ELSEVIER Applied Mathematics and Computation 112 (2000) 113–124

APPLIED MATHEMATICS AND COMPUTATION

www.elsevier.nl/locate/amc

On the approximation of real rational functions via mixed-integer linear programming

Nikos Papamarkos *

Department of Electrical and Computer Engineering, Electric Circuits Analysis Laboratory, Democritus University of Thrace, 67100 Xanthi, Greece

Abstract

This paper is introducing a new method for the approximation of real rational functions via mixed-integer linear programming. The formulation of the linear approximation problem is based on the minimization of a suitable minimax criterion, in combination with a branch and bound linear integer technique. The proposed algorithm can be used in many rational approximation problems, where some coefficients of the rational function are required to take only integer values. The formulation of the problem ensures always the global solution. The proposed algorithm was extensively tested on a variety of problems. An analytical example is presented to illustrate the use and effectiveness of the algorithm. © 2000 Elsevier Science Inc. All rights reserved.

Keywords: Integer programming; Linear programming; Optimization; Rational approximation; Branch and bound tree search

1. Introduction

This paper is concerned with the problem of approximation of real rational functions via integer linear programming. Many algorithms have been proposed for the continuous linear programming case, where there is no condition for the coefficients to obtain integer values. The most important of them are the

* Tel.: +30 541 79566; fax: +30 541 26947; e-mail: papamark@voreas.ee.duth.gr

PII: S0096-3003(99)00052-1

Pade approximation [1] and the Differential Correction algorithm [2]. A fast linear programming method, for real Rational Function Approximation (RFA algorithm) has been proposed by Papamarkos et al. [3,4]. Applications of this algorithm for the design of IIR digital filters, image signature identification and optimal multithresholding of gray level images are described in [5–8]. An important extension of the RFA algorithm to the multirational function approximation is given in [9].

This paper extends the RFA algorithm to the case where some or all of the coefficients of the rational functions are integer variables. The new technique is based on the formulation of an integer approximation problem and on the branch and bound tree search technique [10]. The formulation of the optimization problem and the algorithms used ensure the solution of the problem and additionally that this solution is the global optimal solution. To minimize the memory requirements of the integer programming problem we use the Revised Simplex Algorithm (RSA) [11,12]. In addition, to speed up the entire algorithm an estimation of the global minimum can be given in the beginning of the tree search and this results in a significant cut-off of the tree branches.

The new method was extensively tested on a variety of problems. Experimental results show its effectiveness. In this paper, an analytical example, which confirms the effectiveness of the proposed method, is presented.

2. Description of the algorithm

The general form of a real rational function is

$$H(x) = \frac{A(x)}{B(x)} = \frac{\sum_{j=1}^{N} a_j \tilde{A}_j(x)}{1 + \sum_{j=1}^{M} b_j \tilde{B}_j(x)}, \tag{1}$$

where N and M are given integers, x is the vector of the independent variables, a_j and b_j the unknown coefficients to be determined, and $\tilde{A}_j(x)$, $\tilde{B}_j(x)$ known functions of x.

Let also the independent variables x belong to a sampling region S, i.e., $x_k \in S$ with $k = 1, 2, \ldots, K$, where K is the total number of sampling points. The aim of rational approximation is the determination of the unknown coefficients a_j and b_j such as some conditions to be satisfied. These conditions are referring to the optimization criterion and to the nature of the coefficients values, i.e., if these take only integer values or not. In this approach, the optimization criterion used is based on the minimax criterion. Specifically, at every sampling point k, the optimal criterion requires that

$$|G(x_k) - H(x_k)| \leqslant \delta \quad \forall x_k \in S \quad \text{and} \quad B(x_k) > 0 \tag{2}$$

or

$$\left| G(x_k) - \frac{A(x_k)}{B(x_k)} \right| \leqslant \delta \quad \forall x_k \in S \text{ and } B(x_k) > 0, \tag{3}$$

where δ, a sufficiently small positive value.

To formulate the approximation problem, at each sampling point $x_k \in S$, the variables ξ'_k are defined through the following relation

$$|G(x_k)B(x_k) - A(x_k)| = \xi'_k. \tag{4}$$

Since we accept that $B(x_k) > 0$ for every $x_k \in S$, the relation (4) may also be written in the form

$$\left| G(x_k) - \frac{A(x_k)}{B(x_k)} \right| = \frac{\xi'_k}{B(x_k)}. \tag{5}$$

From Eq. (5), it is clear that for a satisfactory approximation the quantities

$$\delta'_k = \frac{B(x_k)}{\xi'_k} \tag{6}$$

must achieve large positive values. If, therefore,

$$\xi' = \underset{x_k \in S}{\text{maximum}}\{\xi'_k\} \tag{7}$$

and

$$\frac{1}{\delta} = \underset{x_k \in S}{\text{maximum}}\left\{ \frac{\xi'}{B_k} \right\} \tag{8}$$

then, the approximation problem may be formulated as follows:

$$\begin{aligned} &\text{maximize} \quad \delta \\ &\text{subject to the constraints} \\ &|G(x_k)B(x_k) - A(x_k)| \leqslant \xi', \\ &\frac{\xi'}{B(x_k)} \leqslant \frac{1}{\delta} \quad \text{for } k = 1, 2, \ldots, K, \end{aligned} \tag{9}$$

where $\xi' \geqslant 0$ and $\delta > 0$.

From the above formulation of the approximation problem, we can conclude that

- For every $x_k \in S$ the following relations always hold:

$$\left| G(x_k) - \frac{A(x_k)}{B(x_k)} \right| \leqslant \frac{\xi'}{B(x_k)} \leqslant \frac{1}{\delta}. \tag{10}$$

- The second set of constraints guarantees the positiveness of $B(x_k)$ for every $x_k \in S$.

- The approximation problem (9) does not have a linear form.

By using (1), the approximation problem (9) is rewritten in the following form:

$$\begin{aligned}
&\text{maximize} \quad \delta \\
&\text{subject to} \\
&\left| G(x_k)\left(1+\sum_{j=1}^{M} b_j \tilde{B}_j(x_k)\right) - \left(\sum_{j=1}^{N} a_j \tilde{A}_j(x_k)\right)\right| \leqslant \xi', \\
&-\left(1+\sum_{j=1}^{M} b_j \tilde{B}_j(x_k)\right) + \delta\xi_i' \leqslant 0 \quad \text{for } k=1,2,\ldots,K,
\end{aligned} \tag{11}$$

where $\xi' \geqslant 0$ and $\delta > 0$.

The approximation problem, as formulated in (11), is not linear, but it may be converted to a linear one via the transformations:

$$\begin{aligned}
&\xi = \frac{1}{\xi'}, \\
&b_j' = \frac{b_j}{\xi'}, \quad \text{for } j=1,2,\ldots,M, \\
&a_j' = \frac{a_j}{\xi'}, \quad \text{for } j=1,2,\ldots,N.
\end{aligned} \tag{12}$$

Now, using (12), the approximation problem is reformulated as

$$\begin{aligned}
&\text{maximize} \quad \delta \\
&\text{subject to} \\
&G_k\left(\xi+\sum_{j=1}^{M} b_j' \tilde{B}_j(x_k)\right) - \sum_{j=1}^{N} a_j' \tilde{A}_j(x_k) \leqslant 1, \\
&-G_k\left(\xi+\sum_{j=1}^{M} b_j' \tilde{B}_j(x_k)\right) + \sum_{j=1}^{N} a_j' \tilde{A}_j(x_k) \leqslant 1, \\
&-\left(\xi+\sum_{j=1}^{M} b_j' \tilde{B}_j(x_k)\right) + \delta \leqslant 0 \quad \text{for } k=1,2,\ldots,K,
\end{aligned} \tag{13}$$

where $\xi' \geqslant 0$ and $\delta > 0$.

The above problem (13) shows that the approximation problem (11) can be easily linearized by using relations (12). If some or all of the function coefficients must take only integer values then an integer approximation problem must be formulated.

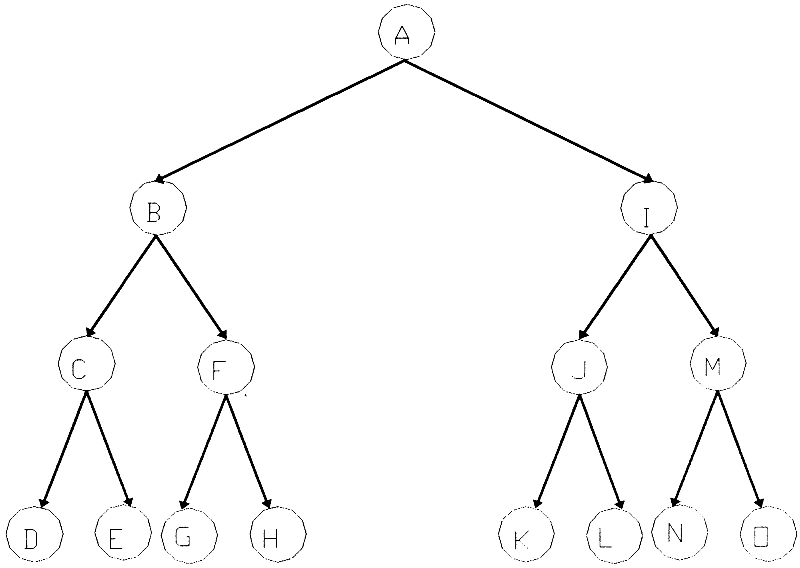

Fig. 1. A typical structure of a binary tree.

2.1. Formulation of the mixed integer linear programming algorithm

The Mixed Integer Linear Programming Algorithm (MILPA) used is based on a tree search technique and specifically on the branch and bound binary tree algorithm. Fig. 1 shows the structure of a binary tree, where from each node two branches are departed. Each branch of the tree ends into a "child" node. The initial node is the root node and is the practical continuous solution of the problem. At each node of the tree, a linear programming problem is solved in which all the variables are considered to be continuous and where additional linear constraints are embedded in the constraints of the original (initial) problem. These constraints correspond to the specific path of the current node. The entire tree technique is based on a systematic search of the tree nodes and can be followed by different tasks. In this implementation, the so-called pre-order technique is used, which for the tree of Fig. 1 corresponds to the (ABCDEFGHIJKLMNO) path. The pre-order technique speeds up the entire process by cutting branches of the tree. In addition, if a good estimation of the final integer solution is known from the beginning, then the pre-order technique achieves the optimal solution faster.

The tree search technique is implemented by using a three-dimensional matrix. This matrix defines for each node the additional linear constraints that must be added to the initial problem. The first column of this matrix gives the consecutive number of the variables. The second column contains the constant values of the additional constraints. Finally, the third column contains only the values -1 or 1, which define if the additional constraints are of type $\leqslant$ or $\geqslant$, respectively. To define the end of the path, all the elements of the last row of the matrix take the value -5. For example the following matrix

$$\begin{bmatrix} 1 & 5 & -1 \\ 2 & 8 & 1 \\ 3 & 14 & -1 \\ -5 & -5 & -5 \end{bmatrix} \tag{14}$$

defines that we are in node E and that the constraints

$$\begin{aligned} &x_1 \leqslant 5 \\ &x_2 \geqslant 8 \\ &x_3 \leqslant 14 \end{aligned} \tag{15}$$

must be added to the original linear problem.

In each node, every integer variable produces two linear programming problems. More specifically, if in a node $\tilde{x}_j$ is the value of the variable x_j then the solution set Q of the integer problem is divided into two sets Q_1 and Q_2 such as

$$Q = Q_1 \cup Q_2 \quad \text{and} \quad Q_1 \cap Q_2 = 0 \tag{16}$$

with

$$\begin{aligned} Q_1 &= Q \cap \{x_j\colon x_j \leqslant [\tilde{x}_j]\}, \\ Q_2 &= Q \cap \{x_j\colon x_j \geqslant [\tilde{x}_j] + 1\}, \end{aligned}$$

where $[x]$ denotes the integer part of x. Obviously, every solution of Q belongs to Q_1 or to Q_2.

As it is analyzed, each node decomposes the problem to two new linear programming problems, which correspond to the two "child" nodes. This top-down process is finished when it is found a solution, which is

(a) an integer solution,
(b) a non-feasible solution,
(c) a non-limited solution,
(d) a non-integer solution, which is not better from a previous solution.

The integer algorithm described above cannot be applied directly to the problem (13), because the variables of this problem are not the coefficients of the rational function but the coefficients divided by the auxiliary variable ξ'. In other words, we want integer values for the variables a_j and b_j and not (or > for instead of) for the variables $a'_j = a_j/\xi'$ and $b'_j = b_j/\xi'$. However, if the tree search algorithm refers to the variables a_j and b_j, then, at each tree node, additional constraints of the form:

$$\begin{aligned} & a_j \leqslant p, \\ & a_j \geqslant p, \\ & b_j \leqslant p, \\ & b_j \geqslant p. \end{aligned} \tag{17}$$

can be transformed into the following form:

$$\begin{aligned} -\rho\xi + a'_j &\leqslant 0, \\ \rho\xi - a'_j &\leqslant 0, \\ -\sigma\xi + b'_j &\leqslant 0, \\ \sigma\xi - b'_j &\leqslant 0. \end{aligned} \tag{18}$$

This form is suitable for the linear programming problem (13) and therefore can be added to it. Doing this, at each tree node a linear programming problem is solved in the following form:

$$\begin{aligned} & \text{maximize} \quad \delta \\ & \text{subject to} \\ & G_k\left(\xi + \sum_{j=1}^{M} b'_j \tilde{B}_j(x_k)\right) - \sum_{j=1}^{N} a'_j \tilde{A}_j(x_k) \leqslant 1, \\ & -G_k\left(\xi + \sum_{j=1}^{M} b'_j \tilde{B}_j(x_k)\right) + \sum_{j=1}^{N} a'_j \tilde{A}_j(x_k) \leqslant 1, \\ & -\left(\xi + \sum_{j=1}^{M} b'_j \tilde{B}_j(x_k)\right) + \delta \leqslant 0, \\ & -\rho_{n_1}\xi + a'_j \leqslant 0, \quad \rho_{n_2}\xi - a'_j \leqslant 0, \qquad j \in [1 \ldots N], \\ & -\sigma_{m_1}\xi + b'_j \leqslant 0, \quad \sigma_{m_2}\xi - b'_j \leqslant 0, \qquad j \in [1 \ldots M], \end{aligned} \tag{19}$$

where ρ_{n_1}, ρ_{n_2} and σ_{m_1}, σ_{m_2} integers, $\xi' \geqslant 0$, $\delta > 0$ and $k = 1, 2, \ldots, K$.

It is noted that in the linear problem (19), the variables a'_j and b'_j are unrestricted in sign. On the other hand, linear programming algorithms, such as the RSA, require only non-negative variables. A simple way to overcome this difficulty is to shift the variables by a suitable shift constant $V > 0$. Therefore, if we define

$$
\begin{aligned}
&p_j = V + a'_j \quad \text{for } j = 1,2,\ldots,N,\\
&q_j = V + b'_j \quad \text{for } j = 1,2,\ldots,M,
\end{aligned}
\tag{20}
$$

then, the linear problem (19) is reformulated as

$$
\begin{aligned}
&\text{maximize} \quad \delta\\
&\text{subject to}\\
&G_k\left(\xi + \sum_{j=1}^{M} q_j \tilde{B}_j(x_k)\right) - \sum_{j=1}^{N} p_j \tilde{A}_j(x_k) + d_k y \leqslant 1,\\
&-G_k\left(\xi + \sum_{j=1}^{M} q_j \tilde{B}_j(x_k)\right) + \sum_{j=1}^{N} p_j \tilde{A}_j(x_k) - d_k y \leqslant 1,\\
&-\left(\xi + \sum_{j=1}^{M} q_j \tilde{B}_j(x_k)\right) + \delta + h_k y \leqslant 0 \quad y = 1,\\
&-\rho_{n_1}\xi + p_j \leqslant V, \quad -\rho_{n_2}\xi + p_j \geqslant V, \qquad j \in [1\ldots N],\\
&-\sigma_{m_1}\xi + q_j \leqslant V, \quad -\sigma_{m_2}\xi + q_j \geqslant V, \qquad j \in [1\ldots M],
\end{aligned}
\tag{21}
$$

where ρ_{n_1}, ρ_{n_2} and σ_{m_1}, σ_{m_2} known integers, $\xi' \geqslant 0$, $\delta > 0$ and $k = 1,2,\ldots,K$. All the variables are positive, and

$$
h_k = V \sum_{j=1}^{M} \tilde{B}_j(x_k), \tag{22}
$$

$$
d_k = -G_k V \sum_{j=1}^{M} \tilde{B}_j(x_k) + V \sum_{j=1}^{N} \tilde{A}_j(x_k) = -G_k h_k + v \sum_{j=1}^{N} \tilde{A}_j(x_k). \tag{23}
$$

Problem (21) is the linear programming problem that must be solved in each node. It corresponds to a total number of $M + N + 3$ variables and $3K + D_n + 1$ linear constraints, where D_n is the number of the additional linear constraints in the node n. Usually, $M + N + 3 \ll 3K + D_n + 1$, which means that the linear programming problem must be solved by its dual, using a method such as the RSA.

According to the above analysis, the mixed integer linear rational approximation algorithm consists of the following steps:

Step 1. The integer and continuous coefficients are defined and the LRA approximation problem (13) is formulated. If it is known from the beginning, we can give an estimation of the global-integer solution.

Step 2. The LRA problem is solved as a continuous variable problem. Let X the vector of variables that must take integer values in the final mixed-integer solution.

Step 3. If all the elements of X are integers then the optimal integer solution is found.

Step 4. A new linear programming sub-problem of the form (21) is formulated and solved by the RSA.

Step 5. If the sub-problem does not have any solution or its solution is worse than a previous solution, then go to Step 6. If the sub-problem has an integer solution better than a previous integer solution then put $X_{\text{in}} = X$ and go to Step 6. In any other case, go to Step 7.

Step 6. Move to the next node of the tree and go to Step 7. If all the nodes of the tree are examined then go to Step 7.

Step 7. Define the additional linear constraints and go to Step 4.

Step 8. The optimal solution is found.

3. Example

The proposed method was successfully tested on a variety of approximation problems. In this paper we use the MIRFA algorithm for the approximation of the rational function with $n = m = 3$ coefficients. The sampling data are produced from the rational function

$$f(x) = \frac{4 + 0.2x + 3x^2}{1 + 2x + 0.8x^2 + 4x^3} \tag{24}$$

in the rage [0,2]. Specifically, $x_k,\ k = 1, \ldots, K$, with $K = 30$, equal spaced values are used for the independent variable x, which from (24) produce 30 sampling data for the function $f(x)$. The values of $f(x_k),\ k = 1, \ldots, K$ are truncated to the fifth decimal point.

The $\tilde{A}_i(x)$ and $\tilde{B}_i(x)$ functions have the following form

$$\tilde{A}_i(x) = x^{i-1}, \quad \tilde{B}_i(x) = x^i. \tag{25}$$

The application of the RFA algorithm, after 22 loops, gives the following continuous optimal solution

$\delta = 336215.824$	$\xi = 2.97428 \times 10^{-6}$
$a_1 = 4.0$	$b_1 = 1.99778$
$a_2 = 0.200093$	$b_2 = 0.799946$
$a_3 = 3.00022$	$b_3 = 4.00034$

Obviously, the values of the optimal coefficients are not exactly the initial of (24) due to the truncating procedure.

The application of the MIRFA algorithm, with $V = 100$, gives the following optimal integer solution:

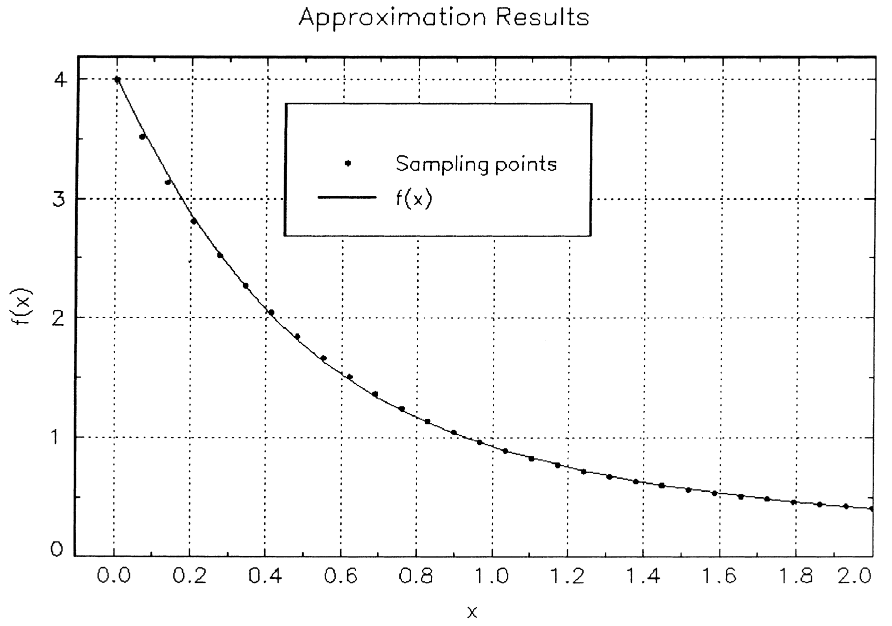

Fig. 2. Approximation results.

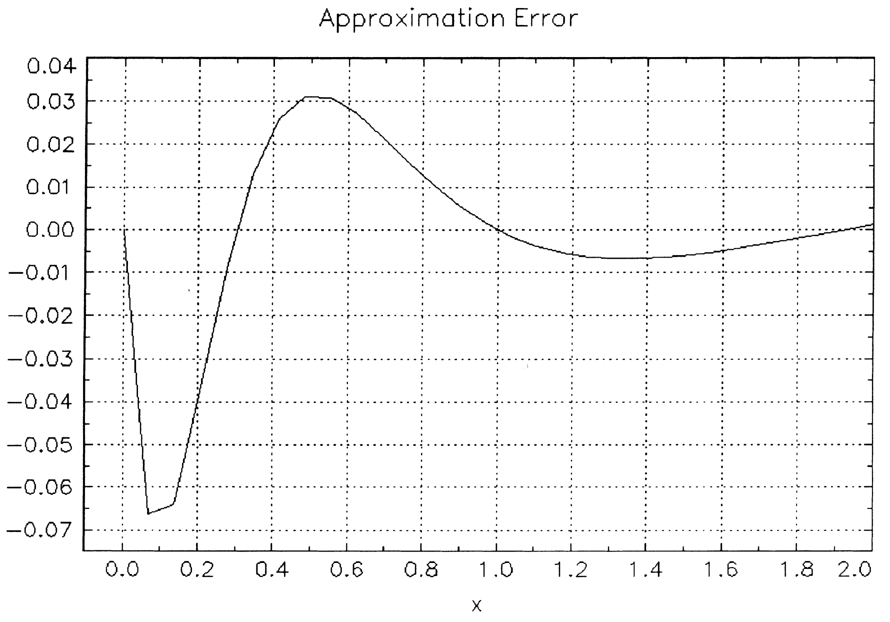

Fig. 3. Final approximation error.

$\delta = 5.545848$	$\xi = 0.18031509$
$a_1 = 4$	$b_1 = 3$
$a_2 = 6$	$b_2 = 5$
$a_3 = 2$	$b_3 = 4$

Fig. 2 depicts the comparison between $f(x)$ and the solutions of RFA and MIRFA. Fig. 3 shows the final approximation error in each sampling point.

4. Conclusions

This paper has introduced a method for approximation of real rational functions via mixed integer linear programming. The proposed algorithm extends the RFA algorithm to the case where some of the function coefficients take integer values. The conversion of the RFA linear programming problem to a mixed integer one, is not a simple task. Our approach is based on the branch and bound tree search technique and on a suitable transformation of the additional constraints produced on tree nodes. Thus, on each tree node a linear programming problem is formulated and solved by using the dual form and the RSA.

The formulation of the problem and the use of linear programming guarantees that the approximation problem has always a solution and that this solution is the global optimal solution.

References

[1] H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes: The Art Scientific Computing, Cambridge University Press, Cambridge, 1986.

[2] I. Barrodale, M.J. Powell, F.D.K. Roberts, The differential correction algorithm for rational l_∞ approximation, SIAM Journal of Numerical Analysis 9 (1972) 493–504.

[3] N. Papamarkos, G. Vachtsevanos, B. Mertzios, On the optimum approximation of real rational functions via linear programming, Applied Mathematics and Computation 26 (1988) 267–287.

[4] N. Papamarkos, A program for the optimum approximation of real rational functions via linear programming, Advances in Engineering Software 11 (1) (1989) 37–48.

[5] N. Papamarkos, G. Vachtsevanos, A fast linear programming approach to the design of 1-D IIR digital filters, International Journal of Circuits Theory Applications 18 (1990) 325–333.

[6] G. Vachtsevanos, N. Paramarkos, B.G. Mertzios, On the approximation of the magnitude response of 2-D IIR digital filters using linear programming, Circuits, Systems and Signal Processing 6 (1987) 45–60.

[7] N. Papamarkos, I. Spiliotis, A. Zoumadakis, Character recognition by signature approximation, International Journal of Pattern Recognition and Artificial Intelligence 8 (5) (1994) 1171–1187.

[8] N. Papamarkos, B. Gatos, A new approach for multithreshold selection, Computer Vision, Graphics, and Image Processing-Graphical Models and Image Processing 56 (5) (1994) 357–370.
[9] N. Papamarkos, Multirational function approximation via linear programming, Applied Mathematics and Computation, Applied Mathematics and Computation 75 (1996) 75–89.
[10] S. Walukiewicz, Integer Programming (Mathematics and Its Applications), Kluwer Academic Publishers, Dordrecht, 1990.
[11] S.I. Gass, Linear Programming: Methods and Applications, 4th ed., McGraw-Hill, New York, 1975.
[12] D.J. Pannell, Introduction to Practical Linear Programming, Wiley, New York, 1996.