)$ the actual value of the frequency response of the filter, the optimisation problem may be stated as the minimisation of $e(\mathbf{a})$ defined as follows:

$$e(\mathbf{a}) = \max_{k=1,\ldots,N_{\mathrm{f}}} \{ |\mathbf{W}(f_{\mathrm{k}})[\mathbf{H}_{\mathrm{d}}(f_{\mathrm{k}}) - \mathbf{H}(f_{\mathrm{k}})]| \} \tag{2}$$

4. Application of TS algorithm to the problem and the simulation results

4.1 Application of TS algorithm to the problem

In this work, the parallel version of TS algorithm described in Section 2.4 is used to design three filters. Several basic TS algorithms with different recency and frequency factors start the search with the same initial solution. Since the fixed-point representation is used for the coefficients in this work, the initial solution is obtained by calculating the integers $A(n)$ from the set of infinite precision coefficients $A(n)$ found by using Parks-Mc Clellan algorithm and rounding them. After a certain number of iterations (*maxit1*), the solutions found by each basic TS are collected in a pool and the crossover operator is applied to them to produce new solutions. These new solutions are taken as the initial solutions for the basic TS algorithms at the next level. This process is repeated for a predetermined number of levels (m). In this work, $maxit1 = 400$ and $m = 7$, and the number of TS algorithms running in parallel is 5 ($h = 5$).

In each iteration of TS, the neighbours of the present solution which are not in the tabu list are created and evaluated to select the next solution. A neighbour is produced by adding an integer between -2 and 2 to the $A(n)$ value of the coefficient $a'(n)$, which was calculated depending on the predetermined value for L. After production of a neighbour, the filter associated with that neighbour is constructed and its response is determined. By means of the desired response and the determined actual response, $e(\mathbf{a})$ is computed from equation (2). The performance of a neighbour is calculated using the following equation:

$$P_i = \frac{1}{e(\mathbf{a})^2 + w} \tag{3}$$

where P_i is the performance value of the neighbour i and w is a small positive constant which is used to avoid overflow. The next solution is the best admissible solution among the neighbours and it is determined by means of the evaluation values of them. The evaluation value of the neighbour i is calculated as

$$e_i = xr_i - yf_i + zP_i \tag{4}$$

where r_i, f_i and P_i are the recency, frequency and performance values of ith neighbour, respectively. In equation (4), x, y and z represent the recency, frequency and performance weighting factors, respectively, and in this work, $x = 2$, $y = 1$ and $z = 0.1$.

The neighbour which has the highest evaluation value is selected as the next solution. After the next solution is determined, the tabu list, recency and frequency memory are modified. For other iterations, the same procedure is repeated. If all neighbours are tabu, then the neighbour for which the recency value is the maximum is chosen as the next solution. In order to compare the performance of both TS algorithms, the total iteration number for the algorithms was taken to be equal to each other ($totaliterationnumber = h \times maxit1 \times m = 5 \times 400 \times 7 = 14,000$).

4.2 Simulation results

In the simulation study, three different filters were considered: low-pass, high-pass and band-pass.

4.2.1 Example 1: low-pass filter. The first example is a low-pass digital FIR filter which has the following desired frequency response:

$$H_{\mathrm{d}}(f) = \begin{cases} 1 & f \in [0, 0.2] \\ 0 & f \in [0.25, 0.5] \end{cases} \tag{5}$$

The word length for this design is 10 bits and filter length is 40. A comparison of the optimal coefficients produced by the PTS algorithm to those obtained by the basic TS algorithm, parallel GA (PGA), integer programming and round-off methods are shown in Table I. The performance characteristics of the filter designed by these techniques are summarised in Table II.

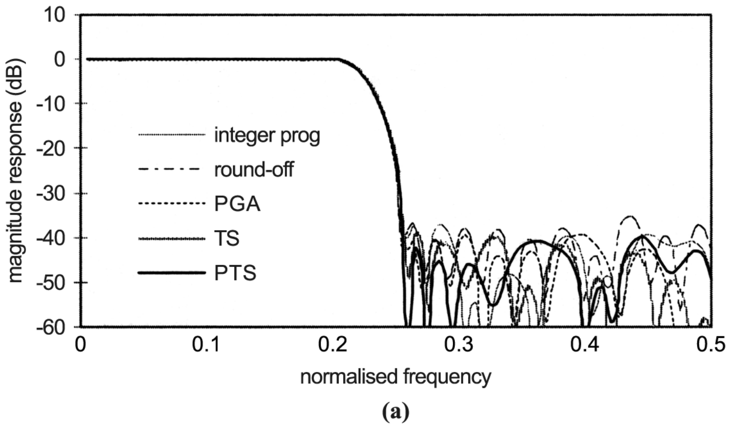
Coefficients	Integer programming (Xu and Daley, 1992)	Round-off	PGA (Xu and Daley, 1992)	TS	PTS
$A(0) = A(39)$	2	1	2	1	2
$A(1) = A(38)$	2	4	3	2	2
$A(2) = A(37)$	−1	−2	−1	−2	−1
$A(3) = A(36)$	−4	−4	−4	−4	−4
$A(4) = A(35)$	0	0	1	0	0
$A(5) = A(34)$	6	6	6	5	5
$A(6) = A(33)$	1	2	1	2	1
$A(7) = A(32)$	−8	−7	−7	−7	−7
$A(8) = A(31)$	−5	−5	−5	−5	−5
$A(9) = A(30)$	8	8	9	8	8
$A(10) = A(29)$	10	10	10	10	10
$A(11) = A(28)$	−9	−8	−8	−8	−8
$A(12) = A(27)$	−17	−17	−16	−17	−16
$A(13) = A(26)$	5	5	5	5	5
$A(14) = A(25)$	27	27	27	27	27
$A(15) = A(24)$	2	3	2	3	3
$A(16) = A(23)$	−43	−44	−44	−43	−43
$A(17) = A(22)$	−25	−24	−24	−24	−25
$A(18) = A(21)$	92	92	92	92	93
$A(19) = A(20)$	211	212	211	213	212

Table I. Filter coefficients obtained by using five different design methods

In this application, the design with the PTS algorithm produces the least error in the stop- and pass-band. This method exceeds the performance criteria of those obtained with the integer programming, round-off method, PGA and basic TS algorithms. The frequency response characteristics for each of these filter designs are also shown in Figure 3. All filter characteristics are very similar to each other in the pass-band, however, in the stop-band the filter designed by the proposed TS algorithm produces

Table II.
The performance characteristics of the FIR filter designed by five methods

Methods	Pass-band deviation	Stop-band deviation
Round-off	0.020087	0.01747 (−35.15 dB)
Integer programming	0.016144	0.01380 (−37.20 dB)
TS	0.014606	0.01136 (−38.89 dB)
PGA	0.016470	0.01105 (−39.13 dB)
PTS	0.014304	0.01011 (−39.90 dB)



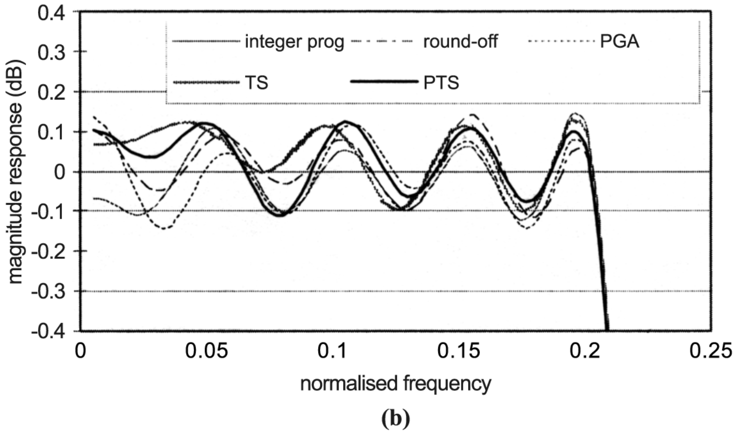


Figure 3.
Comparison of frequency response characteristics of the filters designed by five different methods

Notes: (a) overall response (b) passband region the best performance and the greatest attenuation. It is seen that both TS algorithms produce better responses in the pass-band region. The deviation of the response produced by the proposed TS in this region is 0.014304 while that obtained by the basic TS is 0.014606. In the stop-band region, PGA and PTS produce the better responses. In this case, the proposed method performs better than PGA by ~ 1 dB.

4.2.2 Example 2: high-pass filter. The desired frequency response of the second filter which is a high-pass filter is defined as

$$H_d(f) = \begin{cases} 0 & f \in [0, 0.42] \\ 1 & f \in [0.455, 0.5] \end{cases} \tag{6}$$

The word length for this filter is 8 bits and filter length 51. A comparison of the optimal coefficients produced by the PTS algorithm, basic TS algorithm, GA and round-off methods is presented in Table III. The performance characteristics of the filter designed by these techniques are summarised in Table IV.

In this example, the filter designed by PTS algorithm produces the least error in both the stop- and pass-band regions. The proposed method exceeds the performance criteria of other three methods. The frequency responses of the filters designed by the methods are also shown in Figure 4. From the responses it is seen that both TS algorithms produce better responses in both regions than round-off and GA methods. Moreover, the filter designed by PTS has slightly better frequency response.

Coefficients	Round-off	GA	TS	PTS
$A(0) = A(50)$	−9	−8	−7	−8
$A(1) = A(49)$	8	8	7	7
$A(2) = A(48)$	−10	−10	−8	−9
$A(3) = A(47)$	11	11	10	10
$A(4) = A(46)$	−11	−11	−10	−11
$A(5) = A(45)$	10	11	9	9
$A(6) = A(44)$	−8	−8	−6	−7
$A(7) = A(43)$	4	4	3	3
$A(8) = A(42)$	2	2	4	3
$A(9) = A(41)$	−9	−8	−9	−9
$A(10) = A(40)$	16	15	16	16
$A(11) = A(39)$	−22	−22	−22	−22
$A(12) = A(38)$	27	26	27	26
$A(13) = A(37)$	−29	−28	−27	−28
$A(14) = A(36)$	27	27	26	26
$A(15) = A(35)$	−21	−21	−19	−20
$A(16) = A(34)$	11	11	10	10
$A(17) = A(33)$	3	3	4	5
$A(18) = A(32)$	−21	−21	−22	−22
$A(19) = A(31)$	41	41	42	41
$A(20) = A(30)$	−62	−62	−62	−63
$A(21) = A(29)$	83	83	83	83
$A(22) = A(28)$	−101	−100	−99	−101
$A(23) = A(27)$	115	115	114	115
$A(24) = A(26)$	−124	−124	−123	−124
$A(25)$	127	127	126	127

Table III. Filter coefficients obtained by using four different design methods

4.2.3 Example 3: band-pass filter. The desired frequency response of the band-pass FIR filter is defined as

$$H_d(f) = \begin{cases} 0 & f \in [0, 0.14] \\ 1 & f \in [0.18, 0.33] \\ 0 & f \in [0.375, 0.5] \end{cases} \tag{7}$$

Table IV.
The performance characteristics of the FIR filter designed by four methods

Methods	Pass-band deviation	Stop-band deviation
Round-off	0.1152602	0.015716 (−36.07 dB)
GA	0.1144948	0.011510 (−38.78 dB)
TS	0.1040928	0.011471 (−38.81 dB)
PTS	0.0994023	0.011322 (−38.92 dB)

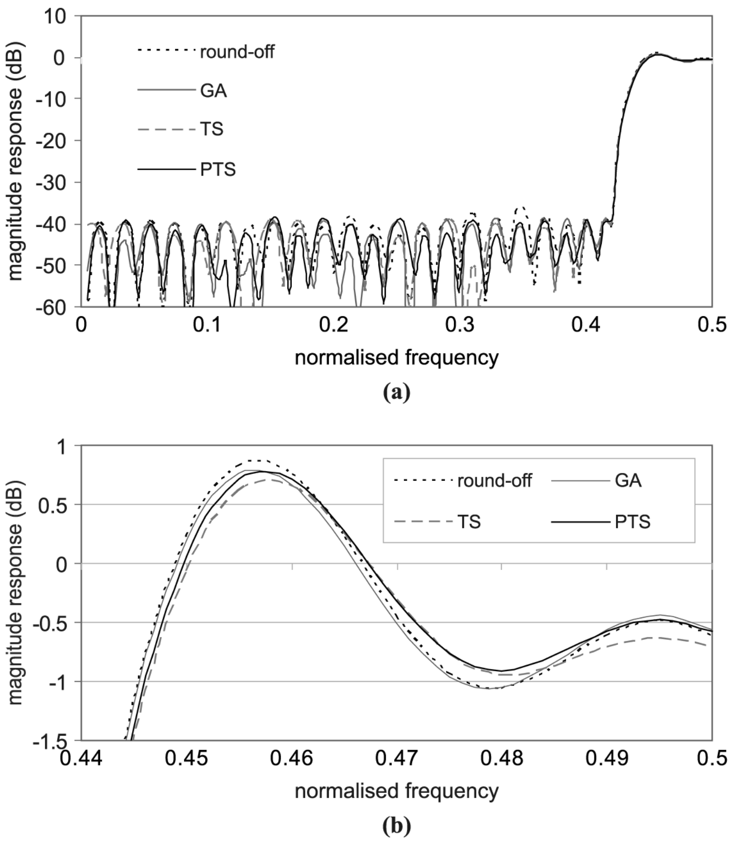


Figure 4.
Comparison of frequency response characteristics of the high-pass filters designed by four different methods

Notes: (a) overall response, (b) passband region

The word length for this filter is 8 bits and filter length 42. A comparison of the coefficients produced by the algorithms is presented in Table V. The performance characteristics of the band-pass filters designed by the algorithms are given in Table VI.

In this application, the design with the PTS algorithm produces the least error in the stop- and pass-band regions. The frequency response characteristics of the filters are also shown in Figure 5. It is clearly seen that the proposed TS algorithm produces the best response. The deviation of the response in the pass-band is 0.037382. In the stop-band regions the proposed method performs better than TS and GA by ~ 1 dB and ~ 2 dB, respectively.

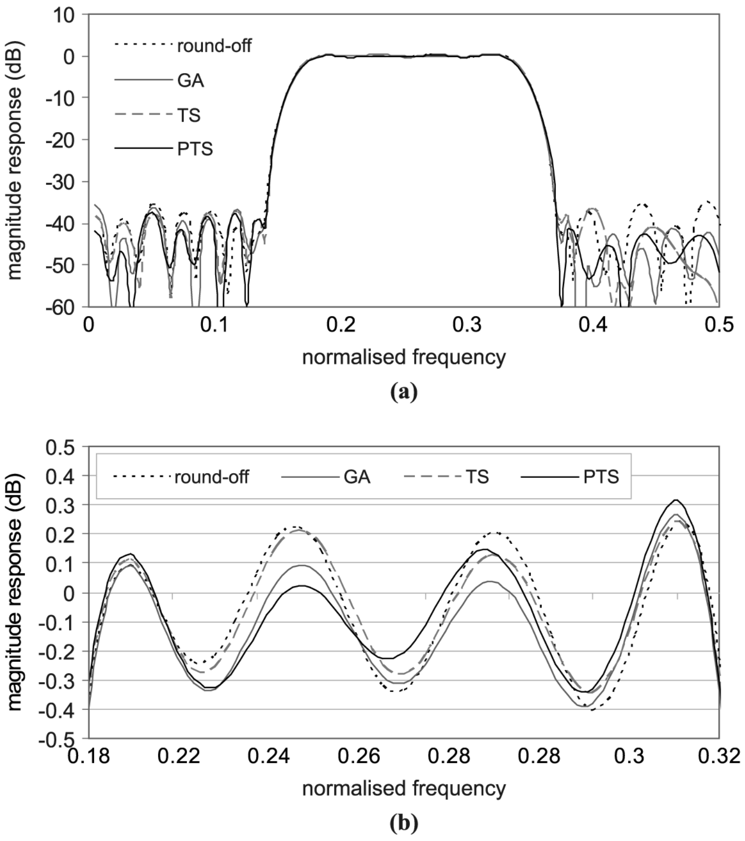
It is clearly seen from the responses shown in Figures 3-5; and the results presented in Tables II, IV and VI that the proposed method can design filters which has better

Coefficients	Round-off	GA	TS	PTS
$A(0) = A(41)$	−4	−4	−4	−4
$A(1) = A(40)$	−6	−5	−5	−4
$A(2) = A(39)$	7	6	6	6
$A(3) = A(38)$	4	4	5	4
$A(4) = A(37)$	−2	−1	−2	−1
$A(5) = A(36)$	3	3	3	3
$A(6) = A(35)$	−7	−8	−8	−8
$A(7) = A(34)$	−11	−11	−11	−11
$A(8) = A(33)$	11	11	11	11
$A(9) = A(32)$	6	6	6	7
$A(10) = A(31)$	2	2	2	2
$A(11) = A(30)$	13	13	13	12
$A(12) = A(29)$	−21	−21	−21	−21
$A(13) = A(28)$	−25	−25	−25	−25
$A(14) = A(27)$	19	19	19	20
$A(15) = A(26)$	3	3	3	3
$A(16) = A(25)$	23	22	23	22
$A(17) = A(24)$	54	54	54	54
$A(18) = A(23)$	−86	−86	−86	−86
$A(19) = A(22)$	−113	−112	−113	−113
$A(20) = A(21)$	127	126	127	127

Table V. Filter coefficients obtained by using five different design methods

Methods	Pass-band deviation	Stop-band deviation
Round-off	0.045802	0.017925 (−34.93 dB)
GA	0.044443	0.016247 (−35.78 dB)
TS	0.039049	0.014667 (−36.67 dB)
PTS	0.037382	0.013314 (−37.51 dB)

Table VI. The performance characteristics of the FIR filter designed by four methods



Notes: (a) overall response, (b) passband region

Figure 5.
Comparison of frequency response characteristics of the band-pass filters designed by four different methods

pass- and stop-band region responses when compared to other methods. Finally, it was concluded that the proposed method can be efficiently used for digital FIR filter design.

5. Conclusion

A new design method for FIR linear phase filters has been proposed. The method is based on a PTS algorithm, which uses the crossover operator of GA. Three design examples have been presented to show that the proposed procedure can provide good solutions to the design of FIR filters. In order to show the validity of the proposed method, the performance of the proposed method has been compared to those of widely-used other methods.

References

Crainic, T.G. and Toulouse, M. (1997), “Parallel metaheuristics”, Research Report, Centre de Recherche sur les Transports Universite de Montreal, France.

Diethorn, E.J. and Munson, D.C. (1986), "Finite wordlength FIR digital filter design using simulated annealing", *Proc. Int. Symp. Circuits and Sys.*, pp. 217-20.

Fanni, A., Marchesi, M., Pilo, F. and Serri, A. (1998), "Tabu search metaheuristic for designing digital filters", *COMPEL The Institutional Journal for Computation and Mathematics in Electrical and Electronic Engineering*, Vol. 17 Nos 5/6, pp. 789-96.

Fanni, A., Manunza, A., Marchesi, M. and Pilo, F. (1999), "Tabu search metaheuristics for electromagnetic problems optimisation in continuous domain", *IEEE Transactions on Magnetics*, Vol. 34 No. 3, pp. 1694-7.

Glover, F. (1986), "Future paths for integer programming and links to artificial intelligence", *Computers and Operations Research*, Vol. 5, pp. 533-49.

Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimisation, and Machine Learning*, Addison-Wesley, Reading, Ma.

Holland, J.H. (1975), *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI.

Karaboga, D. and Kaplan, A. (1995), "Optimising multivariable functions using tabu search algorithms", paper presented at the Tenth Int. Symp. on Computer and Information Sciences, Turkey Vol. II, pp. 793-9.

Karaboga, N., Horrocks, D.H., Karaboga, D. and Alci, M. (1995), "Design of FIR filters using genetic algorithms", paper presented at the European Conference on Circuits Theory and Design (ECCTD'95), Istanbul, Turkey Vol. 2, pp. 553-6.

Kodek, D.M. (1980), "Design of optimal finite wordlength FIR filter using integer programming techniques", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. ASSP-28, pp. 304-8.

Machado, J.M., Shiyou, Y., Ho, S.L. and Peihong, N. (2001), "A common tabu search algorithm for the global optimisation of engineering problems", *Computer Methods in Applied Mechanics and Engineering*, Vol. 190, pp. 3501-10.

Michalewicz, Z. (1996), *Genetic Algorithms + Data Structures = Evolution Programs*, Springer-Verlag, New York, NY.

Ni Guangzhen, Y.S. and Yan, L. (1998), "An universal tabu search algorithm for global optimization of multimodal functions with continuous variables in electromagnetics", *IEEE Transactions on Magnetics*, Vol. 34 No. 5, pp. 2901-4.

Osman, I.O. (1991), "Metastrategy simulated annealing and tabu search algorithms for combinatorial optimisation problems", PhD thesis, Imperial College, University of London,.

Radecki, J., Konrad, J. and Dubiois, E. (1995), "Design of multidimensional finite-wordlength FIR and IIR filters by simulated annealing", *IEEE Transactions Circuits and Systems-II: Analog and Digital Signal Processing*, Vol. 42 No. 6, pp. 424-31.

Schlichthärle, D. (2000), *Digital Filters, Basics and Design*, Springer, Germany.

Traferro, S. and Uncini, A. (2000), "Power-of-two adaptive filters using tabu search", *IEEE Transactions on Circuits and Systems-II: Analog and digital Signal Processing*, Vol. 47 No. 6, pp. 566-9.

Xu, D.J. and Daley, M.L. (1992), "Design of finite word length FIR digital filter using a parallel genetic algorithm", paper presented at the IEEE Southeast Conference, Birmingham, USA Vol. 2, pp. 834-7.

(Adem Kalinli received the BS, MS and PhD degrees in Electronics Engineering from Erciyes University, Turkey, in 1990, 1993 and 1996, respectively. He is now an assistant professor of Industrial Electronics Division of Vocational High School at the Erciyes University of Turkey. He also holds various positions at Erciyes University. His current research interest include intelligent optimisation techniques, neural networks and their engineering applications, particularly electronics engineering problems.

Nurhan Karaboga was born in Osmaniye, Turkey, in December 1965. She received the BSc, MSc and PhD degrees in Electronic Engineering from Erciyes University, Kayseri, Turkey, in 1983, 1990 and 1995, respectively. She was research assistant from 1987 to 1995 in Electronic Engineering Department at Erciyes University where she worked on digital signal processing. From 1992 to 1994 she also worked as academic visitor in the University of Wales College of Cardiff, England. Since 1995 she has been an assistant professor in the Department of Electronic Engineering at the University of Erciyes, Turkey. Her current research interests include Genetic Algorithms, Ant Colony Algorithms, Simulated Annealing Algorithm, digital filter design and data communication.)