JOURNAL OF APPROXIMATION THEORY 2, 323–336 (1969)

An Interval Programming Algorithm for Discrete Linear L_1 Approximation Problems

P. D. ROBERS[1]

Research Analysis Corporation, McLean, Virginia 22101

AND

A. BEN-ISRAEL[2]

Northwestern University, Evanston, Illinois 60201

INTRODUCTION

The suboptimization method of interval linear programming, recently developed by the authors, is the basis of the algorithm proposed here for the solution of the discrete linear L_1 approximation problem, stated in Section 1. Interval programming is introduced in Section 2. The algorithm is presented in Section 3 and its advantages over previous linear programming approaches are discussed in Section 4. Two worked examples are included.

1. THE DISCRETE LINEAR L_1 APPROXIMATION PROBLEM

This problem is stated as:

$$\text{minimize} \qquad \sum_{i=1}^{n} |\epsilon_i|, \tag{1}$$

subject to

$$Fx + \epsilon = t,$$

where the matrix $F = (f_{ij})$ and the vector $t = (t_i)$ are given; the vectors $\epsilon = (\epsilon_i)$ and $x = (x_j)$ are to be found $(i = 1, \ldots, n; j = 1, \ldots, m)$.

Such problems arise, for instance, if a given data $\{(s_i, t_i) : i = 1, \ldots, n\}$ is to be approximated, in the sense of the L_1 norm, by a linear combination of given functions $\{g_j(\ ) : j = 1, \ldots, m\}$. The problem is then:

$$\text{minimize} \qquad \sum_{i=1}^{n} |\epsilon_i|, \tag{2}$$

[1] This research was partly supported by Department of the Army Contract No. DA-44-188-ARO-1.

[2] This research was partly supported by the National Science Foundation, Project GP 7550, and by the U.S. Army Research Office—Durham, Contract No. DA-31-124-ARO-D-322, at Northwestern University.

subject to

$$\sum_{j=1}^{m} g_j(s_i)\, x_j + \epsilon_i = t_i \qquad (i = 1, \ldots, n),$$

which is (1) with $f_{ij} = g_j(s_i)$.

The dual problem (see, e.g., [6], [7]) of the linear program (1) is:

maximize $$t^T y, \tag{3}$$

subject to

$$F^T y = 0$$

$$-e \leqq y \leqq e,$$

where $e^T = (1, 1, \ldots, 1)$.

The problem (1), and the analogously defined discrete linear L_∞ approximation problem, were solved by linear programming techniques (see, e.g., [1]–[3], [12]–[14], and the survey [9]). If m is large, then (1) can be solved via its dual (3), see, e.g., [14], [9]. Likewise, in this paper, (1) is solved via (3), which is solved by the suboptimization method of interval programming, [10], [11]. Possible advantages of this approach are discussed in Section 4.

2. Interval Linear Programming

This name, abbreviated IP, is the term coined in [10] to denote the theory, computations and applications of extremization problems (called *interval programs* or IP's) of the form

maximize $$c^T x \tag{4}$$

subject to

$$b^- \leqq Ax \leqq b^+,$$

where the matrix A, and the vectors b^-, b^+, c are given. IP is an alternative formulation of linear programming, see, e.g., [5], offering explicit solutions in some special cases, see, e.g., [4], [15], and efficient iterative methods in the general case [10], [11]. A simplified version of the suboptimization method of [11] is used here to solve (3), which is an IP with

$$A = \begin{pmatrix} F^T \\ \hline I \end{pmatrix}, \qquad b^- = \begin{pmatrix} 0 \\ \hline -e \end{pmatrix}, \qquad b^+ = \begin{pmatrix} 0 \\ \hline e \end{pmatrix}.$$

The simplification is possible because the IP (3) is feasible, bounded and its coefficient matrix A is of full column rank, thus eliminating the corresponding steps in [11].

3. The Suboptimization Algorithm

This finite iterative method for solving

$$\text{maximize} \qquad t^T y \tag{3}$$

subject to

$$\begin{pmatrix} 0 \\ \hline -e \end{pmatrix} \leqq \begin{pmatrix} F^T \\ \hline I \end{pmatrix} y \leqq \begin{pmatrix} 0 \\ \hline e \end{pmatrix},$$

solves at the νth iteration, $\nu \geqslant 1$, an auxiliary problem, denoted (AP. ν):

$$\text{maximize} \qquad t^T y \tag{4}$$

subject to

$$b^{(\nu)-} \leqq A^{(\nu)} y \leqq b^{(\nu)+} \tag{5.ν}$$

$$b^{h(\nu)-} \leqslant a^{h(\nu)} y \leqslant b^{h(\nu)+}, \tag{6.ν}$$

where (5.ν) is a set of n constraints from (3), chosen so that $A^{(\nu)}$ is non-singular, and (6.ν) is a single constraint of (3).

(AP. ν), $\nu \geqslant 1$, is thus a subproblem of (3), accounting for the title of this section.

The νth iteration, $\nu \geqslant 1$.

Denote by:

$y^{(\nu-1)}$—the optimal solution of (4), (5.ν)

$y^{(\nu)}$ —the optimal solution of (4), (5.ν), (6.ν).

For $\nu = 1$, let

$$A^{(1)} = I, \qquad b^{(1)-} = -e, \qquad b^{(1)+} = e, \tag{7}$$

and let (6.1) be any constraint from the remaining constraints

$$0 \leqq F^T y \leqq 0$$

of (3). Then $y^{(0)} = (y_i^{(0)})$, $i = 1, \ldots, n$,
the solution of

$$\text{maximize} \qquad t^T y \tag{4}$$

subject to

$$-e \leqq y \leqq e \tag{5.1}$$

is clearly:

$$y_i^{(0)} = \begin{cases} 1 \\ \theta_i \\ -1 \end{cases} \quad \text{if } t_i \begin{cases} >0 \\ =0 \\ <0 \end{cases} \tag{8}$$

where $-1 \leqslant \theta_i \leqslant 1$ is arbitrary.

For $\nu \geqslant 1$ we assume that $y^{(\nu-1)}$ and $(A^{(\nu)})^{-1}$ are known.
If $y^{(\nu-1)}$ satisfies (6.ν) then

$$y^{(\nu)} = y^{(\nu-1)} \tag{9.ν}$$

Otherwise $y^{(\nu)}$ is obtained from $y^{(\nu-1)}$ as follows: Let

$$\Delta = \begin{cases} a^{h(\nu)} y^{(\nu-1)} - b^{h(\nu)+} & \text{if positive} \\ a^{h(\nu)} y^{(\nu-1)} - b^{h(\nu)-} & \text{if negative} \end{cases} \tag{10.ν}$$

be the amount by which (6.ν) is violated at $y^{(\nu-1)}$, and let

$$Q = \{i: 1 \leqslant i \leqslant n, (a^{h(\nu)}(A^{(\nu)})^{-1})_i \neq 0, \gamma_i \geqslant 0\} \tag{11.ν}$$

where

$$\gamma_i = \frac{(t^T(A^{(\nu)})^{-1})_i}{(a^{h(\nu)}(A^{(\nu)})^{-1})_i} \operatorname{sign} \Delta. \tag{12.ν}$$

Remark. Q is the index set of the components of $(A^{(\nu)}y^{(\nu-1)})$ which can be changed in order to move (6.ν) towards feasibility, while maintaining feasibility in (5.ν). γ_i is the marginal cost of such a change in $(A^{(\nu)}y^{(\nu-1)})_i$, $i \in Q$.

Now let the (say q) indices in Q be ordered by

$$Q = \{k_1, k_2, \ldots, k_q\} \tag{13.ν}$$

where

$$\gamma_{k_1} \leqslant \gamma_{k_2} \leqslant \ldots \leqslant \gamma_{k_q} \tag{14.ν}$$

and

$$k_i < k_{i+1} \qquad \text{if } \gamma_{k_i} = \gamma_{k_{i+1}}. \tag{15.ν}$$

Remark. (15.ν) is a "tie breaking rule" which, like the Charnes' perturbation in the simplex algorithm, prevents cycling, see, e.g., [*6*].

Then $y^{(\nu)}$ is given by:

$$y^{(\nu)} = y^{(\nu-1)} + (A^{(\nu)})^{-1}\left[\sum_{i=1}^{p-1} \delta_{k_i} e_{k_i} + \theta e_{k_p}\right] \tag{16.ν}$$

where e_k is the kth unit vector,

$$\delta_k = \begin{cases} (b^{(\nu)-} - A^{(\nu)} y^{(\nu-1)})_k & \text{if } \operatorname{sign} \Delta = \operatorname{sign}(a^{h(\nu)}(A^{(\nu)})^{-1})_k, \\ (b^{(\nu)+} - A^{(\nu)} y^{(\nu-1)})_k & \text{if } \operatorname{sign} \Delta = -\operatorname{sign}(a^{h(\nu)}(A^{(\nu)})^{-1})_k, \end{cases} \tag{17.ν}$$

$$\text{for } k \in Q,$$

$$p = \min\left\{i: 1 \leqslant i \leqslant q, \left|\sum_{j=1}^{i} \delta_{k_j}(a^{h(\nu)}(A^{(\nu)})^{-1})_{k_j}\right| \geqslant |\Delta|\right\}, \tag{18.ν}$$

$$\theta = \frac{-\Delta - \sum_{j=1}^{p-1} \delta_{k_j}(a^{h(\nu)}(A^{(\nu)})^{-1})_{k_j}}{(a^{h(\nu)}(A^{(\nu)})^{-1})_{k_p}}. \tag{19.ν}$$

Remark. The δ_k given by (17.ν) is the maximal change of $(A^{(\nu)}y^{(\nu-1)})_k$ moving (6.ν) towards feasibility, without violating (5.ν). Thus $y^{(\nu)}$, given by (16.ν), is obtained from $y^{(\nu-1)}$ by making p such changes. The existence of p, defined by (18.ν), is guaranteed by the feasibility of (3). Indeed, if

$$\left|\sum_{j=1}^{q} \delta_{k_j}(a^{h(\nu)}(A^{(\nu)})^{-1})_{k_j}\right| < |\Delta|,$$

then the constraints (5.ν), (6.ν) are inconsistent. Finally, (14.ν) guarantees that cheaper changes come first. Therefore (16.ν) is an optimal solution of (AP.ν).

If $y^{(\nu)}$, obtained by (9.ν) or (16.ν), satisfies *all* the constraints of (3) (one has to check only the constraints not in (5.ν), (6.ν)) then $y^{(\nu)}$ is clearly also an optimal solution of (3).

Otherwise, (AP.$\nu+1$) is obtained from (AP.ν) as follows: The k_pth constraint (the index k_p is determined by (18.ν) and (13.ν)) of (5.ν) is deleted and replaced by the constraint (6.ν). For the additional constraint (6.$\nu+1$) take any constraint of (3) violated by $y^{(\nu)}$.

The $(\nu+1)$st iteration then solves (AP.$\nu+1$), etc.

This completes the description of the suboptimization method for solving the IP (3).

The following facts concerning the suboptimization algorithm are proved in [*11*]:

(i) This algorithm terminates after a finite number, say f, of iterations. $y^{(f)}$, the optimal solution of (AP.f) computed in the fth iteration (by either (9.f) or (16.f)) is an optimal solution of (3).

(ii) The matrix $A^{(\nu+1)}$, obtained from the nonsingular matrix $A^{(\nu)}$ by replacing its k_pth row with $a^{h(\nu)}$, is likewise nonsingular. (Since $A^{(0)} = I = (A^{(0)})^{-1}$, this suggests computing $(A^{(\nu)})^{-1}$, $\nu \geqslant 1$, by the "product form of the inverse", see, e.g., [7].)

(iii) This algorithm is a dual method [*8*] with *one or more* basic changes per iteration (p basic changes at the νth iteration, where p is given by (18.ν)).

Finally, the optimal solution $y^{(f)}$ of (3) is used to obtain an optimal solution $\epsilon^* = (\epsilon_i^*)$, $x^* = (x_j^*)$ $(i = 1,\ldots,n; j = 1,\ldots,m)$ of (1) as follows: Let $A^{(f+1)}$ be the matrix obtained from $A^{(f)}$ by replacing its k_pth row (see (18.f) and (13.f)) with $a^{h(f)}$. Then

$$x_j^* = \begin{cases} 0 & \text{if row } j \text{ of } F^T \text{ is not in } A^{(f+1)} \\ (t^T(A^{(f+1)})^{-1})_i & \text{if row } j \text{ of } F^T \text{ is the } i\text{th row of } A^{(f+1)} \end{cases} \tag{20.f}$$

and

$$\epsilon^* = t - Fx^*. \tag{21}$$

Indeed, (ϵ^*, x^*) is a feasible solution of (1), by the definition (21). The optimality of (ϵ^*, x^*) follows from the duality theorem of linear programming (see, e.g., [7]) by verifying the fact:

$$t^T y^{(f)} = \sum_{i=1}^{n} |\epsilon_i^*| \tag{22}$$

Example 1. Find $x_1, x_2, \ldots, x_m$ such that the polynomial

$$t = x_1 + x_2 s + \ldots + x_m s^{m-1}$$

is the best L_1 approximation of n given data points:

$$\{(s_i, t_i) : i = 1, \ldots, n\}.$$

That is:

minimize
$$\sum_{i=1}^{n} |\epsilon_i| \tag{23}$$

subject to

$$\sum_{j=1}^{m} x_j s_i^{j-1} + \epsilon_i = t_i, \qquad i = 1, \ldots, n.$$

For example, let $m = 2$, $n = 4$, with

$$\begin{array}{rcccc} i = & 1 & 2 & 3 & 4 \\ s_i = & 0 & 1 & 2 & 3 \\ t_i = & .5 & 1 & 2 & 1. \end{array}$$

Solution. (23) is a special case of (2). Its dual problem, given by (3), is:

maximize
$$\sum_{i=1}^{n} t_i y_i \tag{24}$$

subject to

$$\begin{array}{rcl} 0 \leqslant \sum_{i=1}^{n} s_i^{j-1} y_i \leqslant 0 & & (j = 1, \ldots, m) \\ -1 \leqslant \qquad y_i \leqslant 1 & & (i = 1, \ldots, n). \end{array}$$

With the above data, (24) becomes:

maximize
$$\tfrac{1}{2} y_1 + y_2 + 2y_3 + y_4 \tag{25}$$

subject to

$$\begin{array}{rl} 0 \leqslant y_1 + y_2 + y_3 + y_4 & \leqslant 0 \\ 0 \leqslant y_2 + 2y_3 + 3y_4 & \leqslant 0 \\ -1 \leqslant y_1 & \leqslant 1 \\ -1 \leqslant y_2 & \leqslant 1 \\ -1 \leqslant y_3 & \leqslant 1 \\ -1 \leqslant y_4 & \leqslant 1. \end{array}$$

The solution of (25) by the suboptimization algorithm proceeds as follows.

Iteration 1. The first auxiliary problem is, by (7) and by choosing (6.1) as the 1st constraint of (25),

(AP. 1).

maximize $$\tfrac{1}{2}y_1 + y_2 + 2y_3 + y_4 \tag{4}$$

subject to

$$\begin{cases} -1 \leqslant y_1 \leqslant 1 \\ -1 \leqslant y_2 \leqslant 1 \\ -1 \leqslant y_3 \leqslant 1 \\ -1 \leqslant y_4 \leqslant 1 \end{cases} \tag{5.1}$$

$$0 \leqslant y_1 + y_2 + y_3 + y_4 \leqslant 0. \tag{6.1}$$

The solution $y^{(0)}$ of (4), (5.1) is, by (8):

$$y^{(0)T} = (1, 1, 1, 1).$$

It violates (6.1) by

$$\Delta = 4 - 0 = 4, \qquad \text{see (10.1)}.$$

Using (11.1)–(15.1), we get

$$Q = \{k_1, k_2, k_3, k_4\} = \{1, 2, 4, 3\},$$

since

$$\gamma_1 = \tfrac{1}{2} < \gamma_2 = \gamma_4 = 1 < \gamma_3 = 2.$$

Furthermore,

$$\delta_1 = \delta_2 = \delta_4 = \delta_3 = -2 \qquad \text{by (17.1)}$$

$$p = 2 \qquad \text{by (18.1)}$$

$$\theta = \frac{-4+2}{1} = -2 \qquad \text{by (19.1)},$$

so that, by (16.1),

$$y^{(1)} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} + (-2)\begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} + (-2)\begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \end{pmatrix}$$

is an optimal solution of (AP. 1).

Since $y^{(1)}$ violates the 2nd constraint of (25), it is necessary to perform

Iteration 2. The 2nd auxiliary problem is

(AP. 2).

maximize $$\tfrac{1}{2}y_1 + y_2 + 2y_3 + y_4 \tag{4}$$

subject to

$$\begin{cases} -1 \leqslant y_1 & \leqslant 1 \\ 0 \leqslant y_1 + y_2 + y_3 + y_4 \leqslant 0 \\ -1 \leqslant y_3 & \leqslant 1 \\ -1 \leqslant y_4 & \leqslant 1 \end{cases} \tag{5.2}$$

$$0 \leqslant y_2 + 2y_3 + 3y_4 \leqslant 0, \tag{6.2}$$

obtained from (AP. 1) by replacing the 2nd constraint (since $k_p = 2$ in iteration 1) by (6.1), and by adjoining as (6.2) the 2nd constraint of (25) which is violated by $y^{(1)}$.

Using:

$$y^{(1)T} = (-1, -1, 1, 1)$$

$$A^{(2)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \qquad (A^{(2)})^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$a^{h(2)} = (0, 1, 2, 3), \qquad b^{h(2)-} = 0, \qquad b^{h(2)+} = 0,$$

we compute:

$$\Delta = 4$$
$$t^T(A^{(2)})^{-1} = (-\tfrac{1}{2}, 1, 1, 0)$$
$$a^{h(2)}(A^{(2)})^{-1} = (-1, 1, 1, 2)$$
$$Q = \{k_1 . k_2, k_3, k_4\} = \{4, 1, 2, 3\}$$
$$\delta_4 = -2, \delta_1 = 0, \delta_2 = 0, \delta_3 = -2$$
$$p = 1$$
$$\theta = -2,$$

and, by (16.2),

$$y^{(2)} = \begin{pmatrix} -1 \\ -1 \\ 1 \\ 1 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 0 \\ -1 & 1 & -1 & -1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} (-2) = \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix},$$

which satisfies all the constraints of (25), and is therefore an optimal solution of (25). To find the optimal (x_1^*, x_2^*), we form

$$A^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 2 & 3 \end{pmatrix}$$

and calculate:

$$(A^{(3)})^{-1} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{3}{2} & \frac{3}{2} & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

$$t^T(A^{(3)})^{-1} = (-\tfrac{1}{2}, 1, 1, 0).$$

By (20.2), (21),

$$x_1{}^* = 1, \qquad x_2{}^* = 0$$

$$\epsilon^{*T} = (-\tfrac{1}{2}, 0, 1, 0).$$

The reader can verify that (22) is satisfied, and that

$$t = 1$$

is the line which best approximates the above data in the L_1 sense.

4. Discussion

The algorithm described in Section 3 has two possible advantages over other linear programming solutions of the discrete linear L_1 approximation problem.

The first is computational efficiency. Results of some numerical tests, performed by the authors for a more general algorithm, are described in [*10*] and [*11*]. Our experience is typified by a sample problem ([*3*]) of finding x_1, x_2 so that $t_i = x_1 + x_2 s_i$, $i = 1, \ldots, 6$, is the best L_1 approximation of the data:

$i =$	1	2	3	4	5	6
$s_i =$	0	1	2	3	4	5
$t_i =$	1.520	1.025	0.475	0.010	–0.475	–1.005.

It required, for this example, five iterations or 0.190 sec to solve problem (1) by the simplex algorithm on the CDC6400 at Northwestern University, but only two iterations or 0.128 sec to solve problem (3) by the suboptimization method. A definitive computational evaluation of the technique discussed above would require extensive numerical testing of (i) the simplex method applied to problem (1), (ii) the bounded variables simplex method [7] applied to problem (3), and the suboptimization algorithm applied to problem (3).

To illustrate the second advantage claimed here for the suboptimization algorithm, consider the linear L_1 approximation problem (23) and suppose that an optimal solution $y^{(f)}$ of the dual problem (24) has been found. The optimal solution (ϵ^*, x^*) of (23), computed from $y^{(f)}$ by (20.f) and (21), involves an *approximation error* $\|\epsilon^*\|_1$, given by (22) as:

$$\|\epsilon^*\|_1 = \sum_{i=1}^{n} |\epsilon_i^*| = t^T y^{(f)}. \tag{22}$$

The approximation error can thus be determined by $y^{(f)}$ alone, without computing (ϵ^*, x^*). If $\|\epsilon^*\|_1$ is too large, then it can be decreased by increasing the degree of the approximating polynomial. This means that instead of (23) we solve:

$$\text{minimize} \qquad \sum_{i=1}^{n} |\epsilon_i| \tag{23'}$$

subject to

$$\sum_{j=1}^{m+k} x_j s_i^{j-1} + \epsilon_i = t_i \qquad (i = 1, \ldots, n),$$

where k is some positive integer.

The dual of (23′) is:

$$\text{maximize} \qquad \sum_{i=1}^{n} t_i y_i \tag{24'}$$

subject to

$$0 \leqslant \sum_{i=1}^{n} s_i^{j-1} y_i \leqslant 0 \qquad (j = 1, \ldots, m+k)$$

$$-1 \leqslant \qquad y_i \leqslant 1 \qquad (i = 1, \ldots, n).$$

Problem (24′) is problem (24) with k additional constraints:

$$0 \leqslant \sum_{i=1}^{n} s_i^{j-1} y_i \leqslant 0 \qquad (j = m+1, \ldots, m+k). \tag{26}$$

If the solution $y^{(f)}$ of (24) satisfies (26), then $y^{(f)}$ is also a solution of (24′), and the approximation error (22) is unchanged. In this case the integer k in (23′) and (24′) is further increased, until (26) is violated by $y^{(f)}$.

Any constraint in (26), violated by $y^{(f)}$, can be used as the additional constraint (6.$f+1$) to form the auxiliary problem (AP.$f+1$). An iteration, the $(f+1)$st, of the suboptimization algorithm is then performed, using $y^{(f)}$ as a starting point, and resulting in $y^{(f+1)}$. This is repeated, say, l iterations, until a solution $y^{(f+l)}$ of (24′) is found. The approximation error $t^T y^{(f+l)}$ corresponding to $y^{(f+l)}$ is *less* than that of $y^{(f)}$. If $t^T y^{(f+l)}$ is small enough, then the optimal solution $(\epsilon^{*\prime}, x^{*\prime})$ of (23′) is computed from $y^{(f+l)}$, using (20.$f+l$) and (21).

Thus the suboptimization algorithm enables one, for example,

(i) To increase the degree of approximation (the integer m in (1), (2) or (23)) without completely resolving the problem; the present solution $y^{(f)}$ is used as a starting point.

(ii) For the given data $\{(s_i, t_i) : i = 1, \ldots, n\}$, functions $\{g_j(\ )\}$ and the given bound $\delta > 0$, to determine a "polynomial"

$$\sum_{j=1}^{m} g_j(s) x_j$$

of minimal "degree" m, which satisfies

$$\sum_{i=1}^{n} |t_i - \sum_{j=1}^{m} g_j(s_i) x_j| \leqslant \delta.$$

This advantage, shared by all dual methods [8] for solving (1), is realized only if the data $\{(s_i, t_i) : i = 1, \ldots, n\}$ is fixed. If the number n of data points is increased during the computations, then similar advantages are enjoyed by primal linear programming methods for solving (1), see, e.g., [3].

Example 2. Find a polynomial

$$\sum_{j=1}^{m} x_j s^{j-1}$$

of minimal m, satisfying

$$\sum_{i=1}^{4} \left| t_i - \sum_{j=1}^{m} x_j s_i^{j-1} \right| \leqslant 1 \qquad (27)$$

for the data $\{(t_i, s_i); i = 1, \ldots, 4\}$ of example 1.

Solution. The optimal solution (ϵ^*, x^*) of example 1, with $m = 2$, involves the approximation error:

$$\begin{aligned} \sum_{i=1}^{4} \left| t_i - \sum_{j=1}^{2} x_j^* s_i^{j-1} \right| &= \|\epsilon^*\|_1 = t^T y^{(2)} \\ &= (\tfrac{1}{2}, 1, 2, 1)(-1, 1, 1, -1)^T = \tfrac{3}{2} \\ &> 1. \end{aligned}$$

Therefore $m > 2$.

Next, try $m = 3$, i.e., approximate the given data by:

$$t = x_1 + x_2 s + x_3 s^2.$$

This results in the dual problem:

maximize $\qquad \frac{1}{2} y_1 + y_2 + 2 y_3 + y_4 \qquad (25')$

subject to

$$\begin{aligned}
0 &\leqslant y_1 + y_2 + y_3 + y_4 \leqslant 0\\
0 &\leqslant y_2 + 2y_3 + 3y_4 \leqslant 0\\
0 &\leqslant y_2 + 4y_3 + 9y_4 \leqslant 0\\
-1 &\leqslant y_1 \leqslant 1\\
-1 &\leqslant y_2 \leqslant 1\\
-1 &\leqslant y_3 \leqslant 1\\
-1 &\leqslant y_4 \leqslant 1,
\end{aligned}$$

which is (25) with the additional constraint

$$0 \leqslant y_2 + 4y_3 + 9y_4 \leqslant 0. \tag{26'}$$

Since (26′) is violated by $y^{(2)T} = (-1, 1, 1, -1)$, we use it as the additional constraint (6.3) of the auxiliary problem in

Iteration 3.
(AP. 3).

maximize

$$\tfrac{1}{2}y_1 + y_2 + 2y_3 + y_4 \tag{4}$$

subject to

$$\left\{\begin{aligned}
-1 &\leqslant y_1 \leqslant 1\\
0 &\leqslant y_1 + y_2 + y_3 + y_4 \leqslant 0\\
-1 &\leqslant y_3 \leqslant 1\\
0 &\leqslant y_2 + 2y_3 + 3y_4 \leqslant 0
\end{aligned}\right. \tag{5.3}$$

$$0 \leqslant y_2 + 4y_3 + 9y_4 \leqslant 0. \tag{6.3}$$

(AP. 3) is obtained in the usual way from (AP. 2) of example 1.

Using:

$$y^{(2)T} = (-1, 1, 1, -1)$$

$$A^{(3)} = \begin{pmatrix} 1 & 0 & 0 & 0\\ 1 & 1 & 1 & 1\\ 0 & 0 & 1 & 0\\ 0 & 1 & 2 & 3 \end{pmatrix}, \qquad (A^{(3)-1} = \begin{pmatrix} 1 & 0 & 0 & 0\\ -\frac{3}{2} & \frac{3}{2} & -\frac{1}{2} & -\frac{1}{2}\\ 0 & 0 & 1 & 0\\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix}$$

$$a^{h(3)} = (0, 1, 4, 9), \qquad b^{h(3)-} = 0, \qquad b^{h(3)+} = 0$$

we compute:

$$\begin{aligned}
\Delta &= -4\\
t^T(A^{(3)})^{-1} &= (-\tfrac{1}{2}, 1, 1, 0)\\
a^{h(3)}(A^{(3)})^{-1} &= (3, -3, -1, 4)\\
Q &= \{k_1, k_2, k_3, k_4\} = \{4, 1, 2, 3\}\\
\delta_4 &= 0, \delta_1 = 2, \delta_2 = 0, \delta_3 = -2\\
p &= 2\\
\theta &= \tfrac{4}{3}
\end{aligned}$$

and by (16.3)

$$y^{(3)} = \begin{pmatrix} -1 \\ 1 \\ 1 \\ -1 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 & 0 \\ -\frac{3}{2} & \frac{3}{2} & -\frac{1}{2} & -\frac{1}{2} \\ 0 & 0 & 1 & 0 \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} (\tfrac{4}{3}) = \begin{pmatrix} \frac{1}{3} \\ -1 \\ 1 \\ -\frac{1}{3} \end{pmatrix}.$$

The approximation error corresponding to $y^{(3)}$ is

$$t^T y^{(3)} = (\tfrac{1}{2}, 1, 2, 1)(\tfrac{1}{3}, -1, 1, -\tfrac{1}{3})^T = \tfrac{5}{6}$$
$$< 1.$$

Therefore the minimal m satisfying (27) is 3.

To compute the optimal (ϵ^*, x^*), we form

$$A^{(4)} = \begin{pmatrix} 0 & 1 & 4 & 9 \\ 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 2 & 3 \end{pmatrix}$$

by replacing row 1 of $A^{(3)}$ (since $p = 2$, $k_p = 1$ in iteration 3) with $a^{h(3)}$. Now

$$(A^{(4)})^{-1} = \begin{pmatrix} \frac{1}{3} & 1 & \frac{1}{3} & -\frac{4}{3} \\ -\frac{1}{2} & 0 & -1 & \frac{3}{2} \\ 0 & 0 & 1 & 0 \\ \frac{1}{6} & 0 & -\frac{1}{3} & -\frac{1}{6} \end{pmatrix}$$

$$t^T (A^{(4)})^{-1} = (-\tfrac{1}{6}, \tfrac{1}{2}, \tfrac{5}{6}, \tfrac{2}{3})$$

and by (20.3)

$$x_1^* = \tfrac{1}{2} (= (t^T (A^{(4)})^{-1})_2 \qquad \text{since row 1 of } F^T = \text{row 2 of } A^{(4)})$$
$$x_2^* = \tfrac{2}{3}$$
$$x_3^* = -\tfrac{1}{6}.$$

An approximating polynomial of minimal degree, with approximation error $\leqslant 1$, is therefore:

$$t = \tfrac{1}{2} + \tfrac{2}{3}s - \tfrac{1}{6}s^2$$

which for the above data gives

$$\epsilon^{*T} = (0, 0, \tfrac{5}{6}, 0).$$

References

1. I. Barrodale, Approximation in the L_1 and L_∞ norms by linear programming, PhD. Thesis, Dept. of Computational and Statistical Science, University of Liverpool, 1967.
2. I. Barrodale, An improved algorithm for discrete L_1 linear approximations. *Numer. Math.* (forthcoming).

3. I. Barrodale and A. Young, Algorithms for best L_1 and L_∞ linear approximations on a discrete set. *Numer. Math.* **8** (1966), 295–306.
4. A. Ben-Israel and A. Charnes, An explicit solution of a special class of linear programming problems, *Operations Research* **16** (1968), 1166–1175.
5. A. Ben-Israel, A. Charnes, and P. D. Robers, "On generalized inverses and interval linear programming," Proceedings of the Symposium "Theory and applications of generalized inverses of matrices", Mathematics Series No. 4, pp. 53–70, Texas Technological College, Lubbock, Texas, March, 1968.
6. A. Charnes and W. W. Cooper, "Management Models and Industrial Applications of Linear Programming," Vols. I, II. Wiley, New York, 1961.
7. G. B. Dantzig, "Linear Programming and Extensions." Princeton, 1963.
8. C. E. Lemke, The dual method for solving the linear programming problem. *Naval Res. Logist. Quart.* **1** (1954), 36–47.
9. P. Rabinowitz, Applications of linear programming to numerical analysis. *SIAM Rev.* **10** (1968), 121–59.
10. P. D. Robers, Interval linear programming, Ph.D. Thesis, Dept. of Industrial Engineering and Management Sciences, Northwestern University, August 1968.
11. P. D. Robers and A. Ben-Israel, A suboptimization method for interval linear programming: A new method for linear programming, Systems Research Memo. No. 206, The Technological Institute, Northwestern University, June 1968.
12. E. L. Stiefel, Note on Jordan elimination, linear programming, and Tchebycheff approximation. *Numer. Math.* **2** (1960), 1–17.
13. E. L. Stiefel, Methods—old and new, for solving the Tchebycheff approximation problem. *J. Soc. Ind. Appl. Math. Ser. B. Numer. Anal.* **1** (1964), 164–176.
14. H. M. Wagner, Linear programming techniques for regression analysis. *J. Amer. Statist. Assoc.* **54** (1959), 206–212.
15. S. Zlobec and A. Ben-Israel, On explicit solutions of interval linear programs, Systems Research Memo. No. 210, The Technological Institute, Northwestern University, July 1968.