Elektrotehniški vestnik 69(5): 266–272, 2002
Electrotechnical Review, Ljubljana, Slovenija

An Approximation Error Lower Bound for Integer Polynomial Minimax Approximation

Dušan M. Kodek

University of Ljubljana, Faculty of Computer and Information Science, Tržaška 25, 1000 Ljubljana, Slovenia
E-mail: duke@fri.uni-lj.si

Abstract. The need to solve a polynomial minimax approximation problem appears often in science. It is especially common in signal processing and in particular in filter design. The results presented in this paper originated from a study of the finite wordlength restriction in the FIR digital filter design problem. They are, however, much more general and can be applied to any polynomial minimax approximation problem in which the polynomial coefficients are constrained to a finite set of numbers. The finite set restriction introduces a nonzero lower bound to the approximation error. For any given non-trivial function that is to be approximated there is a nonzero lower bound below which it is not possible to go, no matter how large the polynomial degree n. For practical purposes it is very useful to know this lower bound because it can be used to substantially increase the speed of the branch-and-bound algorithm that gives the optimal integer coefficients. A method for computing such a bound is presented in the paper.

Key words: combinatorial optimization, integer programming, minimax approximation, digital filter design

Spodnja meja aproksimacijske napake pri celoštevilski polinomski minimax aproksimaciji

Povzetek. Potreba po rešitvi polinomskega minimax aproksimacijskega prooblema se pogosto pojavlja v znanosti. Posebno običajna je v procesiranju signalov in še posebej pri načrtovanju filtrov. V tem članku opisani rezultati so nastali iz študija vpliva omejitve dolžine besede na načrtovanju KEO digitalnih filtrov. Vendar so rezultati bistveno splošnejši in jih je mogoče uporabiti pri vsaki polinomski minimax aproksimaciji, pri kateri so koeficienti omejeni na končno množico števil. Omejitev na končno množico ima za posledico pojav neničelne spodnje meje aproksimacijske napake. Za vsako netrivialno funkcijo, ki jo želimo aproksimirati, obstaja spodnja meja aproksimacijske napake pod katero ni mogoče priti ne glede na to kako velika je stopnja polinoma n. Za praktične namene je zelo uporabno poznati to spodnjo mejo, ker omogoča bistveno povečanje hitrosti branch-and-bound algoritma, ki daje optimalne celoštevilske koeficente. V članku je podana metoda za izračun te meje.

Ključne besede: kombinatorična optimizacija, celoštevilsko programiranje, minimaks aproksimacija, načrtovanje digitalnih filtrov

1 Introduction

The unconstrained polynomial coefficients which are easily obtained by some standard approximation algorithm, are often called “infinite precision” coefficients. They are typically 32-bit floating point numbers and are of course hardly of infinite precision. But the 32-bit wordlength is much longer than 8 or 10-bit wordlengths that we would like to use in practical applications. One may, for example, wish to use a fixed point DSP processor which is cheaper and/or faster than a floating point one. The number of bits b that can be used to represent the coefficients (of an FIR digital filter, for example) will in general depend on the polynomial degree n, processor properties, and on signal quantization. It is almost always desirable that the coefficients are represented with a number of bits b that is as small as possible.

Note that a digital filter is used here as an example only (albeit an important one). The above reasoning is much more general and holds for any device that is based on the solution of the minimax approximation problem. The difficulty is that solving this problem, with the coefficients constrained to b-bit integers, is very hard. The problem is *NP*-complete and all known algorithms are exponential. They work only if the polynomial degree n is relatively small. It was shown in [1] that it is possible to solve the integer minimax approximation problem using the general purpose integer programming techniques. The problem with this approach is that it is slow and may not give the result in a reasonable time. A better approach is to use an algorithm that is tailored to the specifics of the minimax approximation. Such an algorithm requires solutions of a large number of suitably redefined unconstrained minimax approximation problems. It is crucial for the success of the algorithm to have a technique that

Received 25 Maj 2002
Accepted 6 November 2002

minimizes the number of these subproblems. A lower bound for the increase of minimax approximation error is such a technique and is presented in this paper.

2 Formulation of the problem

Let us start with the usual unconstrained polynomial minimax approximation problem in which the polynomial coefficients a_j can be any real numbers. The class of generalized polynomials $p(x)$ of degree n is defined as

$$p(x) = \sum_{j=0}^{n} a_j \Phi_j(x), \tag{1}$$

where $\Phi_j(x),\ j = 0, 1, \cdots, n$, is a polynomial basis. Any basis can be used as long as it satisfies the Haar condition (see [3] for the description of the Haar condition). Note that this definition includes the usual polynomials ($\Phi_j(x) = x^j,\ j = 0, 1, \cdots, n$) and cosine polynomials ($\Phi_j(x) = \cos_j x,\ j = 0, 1, \cdots, n$) as a special case. The cosine polynomials are used in the FIR digital filter design case, although this is not important here. The minimax approximation problem is defined as the search for the polynomial $p(x)$ that that minimizes the expression

$$\|D - p\|_\infty = \max_{a \le x \le b} |W(x)(D(x) - p(x))|. \tag{2}$$

$D(x)$ is the real function that is to be approximated, the weighting function $W(x)$ is by definition real and positive, and the interval $[a, b]$ is a subset of the real line.

Let $p^*(x)$ be the optimal approximation to $D(x)$

$$\begin{aligned} p^*(x) &= \sum_{j=0}^{n} a_j^* \Phi_j(x), \\ \|D - p^*\|_\infty &\le \|D - p\|_\infty, \quad \forall p(x). \end{aligned} \tag{3}$$

Several algorithms, from linear programming to various versions of the *exchange algorithm*, can be used to find $p^*(x)$ [3]. Finding the $p^*(x)$ is considered an easy problem. The main reason for this is the following well-known property of the optimal minimax approximation: There are exactly $n + 2$ so called extremal points in $[a, b]$ at which the approximation error achieves its maximum. Let $x_i,\ i = 0, 1, \cdots, n+1$, be these extremal points. The following equations hold

$$W(x_i)(D(x_i) - \sum_{j=0}^{n} a_j^* \Phi_j(x_i)) = (-1)^i h^*, \tag{4}$$

for $i = 0, 1, \cdots, n + 1$, and $|h^*|$ is the optimal approximation error.

Things change dramatically when coefficients a_j are constrained to values from a finite set of numbers. We can, without loss of generality, make this set equal to a set of b-bit integers $\{-2^{b-1}, \cdots, -1, 0, 1, \cdots, 2^{b-1}\}$. Note that the integers are chosen for convenience only; any other finite set of real numbers can be used instead. This will in most practical cases also require multiplication of $D(x)$ and division of $W(x)$ by a suitable scaling factor S. Selection of the scaling factor S is not trivial; it will, however, be ignored in this paper because we wish to concentrate on the lower bound derivation. In other words, $D(x)$ and $W(x)$ are left unchanged, and the approximating polynomial $p(x)$ is from here on defined as

$$p(x) = \sum_{j=0}^{n} a_j \Phi_j(x), \tag{5}$$

where $a_j \in \{-2^{b-1}, \cdots, -1, 0, 1, \cdots, 2^{b-1}\}$. The problem of finding the optimal integer polynomial $p(x)$ is much more difficult than the unconstrained case, although it may not appear so at first.

The problem we wish to solve can be stated like this: What is the lower bound on the increase of approximation error that is caused by the b-bit integer constraint? Let us denote this lower bound as δ and define it formally as

$$\delta \ge \min_{b\text{-bit } p(x)} (\|D - p\|_\infty - |h^*|). \tag{6}$$

Note that δ is defined over all b-bit integer polynomials of degree n, not just one particular $p(x)$.

3 Lower bound theorem

Let us investigate the case of a particular polynomial $p(x)$ first. Assume that its coefficients a_j are known and that they are different from a_j^*. We wish to compute δ for this $p(x)$. A special property of all functions that satisfy the Haar condition is useful here [3]. It says that there always exist multipliers $\sigma_i,\ i = 0, 1, \cdots, n + 1$, not all zero, that satisfy the conditions

$$\sum_{i=0}^{n+1} \sigma_i \Phi_j(x_i) = 0, \quad j = 0, 1, \cdots, n, \tag{7}$$

for any $(n + 2)$ points x_i from the interval $[a, b]$. It is easy to see that equations (1) and (7) imply

$$\sum_{i=0}^{n+1} \sigma_i p(x_i) = 0, \tag{8}$$

for any $p(x)$. The numbers $\sigma_i,\ i = 0, 1, \cdots, n + 1$, have a very important property. All are nonzero and their signs alternate. That is

$$\operatorname{sign}(\sigma_{i+1}) = -\operatorname{sign}(\sigma_i), \quad i = 0, 1, \cdots, n. \tag{9}$$

The numbers σ_i are needed to prove the following theorem for the lower bound δ when a $p(x)$ is known:

Theorem 1 *Let $p^*(x)$ be the optimal weighted minimax approximation to a real function $D(x)$ on the interval $[a, b]$ and let $p(x)$ be any other polynomial. Then the increase in approximation error δ is bounded by*

$$\delta \ge \max_{0 \le i \le n+1} |c_i W(x_i)(p^*(x_i) - p(x_i))|, \tag{10}$$

where x_i are extremal points corresponding to $p^*(x)$ and multipliers c_i, $i = 0, 1, \cdots, n+1$, are defined as

$$c_i = \begin{cases} \dfrac{|\dfrac{\sigma_i}{W(x_i)}|}{\displaystyle\sum_{\substack{k=0\\k\neq i}}^{n+1} |\dfrac{\sigma_k}{W(x_k)}|} & \text{if } (-1)^i h^*(p^*(x_i) - p(x_i)) < 0 \\ 1 & \text{if } (-1)^i h^*(p^*(x_i) - p(x_i)) \geq 0. \end{cases} \tag{11}$$

Proof: The approximation error $e(x)$ of a polynomial $p(x)$ is equal to

$$e(x) = W(x)(D(x) - p(x)). \tag{12}$$

By subtracting (4) it can be rewritten as

$$\begin{aligned} e(x_i) = (-1)^i h^* + W(x_i)(p^*(x_i) - p(x_i)),& \\ i = 0, 1, \cdots, n+1,& \end{aligned} \tag{13}$$

where x_i are extremal points. The error $e(x_i)$ depends on the sign of $p^*(x_i) - p(x_i)$ relative to the sign of $(-1)^i h^*$ ($W(x)$ is by definition positive). Let us divide the set of extremal points $\{x_i; i = 0, 1, \cdots, n+1\}$ into two subsets. The subset Z_M contains points x_i for which $(-1)^i h^*(p^*(x_i) - p(x_i)) < 0$. The rest of the points x_i form the subset Z_P.

Let us examine the points in Z_P first. Both terms of (13) have the same sign and the error $|e(x_i)|$ is simply equal to

$$|e(x_i)| = |h^*| + |W(x_i)(p^*(x_i) - p(x_i))|, \quad x_i \in Z_P. \tag{14}$$

This corresponds to (10) and (11) with $c_i = 1$ and proves the theorem for the points in Z_P.

Things are more complicated for the points in Z_M. We start by rewriting (13) as

$$\frac{e(x_i)}{W(x_i)} = \frac{(-1)^i h^*}{W(x_i)} + p^*(x_i) - p(x_i). \tag{15}$$

By multiplying each of the equations (15) with the corresponding multiplier σ_i defined in (7) and adding them together we get

$$\sum_{i=0}^{n+1} \frac{\sigma_i}{W(x_i)} e(x_i) = \sum_{i=0}^{n+1} (-1)^i h^* \frac{\sigma_i}{W(x_i)}, \tag{16}$$

where (8) was used to eliminate $p^*(x_i) - p(x_i)$. Let x_r be any point from Z_M. Eq. (8) can be rewritten as

$$\begin{aligned} \sum_{\substack{i=0\\i\neq r}}^{n+1} \frac{\sigma_i}{W(x_i)} e(x_i) &= \sum_{\substack{i=0\\i\neq r}}^{n+1} (-1)^i h^* \frac{\sigma_i}{W(x_i)} - \\ &\quad - \sigma_r(p^*(x_r) - p(x_r)). \end{aligned} \tag{17}$$

It follows from (9) that all the terms $(-1)^i h^* \sigma_i, i = 0, 1, \cdots, n+1$, have equal sign. Since the signs of $(-1)^r h^*$ and $p^*(x_r) - p(x_r)$ are by definition opposite for the points in Z_M we must also have

$$\begin{aligned} |\sum_{\substack{i=0\\i\neq r}}^{n+1} \frac{\sigma_i}{W(x_i)} e(x_i)| = |h^*| \sum_{\substack{i=0\\i\neq r}}^{n+1} |\frac{\sigma_i}{W(x_i}| &+ \\ + |\sigma_r(p^*(x_r) - p(x_r))|, \quad x_r \in Z_M.& \end{aligned} \tag{18}$$

The maximum absolute error e_{max} over all extremal points x_i is defined as

$$e_{max} = \max_{0 \leq i \leq n+1} |e(x_i)|, \tag{19}$$

and it now follows from (18) that e_{max} is bounded by

$$e_{max} \geq |h^*| + \frac{|\dfrac{\sigma_r}{W(x_r)}|}{\displaystyle\sum_{\substack{i=0\\i\neq r}}^{n+1} |\dfrac{\sigma_i}{W(x_i)}|} W(x_r)|p^*(x_r) - p(x_r)|. \tag{20}$$

This is exactly what the theorem states for the points in Z_M and completes the proof. □

Theorem 1 can be used to compute the lower bound δ for a given $p(x)$. The c_is are easily obtained from eq. (11) since the sign of $(-1)^i h^*(p^*(x_i) - p(x_i))$ is known. But we are not really interested in the case of a single $p(x)$. Instead, we need a lower bound δ that holds for all $p(x)$ with integer coefficients a_i. This lower bound will be derived in the next section.

4 Lower bound over all integer polynomials

To compute a lower bound over all integer polynomials $p(x)$, it is necessary to express δ as a function of differences $a_j^* - a_j$, $j = 0, 1, \cdots, n$. This will be done following an approach similar to the one used in [4]. We start by writing the following system of $n+2$ equations with $n+2$ unknowns

$$\begin{aligned} \frac{e(x_i)}{W(x_i)} = \sum_{j=0}^{n} (a_j^* - a_j)\Phi_j(x_i) + \frac{(-1)^i}{W(x_i)} h^*,& \\ i = 0, 1, \cdots, n+1,& \end{aligned} \tag{21}$$

where equations (4) and (13) were used. The unknowns are $a_j^* - a_j$ and h^*. The system matrix is identical to the one in (4) which is already solved to get a_j^*. This means that (21) is always invertible. The inverse can be written

$$\begin{aligned} a_j^* - a_j &= \sum_{i=0}^{n+1} g_{ji} \frac{e(x_i)}{W(x_i)}, \quad j = 0, 1, \cdots, n, \\ h^* &= \sum_{i=0}^{n+1} g_{n+1i} \frac{e(x_i)}{W(x_i)}, \end{aligned} \tag{22}$$

where g_{ji} are the elements of the inverted matrix. To express the differences $a_j^* - a_j$ in terms of $p^*(x_i) - p(x_i)$

insert (13) into (22)

$$\begin{aligned} a_j^* - a_j &= \sum_{i=0}^{n+1} g_{ji}(p^*(x_i) - p(x_i) + \frac{(-1)^i}{W(x_i)} h^*), \\ h^* &= \sum_{i=0}^{n+1} g_{n+1i}(p^*(x_i) - p(x_i) + \frac{(-1)^i}{W(x_i)} h^*). \end{aligned} \tag{23}$$

Setting $a_j = a_j^*$ for all j gives $p(x) = p^*(x)$ for all x and the following property of matrix $[g_{ji}]$ is revealed

$$\sum_{i=0}^{n+1} g_{ji} \frac{(-1)^i}{W(x_i)} = 0, \quad j = 0, 1, \cdots, n, \tag{24}$$

$$\sum_{i=0}^{n+1} g_{n+1i} \frac{(-1)^i}{W(x_i)} = 1. \tag{25}$$

Equations (23) can be rewritten as

$$a_j^* - a_j = \sum_{i=0}^{n+1} g_{ji}(p^*(x_i) - p(x_i)),\ j = 0, 1, \cdots, n, \tag{26}$$

$$0 = \sum_{i=0}^{n+1} g_{n+1i}(p^*(x_i) - p(x_i)). \tag{27}$$

A very useful property of the coefficients g_{n+1i} follows from (27). Since $p(x)$ can be any polynomial it is obvious that g_{n+1i} are the multipliers σ_i described in (7) and (8). This means that the signs of g_{n+1i} alternate as

$$\operatorname{sign}(g_{n+1i+1}) = -\operatorname{sign}(g_{n+1i}),\ i = 0, 1, \cdots, n. \tag{28}$$

Before continuing let us first simplify the notation by defining the modified matrix $[t_{ji}]$

$$t_{ji} = \operatorname{sign}(h^*) \frac{(-1)^i g_{ji}}{W(x_i)}, \quad j, i = 0, 1, \cdots, n+1. \tag{29}$$

The matrix $[t_{ji}]$ has the following properties that come directly from (24) and (25)

$$\sum_{i=0}^{n+1} t_{ji} = 0, \quad j = 0, 1, \cdots, n, \tag{30}$$

$$\sum_{i=0}^{n+1} t_{n+1i} = \operatorname{sign}(h^*). \tag{31}$$

Let us now use t_{ji} instead of g_{ji} and multiply and divide each term in (26) and (27) by $(-1)^i c_i W(x_i)$

$$\begin{aligned} a_j^* - a_j &= \sum_{i=0}^{n+1} \frac{t_{ji}}{c_i} \operatorname{sign}(h^*)(-1)^i c_i W(x_i)(p^*(x_i) - p(x_i)), \\ 0 &= \sum_{i=0}^{n+1} \frac{t_{n+1i}}{c_i} (-1)^i c_i W(x_i)(p^*(x_i) - p(x_i)). \end{aligned} \tag{32}$$

These equations contain the terms $c_i W(x_i)(p^*(x_i) - p(x_i))$ that appear in Theorem 1. It follows from (32) that the lower bound δ is equal to

$$\begin{aligned} \delta &\geq \max_{0 \leq i \leq n+1} |c_i W(x_i)(p^*(x_i) - p(x_i))| \\ &\geq \max_{0 \leq j \leq n} \left(\frac{|a_j^* - a_j|}{\sum_{i=0}^{n+1} |\frac{t_{ji}}{c_i}|} \right). \end{aligned} \tag{33}$$

There is a small problem here because the sign of $p^*(x) - p(x)$ is required by (11) in order to compute c_is. But this is easily solved since (33) assumes that the signs of all the terms in (32) are equal. Or formally

$$\operatorname{sign}(a_j^* - a_j) = \operatorname{sign}(t_{ji} \operatorname{sign}(h^*)(-1)^i (p^*(x_i) - p(x_i))), \tag{34}$$

for all i and j. This means that the $(-1)^i h^*(p^*(x_i) - p(x_i)) < 0$ criterion in (11) can be replaced by $t_{ji}(a_j^* - a_j) < 0$. It is again convenient to divide the points x_i into the subsets Z_{Mj} and Z_{Pj}

$$x_i \in \begin{cases} Z_{Mj} & \text{if} \quad t_{ji} < 0, \\ Z_{Pj} & \text{if} \quad t_{ji} \geq 0\,. \end{cases} \tag{35}$$

Note that $c_i = 1$ for $x_i \in Z_{Pj}$ if $a_j^* - a_j \geq 0$ and for $x_i \in Z_{Mj}$ if $a_j^* - a_j < 0$. The denominator of (33) can be written as

$$\sum_{i=0}^{n+1} |\frac{t_{ji}}{c_i}| = \begin{cases} \sum_{x_i \in Z_{Pj}} \frac{t_{ji}}{c_i} - \sum_{x_i \in Z_{Mj}} t_{ji} & \text{if } a_j^* - a_j < 0 \\ \sum_{x_i \in Z_{Pj}} t_{ji} - \sum_{x_i \in Z_{Mj}} \frac{t_{ji}}{c_i} & \text{if } a_j^* - a_j \geq 0. \end{cases} \tag{36}$$

Let us now remove the assumption about knowing the coefficients a_j. This is necessary in order to get the lower bound for δ (equation (6)) which is valid over all integer polynomials $p(x)$. For any set of optimal coefficients a_j^* there exist integers a_{j+} and a_{j-} that are the nearest upper and lower neighbors of a_j^*. In other words, a_{j+} is an integer that gives the smallest (in an absolute sense) negative difference $a_j^* - a_j$ and a_{j-} is an integer that gives the smallest positive difference $a_j^* - a_j$. Having a_{j+} and a_{j-} it is easy to compute δ by simply inserting a_{j+} and a_{j-} with the corresponding part of (36) into (33). The lower of two values δ is our lower bound because it is obvious that there are no integer coefficients a_j that could possibly give lower deviation increase.

5 Improved lower bound

The lower bound δ that can be computed by (33) and (36) depends on the partitioning of extremal points x_i into the sets Z_{Pj} and Z_{Mj}. It follows from the set definitions (35) that partitioning into Z_{Pj} and Z_{Mj} depends on the sign of t_{ji} only. This sign, however, can be easily changed by multiplying the lower of the equations (32) with a suitable factor f and subtracting it from the other equations. The new set of equations

$$\begin{aligned} a_j^* - a_j = \sum_{i=0}^{n+1} \frac{t_{ji} - f t_{n+1i}}{c_i} \, \mathrm{sign}(h^*)(-1)^i \cdot \\ \cdot\; c_i W(x_i)(p^*(x_i) - p(x_i)), \;\; j = 0, 1, \cdots, n, \end{aligned} \tag{37}$$

leads to an improved lower bound. Note first that a special property of the matrix coefficients t_{n+1i} ensures that factors which make Z_{Pj} or Z_{Mj} empty always exist. We see from (28) and the definition (29) that

$$\mathrm{sign}(t_{n+1i+1}) = \mathrm{sign}(t_{n+1i}), \;\; i = 0, 1, \cdots, n. \tag{38}$$

But the left side of the lower of equations (32) is zero; this means that t_{n+1i} can always be multiplied by -1 without any change to the left side. The coefficients t_{n+1i} in (37) can therefore always be used as if they are positive.

Let us now define the factor $f = f_{mj}$ which makes the set Z_{Pj} empty. It follows from (36) that this is useful for $a_j^* - a_j < 0$ and occurs when

$$t_{ji} - f_{mj}|t_{n+1i}| < 0, \quad i = 0, 1, \cdots, n+1, \tag{39}$$

where the absolute values of t_{n+1i} are used to take advantage of the fact that, as discussed above, they can be used as positive. Before continuing let us simplify the notation by defining the modified coefficients t'_{ji}

$$t'_{ji} = \frac{t_{ji}}{|t_{n+1i}|}, \quad j, i = 0, 1, \cdots, n+1. \tag{40}$$

Factor f_{mj} is nonzero and positive because it follows from (30) that there is always at least one positive t_{ji}. Since all the terms $t_{ji} - f_{mj}|t_{n+1i}|$ are negative the upper of equations (36) becomes equal to

$$\sum_{i=0}^{n+1} |\frac{t_{ji}}{c_i}| = -\sum_{i=1}^{n+1} (t_{ji} - f_{mj}|t_{n+1i}|) = f_{mj}. \tag{41}$$

Properties (30) and (31) of t_{ji} were used in (41). Factor f_{mj} should be as small as possible. The smallest possible f_{mj} that satisfies (39) is equal to

$$f_{mj} = \max_{0 \le i \le n+1} t'_{ji} = t'_{ji_0} \;, \tag{42}$$

where $i = i_0$ denotes the index i at which the maximum is obtained. A similar factor $f = f_{pj}$ which makes the set Z_{Mj} empty must satisfy

$$t_{ji} - f_{pj}|t_{n+1i}| \ge 0, \quad i = 0, 1, \cdots, n+1. \tag{43}$$

The factor f_{pj} is nonzero and negative because there is always at least one negative t_{ji}. The equation for f_{pj} is

$$\sum_{i=0}^{n+1} |\frac{t_{ji}}{c_i}| = \sum_{i=1}^{n+1} (t_{ji} - f_{pj}|t_{n+1i}|) = -f_{pj}. \tag{44}$$

Again, the factor f_{pj} should be as small (in an absolute sense) as possible. The smallest possible f_{pj} is equal to

$$f_{pj} = \min_{0 \le i \le n+1} t'_{ji} = t'_{ji_0} \;, \tag{45}$$

where $i = i_0$ again denotes the index i at which the minimum is obtained.

Equation (33) can now be rewritten as

$$\delta \ge \begin{cases} \max\limits_{0 \le j \le n} \left(\dfrac{a_j^* - a_j}{-f_{mj}} \right) & \text{if } a_j^* - a_j < 0 \\ \max\limits_{0 \le j \le n} \left(\dfrac{a_j^* - a_j}{-f_{pj}} \right) & \text{if } a_j^* - a_j \ge 0. \end{cases} \tag{46}$$

This equation does not need the multipliers c_i at all and is remarkably easy to compute. However, we still need to prove formally that it gives a better lower bound than (33). This is done in the following theorem.

Theorem 2 *Partitioning of extremal points x_i by factors f_{mj} or f_{pj}, defined by (42) and (45), always leads to a lower bound δ that is at least as good or better (higher) than the bound which does not use f_{mj} or f_{pj}.*

Proof: The δ that uses f_{mj} and f_{pj} (46) and δ that does not use them (33) differ in the denominators only. Proving that using f_{mj} and f_{pj} results in better lower bounds is therefore equivalent to proving that their denominators are lower (in an absolute sense). In order to do this examine the multipliers c_i which are defined by (11). Taking into account that σ_is are equivalent to g_{n+1i}s (see eq. (28)) we get

$$c_i = \frac{|\dfrac{\sigma_i}{W(x_i)}|}{\sum\limits_{\substack{k=0 \\ k \ne i}}^{n+1} |\dfrac{\sigma_k}{W(x_k)}|} = \frac{|\dfrac{g_{n+1i}}{W(x_i)}|}{\sum\limits_{\substack{k=0 \\ k \ne i}}^{n+1} |\dfrac{g_{n+1k}}{W(x_k)}|} = \frac{|t_{n+1i}|}{\sum\limits_{\substack{k=0 \\ k \ne i}}^{n+1} |t_{n+1k}|} \;, \tag{47}$$

where definition (29) was also used. According to (31) the sum of all $|t_{n+1k}|$ equals 1 and c_is are also equal to

$$\frac{1}{c_i} = \frac{1}{|t_{n+1i}|} - 1, \;\; i = 0, 1, \cdots, n+1. \tag{48}$$

Let us use this and prove the case $a_j^* - aj < 0$ first. The denominator in (33) is given by (36) and can be rewritten

$$\begin{aligned} \sum_{x_i \in Z_{Pj}} \frac{t_{ji}}{c_i} - \sum_{x_i \in Z_{Mj}} t_{ji} = \\ \sum_{x_i \in Z_{Pj}} t'_{ji} - \sum_{x_i \in Z_{Pj}} t_{ji} - \sum_{x_i \in Z_{Mj}} t_{ji} = \sum_{x_i \in Z_{Pj}} t'_{ji} \;, \end{aligned} \tag{49}$$

where $\sum_{x_i \in Z_{Pj}} t_{ji} + \sum_{x_i \in Z_{Mj}} t_{ji} = 0$ comes from (30). Since f_{mj} is defined by (42) there is also

$$\sum_{x_i \in Z_{Pj}} t'_{ji} = f_{mj} + \sum_{\substack{i \in Z_{Pj} \\ i \ne i_0}} t'_{ji} \;. \tag{50}$$

Now f_{mj} is positive and so are all the t'_{ji} in Z_{Pj}. The absolute value of (50) cannot be lower than $|f_{mj}|$ and the theorem is proved for $a_j^* - aj < 0$. The proof for $a_j^* - aj \ge 0$ is almost identical and need not be repeated. □

6 Level 2 lower bound

The lower bound derived in the previous section can be significantly improved if 2 (or more) equations from (37) are multiplied and subtracted. We will call this “level 2 lower bound”. Let us start by multiplying the equation $j = k$ with a factor γ and subtracting it from equation $j = l$. Equations (37) are replaced by

$$\begin{aligned} a_l^* - a_l - \gamma(a_k^* - a_k) = & \sum_{i=0}^{n+1} (t_{ji} - \gamma t_{ki} - f t_{n+1i}) \cdot \\ & \cdot \operatorname{sign}(h^*)(-1)^i W(x_i)(p^*(x_i) - p(x_i)), \end{aligned} \tag{51}$$

where c_is are now omitted. Following the same approach as before we define factors $f = f_{mlk}$ and $f = f_{plk}$ as

$$t_{li} - \gamma t_{ki} - f_{mlk}|t_{n+1i}| < 0, \; i = 0, \cdots, n+1, \tag{52}$$

$$t_{li} - \gamma t_{ki} - f_{plk}|t_{n+1i}| \geq 0, \; i = 0, \cdots, n+1. \tag{53}$$

The smallest f_{mlk} that satisfies (52) is

$$f_{mlk} = \max_{0 \leq i \leq n+1} (t'_{li} - \gamma t'_{ki}) = t'_{li_0} - \gamma t'_{ki_0} \ , \tag{54}$$

where $i = i_0$ again denotes the maximal index. The smallest (in an absolute sense) f_{plk} that satisfies (53) is similar ($\max$ is replaced by $\min$). The level 2 lower bound δ_{lk} (equations l and k) for $a_l^* - a_l - \gamma(a_k^* - a_k) < 0$ can now be written as

$$\delta_{lk} = \frac{a_l^* - a_l - \gamma(a_k^* - a_k)}{-f_{mlk}} \ , \tag{55}$$

and a similar equation can be written for $a_l^* - a_l - \gamma(a_k^* - a_k) \geq 0$. The lower bound δ_{lk} is a function of γ which must be chosen so that δ_{lk} is as high as possible. To find such γ it is necessary to solve the following optimization problem: Find γ to

$$\text{maximize} \quad \frac{a_l^* - a_l - \gamma(a_k^* - a_k)}{-f_{mlk}} \tag{56}$$

$$\begin{aligned} \text{subject to} \quad & t_{li} - \gamma t_{ki} - f_{mlk}|t_{n+1i}| < 0, \\ & i = 0, 1, \cdots, n+1. \end{aligned} \tag{57}$$

This problem describes the case $a_l^* - a_l - \gamma(a_k^* - a_k) < 0$. The other case is almost identical with f_{plk} replacing f_{mlk}. It is obvious that any algorithm that solves the above also solves the f_{plk} case. The objective function (56) is nonlinear (f_{mlk} is a function of γ) while the constraints (57) are linear. This type of optimization problem is usually solved by the so-called gradient-projection method and is not very difficult. To see how the algorithm works let us rewrite (55) by using f_{mlk} from (54)

$$\delta_{lk} = \frac{a_l^* - a_l - \gamma(a_k^* - a_k)}{t'_{li_0} - \gamma t'_{ki_0}} \ , \tag{58}$$

where i_0 is the maximal index for a given γ. The first derivative of (58) is equal to

$$\frac{d\,\delta_{lk}}{d\,\gamma} = \frac{(a_l^* - a_l)t'_{ki_0} - (a_k^* - a_k)t'_{li_0}}{(t'_{li_0} - \gamma t'_{ki_0})^2} \ . \tag{59}$$

It is clear that if γ is replaced by $\gamma + \Delta\gamma$ so that

$$\operatorname{sign}(\frac{d\,\delta_{lk}}{d\,\gamma})\Delta\gamma > 0, \tag{60}$$

the lower bound δ_{lk} will increase provided that the maximal index i_0 does not change. In other words, $\Delta\gamma$ must be small enough. Note that the sign of the derivative (59) changes only when i_0 is no longer the maximal index as defined in (54). The maximal $\Delta\gamma$ which does not change i_0 is limited by (54) when the following equality is reached for some i

$$t'_{li_0} - (\gamma + \Delta\gamma)t'_{ki_0} = t'_{li} - (\gamma + \Delta\gamma)t'_{ki} \ . \tag{61}$$

This gives the limit for $\Delta\gamma$

$$\begin{aligned} \Delta\gamma &= \min_{\substack{0 \leq i \leq n+1 \\ i \neq i_0}} \left(\frac{t'_{li_0} - t'_{li}}{t'_{ki_0} - t'_{ki}} - \gamma \right), \\ & \operatorname{sign}(\frac{d\,\delta_{lk}}{d\,\gamma})\Delta\gamma > 0, \end{aligned} \tag{62}$$

where only indices i giving $\Delta\gamma$ that satisfies (60) are used in the search for minimum. It is useful to denote the minimal index $i = i_1$ and rewrite (62) as

$$\Delta\gamma = \frac{t'_{li_0} - t'_{li_1}}{t'_{ki_0} - t'_{ki_1}} - \gamma. \tag{63}$$

The algorithm that solves our optimization problem can now be described. The steps are as follows:

1. Start with $\gamma = 0$ and find the maximal index i_0 from (54) for l. This index is because of $\gamma = 0$ identical to the level 1 index given by (42) (if $a_l^* - a_l \geq 0$ use (45)). Compute the level 1 lower bounds δ_l and δ_k at index i_0. If $\delta_l \geq \delta_k$, or in other words, if

$$\frac{a_l^* - a_l}{t'_{li_0}} \geq \frac{a_k^* - a_k}{t'_{ki_0}} \ , \tag{64}$$

go to step 2 else exchange equations l and k (multiply equation l by γ and subtract it from k) then go to step 2. This exchange ensures that γ is finite.

2. Compute the derivative (59) and keep its sign.
3. Use (62) to compute $\Delta\gamma$ and the minimal index i_1. Stop if no $\Delta\gamma$ is found or if $\Delta\gamma = 0$.
4. Replace γ by the new value

$$\gamma \leftarrow \gamma + \Delta\gamma \ , \tag{65}$$

and compute δ_{lk} using (58).

5. Replace index i_0 by the new value

$$i_0 \leftarrow i_1. \tag{66}$$

6. Compute the new derivative (59) and compare its sign with the previous one. If they are the same, return to step 3 for the next iteration, or else stop. Note that both $i = i_0$ and $i = i_1$ must now be excluded in the search for the minimum in (62) since we would always get $\Delta\gamma = 0$ otherwise.

The algorithm is robust and typically needs 2 or 3 iterations before the optimal γ and δ_{lk} are found. The level 2 lower bound δ_{lk} that is computed by the algorithm is valid for a given a_l and a_k. Note that the algorithm automatically takes care of the sign of $a_l^* - a_l - \gamma(a_k^* - a_k)$ (or $a_k^* - a_k - \gamma(a_l^* - a_l)$) if the exchange was done in step 1). This means that it is no longer necessary to treat cases f_{mlk} and f_{plk} separately.

It is, however, necessary to check all possible combinations of integers a_l and a_k to get the level 2 lower bound over all possible integer coefficients. In other words, we must find

$$\min_{a_l, a_k} \delta_{lk}. \tag{67}$$

This is not difficult because the function (58) for δ_{lk} is convex and all local optima are global. Still, it is necessary to compute δ_{lk} for all pairs of a_{l-}, a_{l+} and a_{k-}, a_{k+} and then check if adding $+1$ or -1 to either possibly reduces δ_{lk}. A new optimal γ must be computed each time.

7 Experimental results

The level 2 bound δ was implemented in a program for the optimal finite wordlength FIR digital filter design. The program is based on the branch-and-bound algorithm.

Sixteen filters with five different sets of frequency-domain specifications, denoted A through E, were used for testing. The frequency specifications are identical to those that were used in [2]. A is a low-pass filter with $W(x) = 1$ in passband and stopband. B is the same, except the stopband has $W(x) = 10$. C is a bandstop filter with $W(x) = 1$ in all bands, while D has $W(x) = 10$ in the stopband. E is a low-pass filter similar to A whose passband and stopband do not include $x = 0$ and $x = \pi$. We denote by A15/5 the filter design problem for specification A, length 15 (8 independent coefficients), and $b = 5$ bits (sign included); and similarly for A25/5, B15/7, and so on.

Table 1 shows a summary of the results, comparing the number of branch-and-bound subproblems that must be solved when lower bound δ is used and when it is not used. The results show a significant improvement which averages at about 2.5.

Filter	Number of subproblems		Improvement
	With δ	Without δ	
A15/5	43	125	2.91
A25/5	180	497	2.76
A35/7	797	1901	2.39
B15/7	86	275	3.20
B25/7	282	584	2.07
B35/7	1358	2873	2.12
B45/10	46479	102071	2.20
C15/5	63	152	2.41
C25/5	147	368	2.50
C35/7	4507	11720	2.60
D15/7	82	263	3.21
D25/7	860	2000	2.33
D35/7	7091	17939	2.53
D45/9	133802	333836	2.50
E25/6	393	899	2.29
E35/11	17596	47657	2.71

Table 1. Experimental results of the lower bound δ effectiveness for 16 cases. The program used is identical for "with δ" and "without δ" case, except for the lower bound

8 References

[1] D. M. Kodek, "Design of optimal finite word-length FIR digital filters using integer programming techniques," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-28, pp. 304-308, June 1980.

[2] D. M. Kodek, K. Steiglitz, "Comparison of optimal and local search methods for designing finite wordlength FIR digital filters," *IEEE Trans. on Circuits and Systems*, vol. CAS-28, pp. 28-32, January 1981.

[3] M. J. D. Powell, "*Approximation theory and methods*", Cambridge University Press, Cambridge, pp.85-110, 1981.

[4] W. P. Niedringhaus, K. Steiglitz, D. M. Kodek, "An easily computed performance bound for finite wordlength direct-form FIR digital filters," *IEEE Trans. Circuits Syst.*, vol. CAS-29, pp. 191-193, March 18-20, 1982.

[5] D. M. Kodek, "A theoretical limit for finite wordlength FIR digital filters," *Proc. of the 1998 CISS Conference*, pp. 836-841, Princeton, March 20-22, 1998.

[6] C. H. Papadimitrou, K. Steiglitz, "*Combinatorial optimization*", Prentice-Hall, Englewood Cliffs, N.J., pp. 433-453, 1982.

Dušan M. Kodek received his Dipl.Ing., Master, and Doctoral degrees in Electrical Engineering from the University of Ljubljana in 1970, 1973, and 1975 respectively. He is a professor of computer science at the Faculty of Computer and Information Science, University of Ljubljana. His main research interests are digital signal processing and computer architecture.