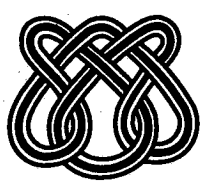
UNIVERSIDADE DE SÃO PAULO

PIECEWISE LINEAR PROGRAMMING VIA INTERIOR POINTS

MARIO CONRADO CAVICHIA
MARCOS NEREU ARENALES

Nº 38

NOTAS



Instituto de Ciências Matemáticas de São Carlos

Instituto de Ciências Matemáticas de São Carlos

ISSN - 0103-2577

PIECEWISE LINEAR PROGRAMMING VIA INTERIOR POINTS

MARIO CONRADO CAVICHIA
MARCOS NEREU ARENALES

Nº 38

NOTAS DO ICMSC
Série Computação

São Carlos
Abr./1998

Piecewise Linear Programming via Interior Points

Mario Conrado Cavichia
Departamento de Estruturas - Faculdade de Engenharia Civil
Universidade Estadual de Campinas-UNICAMP, C.P.6021
Campinas -S.P. - Brasil
e-mail: mcavichi@fec.unicamp.br

Marcos Nereu Arenales
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo-Campus de São Carlos, C.P. 668
13560.970-São Carlos - S.P. - Brasil
e-mail: arenales@icmsc.sc.usp.br

Abstract

The main aim of this work is to provide some basis for the development of algorithms of interior points to minimize piecewise linear objective functions. Specifically, we study a piecewise linear separable and convex objective function, subject to linear constraints. The available methods in the literature for this class of problem are of the Simplex type, except for specific cases, such as linear fitting in the sense of L_1-*norm*. A common practice for the resolution of piecewise linear programs consists of transforming them into equivalent linear programs and exploring their structure. This strategy is suitable for Simplex type methods, but inadequate for interior point methods. We show how to extend known interior point methods devised for linear programming to piecewise linear programming without resorting to equivalent linear programs. We also show that the generated interior points for the original piecewise linear program are not interior points for the equivalent linear program. Finally, some computational experiments are presented.

Keywords: Interior Point Methods, Piecewise Linear Programming

Programação Linear por Partes via Pontos Interiores

Mario Conrado Cavichia
Departamento de Estruturas - Faculdade de Engenharia Civil
Universidade Estadual de Campinas-UNICAMP, C.P.6021
Campinas -S.P. - Brasil
e-mail: mcavichi@fec.unicamp.br

Marcos Nereu Arenales
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo-Campus de São Carlos, C.P. 668
13560.970-São Carlos - S.P. - Brasil
e-mail: arenales@icmsc.sc.usp.br

Resumo

O objetivo principal deste trabalho consiste em fornecer bases para o desenvolvimento de algoritmos de pontos interiores para minimizar funções lineares por partes. Especificamente, estudamos os problema cuja função objetivo é linear por partes, separável e convexa, sujeito a restrições lineares. Os métodos disponíveis na literatura para esta classe de problemas são do tipo simplex, exceto para casos especiais, tais como o ajuste de curvas no senso da norma L_1. Uma prática comum para a resolução de programas lineares por partes consiste em transformá-los em programas lineares equivalentes e explorar suas estruturas. Esta estratégia é bem razoável para métodos do tipo simplex, mas é inadequada para métodos de pontos interiores. Mostramos como estender para o problema linear por partes métodos conhecidos de pontos interiores, inicialmente descritos para programas lineares, sem necessidade de recorrer ao programa linear equivalente. Mostramos também que os pontos interiores gerados para o problema original linear por partes não são interiores para o programa linear equivalente. Finalmente, apresentamos alguns experimentos computacionais.

Palavras-chave: Métodos de Pontos Interiores, Programação Linear por Partes.

1. Introduction

Since 1984, when Karmarkar[14] proposed an interior point method for linear programming a number of variant techniques has arisen (see Gonzaga[12], Hertog and Roos[13], Adler *et al.*[1]) and some effort has been made to unify the view of these techniques (Shanno and Bagchi[18]). More recently, the principles of the interior point method have been extended to non-linear programming (see Monteiro and Adler[15], Tapia[20]). Note that the techniques used to develop interior point methods, such as barrier functions, solution of non-linear systems due to the first order optimality conditions, Newton direction etc., were already well known.

In between linear and non-linear programming there is *Piecewise Linear Programming* (PLP), the focus of this paper, restricted to the special case where the objective function is separable and convex, subject to linear constraints. Golstein and Youdine [11], and more recently Fourer ([7], [8], [9]) studied this subject based on the simplex method. On the other hand, interior methods for piecewise linear programming are restricted to special problems such as L_1 and L_∞ solutions of overdetermined systems (see Zhang [21], Ruzinski [17], Sherali *et al.* [19]).

A common technique for developing specialized algorithms for piecewise linear programs consists of transforming the original problem into an equivalent linear program (causing a considerable increase in the number of variables) and then applying a linear programming method, and exploring the special structure created (see Barrodale and Young [4], Barrodale and Roberts [3], Zhang [21]). For Simplex type methods this strategy is quite appropriate, in the sense that in this way it is possible to obtain an equivalent method which consists of directly approaching the piecewise linear program, as will be shown in this paper. However, the application of interior point methods to the equivalent linear program cannot be specialized in order to obtain methods that directly approach the piecewise structure, since these produce *hybrid interior point methods* for the equivalent linear program, that is, the optimal solution in a vertex is achieved by the boundary in the equivalent program.

Problems of piecewise linear programming can be found in production planning (see Golstein and Youdine [11]), expansion of telephone network (see França *et al.* [10]) etc., and theoretically, in the computation of initial solutions to linear programming problems, approximations of non-linear problems(see Fourer [6]), fitting curves(see Armstrong *et al.* [2]), etc.

2. The Piecewise Linear Program (PLP)

2.1 The Problem Definition

Consider the following piecewise linear program:

$$\min f(x) = \sum_{j=1}^{n} f_j(x_j) \tag{1.1}$$

subject to: $Ax = b,\ 0 \le x \le u.$ (1.2)

where $A \in \mathcal{R}^{m \times n}$, with rank$(A) = m$, and each $f_j(x_j)$ is a convex piecewise linear function given by:

$$f_j(x_j) = \begin{cases} c_j^1 (x_j - \beta_j^0) + f_j(\beta_j^0) & \beta_j^o \le x_j \le \beta_j^1 \\ c_j^2 (x_j - \beta_j^1) + f_j(\beta_j^1), & \beta_j^1 \le x_j \le \beta_j^2 \\ \vdots & \\ c_j^{m_j} (x_j - \beta_j^{m_j - 1}) + f_j(\beta_j^{m_j - 1}), & \beta_j^{m_j - 1} \le x_j \le \beta_j^{m_j} \end{cases} \tag{2}$$

with $c_j^i < c_j^{i+1}$, $\beta_j^i < \beta_j^{i+1}$, $i = 1, \cdots, m_j - 1$, $j = 1, \cdots, n$.

The scalars β_j^i are non-differentiability points of f_j called *break points*. To make easy the notation it is assumed that:

$$\beta_j^0 = 0 \quad and \quad \beta_j^{m_j} = u_j .$$

2.2. An Example

To illustrate, consider $f(x_1,x_2) = f_1(x_1) + f_2(x_2)$ given by:

$$f_1(x_1) = \begin{cases} -3x_1 + 9, & 0 \le x_1 \le 1 \\ -2x_1 + 8, & 1 \le x_1 \le 2 \\ 4, & 2 \le x_1 \le 3 \\ x_1 + 1, & 3 \le x_1 \le 4 \end{cases} \qquad f_2(x_2) = \begin{cases} -3x_2 + 3, & 0 \le x_2 \le 1 \\ -2x_2 + 2, & 1 \le x_2 \le 2 \\ x_2 - 4, & 2 \le x_2 \le 3 \\ 2x_2 - 7, & 3 \le x_2 \le 4 \end{cases}$$

whose graphic of which is showed in Figure 1.

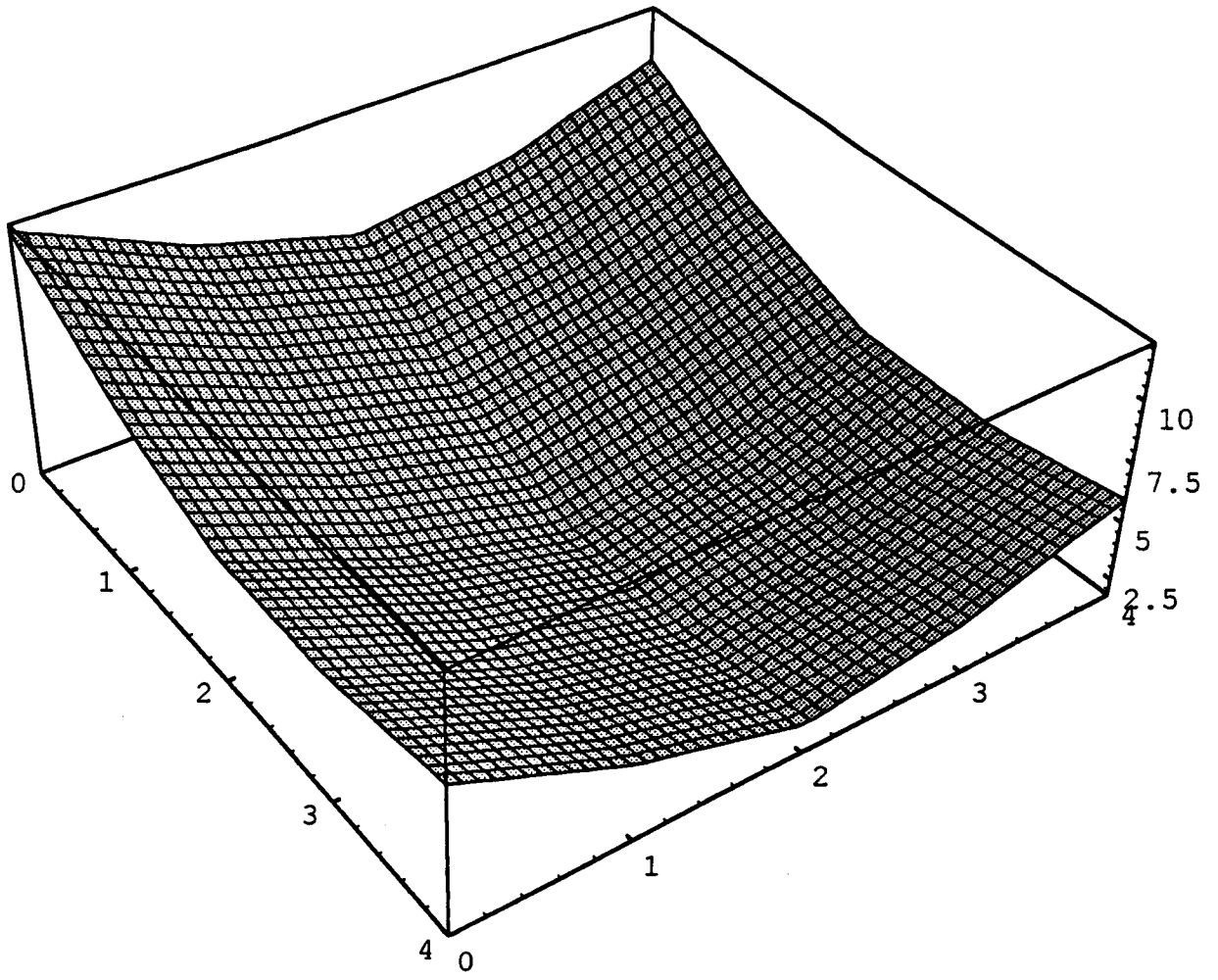


Figure 1.

Also consider the linear constraints:

$$\begin{cases} -x_1 + x_2 \le 2 \\ x_1 + 3x_2 \le 14 \\ 3x_1 - x_2 \le 12 \end{cases}$$

$$x_1 \ge 0,\ x_2 \ge 0.$$

Figure 2 shows the feasible region (delimited by bold lines) and the regions where the objective function f(x) is linear (delimited by doted lines).

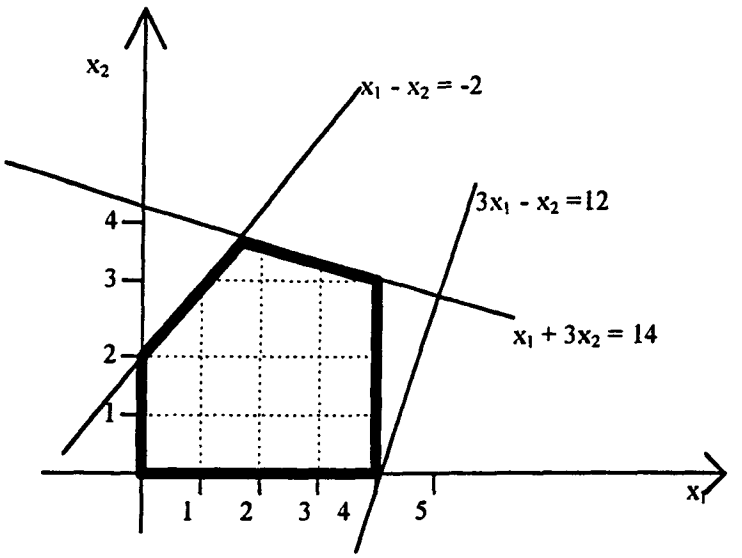


Figure 2

Note that the solution can be found on a vertex, or belonging to an edge, or in the interior of the feasible region, as happens in non-linear programming problems. However, if in considering the non-differentiability lines as *virtual boundaries*, an optimal solution will occur on a vertex. The following definition characterizes these points, which, by analogy which linear programming, will be called basic solutions.

2.3 Basic Elements

Let $\mathbf{A}=[\mathbf{a}_1, \mathbf{a}_2,..., \mathbf{a}_n]$ be the matrix of coefficients and $\{B_1,...B_m\} \subset \{1,...,n\}$ be a subset of the indices of the columns of matrix **A**, such that $\mathbf{B}=[\mathbf{a}_{B_1},...,\mathbf{a}_{B_m}]$ is a non-singular matrix. The complementary indices are denoted by $N_1,...,N_{n-m}$ which define the matrix $\mathbf{N}=[\mathbf{a}_{N_1},...,\mathbf{a}_{N_{n-m}}]$. The variables x_{B_i}, $i=1,...,m$ are called basic and x_{N_j}, $j=1,...,n-m$ are called non-basic and they define the vectors $\mathbf{x_B}$ and $\mathbf{x_N}$ respectively. The partition in the columns of **A=[B,N]** is called a basic partition.

Definition 1. Consider a basic partition of **A**. Then the solution $\mathbf{x}=(\mathbf{x_B},\mathbf{x_N})$, such that:

- $x_{N_j} = \beta^i_{N_j}$ for some $i = 0,1,..., m_{N_j}$ and $j=1,...,n-m$, and
- $\mathbf{x_B} = \mathbf{B}^{-1}(\mathbf{b} - \mathbf{N}\mathbf{x_N})$,

is called a *basic solution to the Piecewise Linear Program*, or simply, a *basic solution*. If, in addition: $\mathbf{0} \le \mathbf{x_B} \le \mathbf{u_B}$, then $\mathbf{x}$ is a *feasible basic solution*. Moreover, if $x_{B_k} = \beta^i_{B_k}$, for some i, $\mathbf{x}$ is called a *degenerated basic solution*.

As has already been noted, a feasible basic solution is not necessarily a vertex of the feasible region. In fact, in the absence of degeneracy, a feasible basic solution is a vertex if and only if $x_{N_j} = \beta^0_{N_j}$ *or* $\beta^{m_j}_{N_j}$, $j = 1,2,\ldots,,n-m$.

Theorem 1. If the Piecewise Linear Program has an optimal solution, then there exists an optimal feasible basic solution.

The steps to prove Theorem 1 are the same used to prove its well-known linear version. We say that a (primal) simplex method is devised for PLP if it reduces the search for feasible basic solutions according to Definition 1. Next, we provide a common strategy to deal with PLP.

3. Equivalent Linear Program

3.1. The Equivalent Problem

In this section a linear programming problem equivalent to Problem (1) is presented. For this, each original variable is written as:

$$x_j = \sum_{i=1}^{m_j} x_j^i \,, \qquad with \qquad 0 \le x_j^i \le b_j^i - b_j^{i-1}, \quad i = 1,\cdots,m_j, \quad (3)$$

and x_j^i is called a *component* of x_j. In addition, we impose the following condition:

$$x_j \in (\beta_j^k,\ \beta_j^{k+1}) \Leftrightarrow \begin{cases} x_j^i = \beta_j^{i+1} - \beta_j^i, \ i < k-1, \\ x_j^k = x_j - \beta_j^k \in (0,\ \beta_j^{k+1} - \beta_j^k), \\ x_j^i = 0, \ i > k+1. \end{cases} \quad (4)$$

Condition (4) simply defines an one-to-one relationship among a variable x_j and its components $x_j^1, x_j^2, \ldots, x_j^{m_j}$. For example in section 2.2, $x_1 = x_1^1 + x_1^2 + x_1^3 + x_1^4$, $0 \le x_1^i \le 1$, and $x_1=2.5$ can only be written in terms of its components by: $x_1^1 = 1$, $x_1^2 = 1$, $x_1^3 = 0.5$, $x_1^4 = 0$.

Therefore, Problem (1) is equivalent to:

$$min \sum_{j=1}^{n} \sum_{i=1}^{m_j} c_j^i x_j^i \tag{5.1}$$

$$subject\ to: \sum_{j=1}^{n} \sum_{i=1}^{m_j} a_j x_j^i = \mathbf{b}, \tag{5.2}$$

$$0 \le x_j^i \le \beta_j^i - \beta_j^{i-1},\ i = 1,\ldots,m_j,\ j = 1,\ldots,n. \tag{5.3}$$

Problem (5) has m restrictions and $\sum_{j=1}^{n} m_j$ variables, and the constraint matrix has the form:

$$\tilde{\mathbf{A}} = [\mathbf{a}_1\ \mathbf{a}_1 \ldots \mathbf{a}_1\ \mathbf{a}_2\ \mathbf{a}_2 \ldots \mathbf{a}_2 \ldots \mathbf{a}_n\ \mathbf{a}_n \ldots \mathbf{a}_n]$$

with m_j columns formed by $\mathbf{a}_j$, $j = 1,2,\ldots,n$, and a solution is given by:

$$\tilde{x} = (x_1^1, x_1^2 \cdots x_1^{m_1}, x_2^1, x_2^2 \cdots x_2^{m_2}, x_n^1 \cdots x_n^{m_n}).$$

Next, some properties of an optimal solution to Problem (5) are presented.

3.2. Some Properties

Theorem 2. Let $\tilde{\mathbf{x}} = \left(x_1^1, x_1^2, \ldots, x_1^{m_1}, x_2^1, x_2^2, \ldots, x_2^{m_2}, x_n^1, \ldots, x_n^{m_n}\right)$ be a non-degenerated basic optimal solution to Problem (5). Then:

- If x_j^k is a basic variable, then $x_j^1, x_j^2, \ldots, x_j^{k-1}$ *and* are necessarily non-basic, with $x_j^i = \beta_j^{i+1} - \beta_j^i$, $i \le k-1$ *e* $x_j^i = 0$, $i \ge k+1$;
- If $x_j^1 = 0$ then $x_j^2 = \cdots = x_k^{m_j} = 0$ (i.e., if x_j^1 is a non-basic variable at its lower bound, then x_j^i, for each i, is also a non-basic variable at its lower bound);

- If $x_j^{m_j} = \beta_j^{m_j} - \beta_j^{m_j-1}$, then $x_j^i = \beta_j^i - \beta_j^{i-1}$, $i = 1, \cdots, m_j - 1$ (i.e., if $x_j^{m_j}$ is a non-basic variable at its upper bound, then x_j^i, for each i, is also a non-basic variable at its upper bound).

Proof: Since x_j^k is a basic variable, then $c_j^k - \pi^t a_j = 0$, where π is the simplex multiplier vector. For $i = 0, 1, \ldots k-1$, by hypothesis $c_j^i < c_j^k$, hence $c_j^i - \pi^t a_j < c_j^k - \pi^t a_j$, which results in $c_j^i - \pi^t a_j < 0$. Then, $x_j^i = \beta_j^{i+1} - \beta_j^i$. Otherwise, a better solution could be achieved by a simplex method step, since the solution $\bar{x}$ is non-degenerated. Furthermore, for $i = k+1, \ldots, m_j$, $c_j^i > c_j^k$ implies $c_j^i - \pi^t a_j > 0$. Then, $x_j^i = 0$. Otherwise $\bar{x}$ would not be optimal. The arguments for the other parts are similar. ■

The Simplex Method can be specialized for Problem (5), searching only for basic solutions that verify the necessary conditions in Theorem 1. For example, if x_j^k is a basic variable that achieves its upper bound indicating a basis exchange, it can be changed by x_j^{k+1} without changing the basis (because both variables have the same column $\mathbf{a_j}$). In the original problem (1) this just means that x_j crosses the break-point β_j^k. Figure 3 illustrates these possibilities: when x_j^k is increasing, x_j^{k+1} remains at its lower bound as a non-basic variable. After x_j^k achieves its upper bound, it becomes a non-basic variable at its upper bound, and x_j^{k+1} is increased.

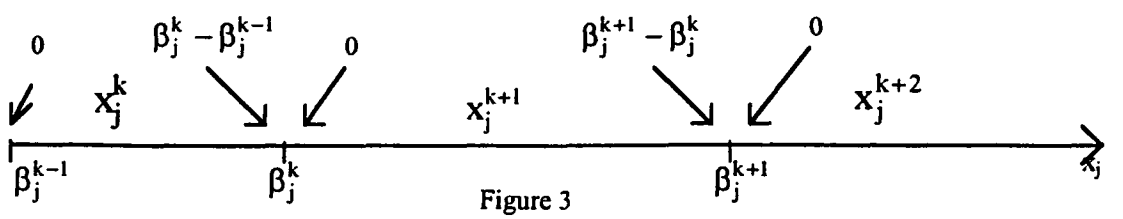


Figure 3

As the cost of x_j^{k+1} is different from the cost of x_j^k, it follows that the relative costs of non-basic variables have to be updated (simple formulas can be easily developed). For the

original problem (1), this means that the moving direction is not changed, but a non-differentiability point has been crossed. Meanwhile, for the equivalent linear program (5), the moving direction is changed in the following way: suppose, for example, that x_j^1 is a basic variable and reaches its upper bound, so it is changed by x_j^2 in the basis. Although the basis has not been changed, the moving direction to Problem (5), that before was: $\underbrace{(d_1,0,\cdots,0,}_{x_1 \text{ comp.}}...)$, becomes: $\underbrace{(0,d_1,\cdots,0,}_{x_1 \text{ comp.}}...)$, where d_1 is the coordinate of the Simplex direction. This specialization of the Simplex method is equivalent to the simplex method directly developed for Problem (1), searching only for basic solutions according to Definition 1. These steps were performed by Barrodale and Roberts[2] skipping some basis exchange of the Simplex Method for L_1 Linear Approximation Problem. We shall see that this strategy of improving a method from Linear Programming is not possible for the interior point methods.

Theorem 3. Let $\tilde{x} = (x_1^1, x_1^2 \cdots x_1^{m_1}, x_2^1, x_2^2 \cdots x_2^{m_2}, x_n^1 \cdots x_n^{m_n})$ be an optimal solution to Problem (5) and **x** be computed according to (3). Then Condition (4) holds.

Proof. Suppose that (4) is not satisfied. Then there are only three possibilities:

i) There exists a variable x_j with components x_j^r and x_j^s such as

$$x_j^r \in (0,\ \beta_j^{r+1} - \beta_j^r) \text{ and } x_j^s \in (0,\ \beta_j^{s+1} - \beta_j^s).$$

ii) There exists a variable x_j with components

$$x_j^k \in (0,\ \beta_j^{k+1} - \beta_j^k) \text{ and } x_j^r = 0, \text{ for } r < k.$$

iii) There exists a variable x_j with components

$$x_j^k \in (0,\ \beta_j^{k+1} - \beta_j^k) \text{ and } x_j^s = \beta_j^{s+1} - \beta_j^s, \text{ for } s > k.$$

If i) occurs, then define $\tilde{\mathbf{d}}$ such that $d_j^r = -d_j^s = 1$ and all other components are equal to zero. It is easy to see that $\tilde{\mathbf{A}}\tilde{\mathbf{d}} = \mathbf{0}$, and consequently $\tilde{\mathbf{d}}$ is a feasible direction at $\tilde{\mathbf{x}}$, because x_j^r and x_j^s are not at their bounds. Now, let $\tilde{\mathbf{c}}$ be the gradient of the objective function of Problem (5). So, assuming $s > r$, without loss of generality, it follows that: $\tilde{c}^t \tilde{d} = c_j^r - c_j^s < 0$.

Hence, $\tilde{\mathbf{d}}$ is a feasible descent direction at $\tilde{\mathbf{x}}$, which is a contradiction, since $\tilde{\mathbf{x}}$ is optimal.

If ii) holds, consider $\tilde{\mathbf{d}}$ such that $d_j^k = -d_j^r = -1$ and all other components are equal to zero.

As $\tilde{A}\tilde{d} = 0$ and $d_j^r = 1$ with $x_j^r = 0$ the chosen direction is feasible at $\tilde{\mathbf{x}}$. Furthermore, $\tilde{c}^t \tilde{d} = -c_j^k + c_j^r < 0$, contradicting the optimality of $\tilde{\mathbf{x}}$. A similar proof follows if iii) holds. ■

A corollary of Theorem 3 is that Condition (4) can be abandoned without loss of generality.

In the following section an algorithm will be developed to solve Problem (1) without transforming it into a linear problem nor making use of some variant of the simplex method.

4. A Piecewise Linear Interior Point Method

4.1 General Principles

Definition 2: We say $\mathbf{x} \in \Re^n$ is a *piecewise linear interior point* if:

- $Ax = b$, and
- $\beta_j^{i_j - 1} < x_j < \beta_j^{i_j}$, for some i_j, $1 \le i_j \le m_j$ and $j = 1,2,\ldots,n$.

Let x^k be a piecewise linear interior point, and i_j the indices such that $\beta_j^{i_j - 1} \le x_j^k \le \beta_j^{i_j}$, $1 \le j \le n$. Consider now the problem with the constraints: $\beta_j^{i_j - 1} \le x_j \le \beta_j^{i_j}$. These constraints define a kind of *virtual boundary*, under which we have a linear program. As $\mathbf{x}^k$ belongs to the interior of this feasible sub-region, a moving direction of any interior point method from linear programming can be used. After the moving direction has been determined we forget the virtual boundary and proceed with a piecewise linear line search in order to obtain the minimum of the objective function in this direction. Note that, as long as one moves in the chosen direction, non-differentiability regions can be crossed. This strategy can be used for other piecewise linear programs, such as the minmax problem. To compute the moving direction we used the algorithm developed by Sherali *et al.* [19].

4.2 The Moving Direction

Let x^k be a piecewise linear interior point and i_j, $j = 1,2,...,n$, according to Definition 2. Consider the following linear program:

$$\min f(x) = c^t x, \tag{6.1}$$

$$s.a.\ Ax = b \tag{6.2}$$

$$\beta_j^{i_j-1} < x_j < \beta_j^{i_j},\ j = 1,2,...,n, \tag{6.3}$$

where

$$c^k = \nabla f(x^k) = (c_1^{i_1}, c_2^{i_2}, ..., c_n^{i_n})^t.$$

Note that the standard problem used by Sherali *et al.*, with $-1 \le x_j \le 1$, $j=1,...,n$, can be obtained with the changing variable:

$$x_i \leftarrow \frac{2x_i - \left(\beta_j^{i_j-1} + \beta_j^{i_j}\right)}{\left(\beta_j^{i_j} - \beta_j^{i_j-1}\right)},\ j = 1,2,...,n.$$

In order to keep the points out of the boundaries (which may not be a real boundary of the original problem), consider the barrier function as follows:

$$\min (c^k)^t x - \mu \sum_{j=1}^{n} ln\left(\beta_j^{i_j} - x_j\right) - \mu \sum_{i=1}^{n} ln\left(x_j - \beta_j^{i_j-1}\right) \tag{7.1}$$

$$s.t.\quad Ax = b. \tag{7.2}$$

According to Shanno and Bagchi[18] most interior point methods can be constructed by this strategy, producing a search direction consisting of a linear combination of a descent direction and a centering direction.

The first order optimality conditions of Problem (7) produces:

- $c - A^t\lambda - \mu D^{-1} v(x) = 0$ (8.1)
- $Ax = b$, (8.2)

with

$D_k = diag(d_j)$, where $d_j = min\left\{\beta_j^{i_j} - x_j^k, x_j^k - \beta_j^{i_j-1}\right\}$

and

$$v_j\left(x_j^k\right) = \frac{d_j\left(\beta_j^{i_j} + \beta_j^{i_j-1} - 2x_j^k\right)}{\left(x_j^k - \beta_j^{i_j}\right)\left(x_j^k - \beta_j^{i_j-1}\right)}, \quad j = 1,2,\ldots,n.$$

In the method of Sherali et al.[19], the usual strategy is not adopted, which consists of applying a Newton method step to the non-linear system (8). By considering a feasible primal solution $\mathbf{x}^k$, the linear system (8.1) rests on variable λ for which the least square solution is determined:

$$\lambda^k = \left(AD_k^2A^t\right)^{-1} AD_k\left(D_kc^k - \mu v\left(x^k\right)\right),$$

which is called a *dual estimate*.

Note that the gradient of the Lagrangian function of Problem (7) is

$$g^k = c^k - A^t\lambda^k - \mu D_k^{-1}v\left(x^k\right)$$

$$D_kg^k = \left[I - \left(AD_k\right)^t\left(AD_k^2A^t\right)^{-1}AD_k\right]\left[D_kc^k - \mu v\left(x^k\right)\right] = P_k\left[D_kc^k - \mu v(x^k)\right],$$

where $P_k = \left[I - \left(AD_k\right)^t\left(AD_k^2A^t\right)^{-1}AD_k\right]$ is the projection matrix onto the null space of AD_k. Thus, D_kg^k is the scaling projection of the gradient of the objective function (7.1) onto the null space of AD_k. Therefore

$$d^k = -D_kg^k \tag{9}$$

is the feasible descent direction to be chosen at $\mathbf{x}^k$, and the new solution is computed by:

$$x^{k+1} = x^k + \varepsilon_k d^k,$$

where ε_k is an appropriate steplength.

Sherali *et al.* [19] used as termination criterion: $\left\|d^k\right\| < Tol$, where $Tol > 0$ and sufficiently small. Although the problem we are dealing with does not have the bounds (6.3), this termination has been successfully employed.

4.3 A Piecewise Linear Line Search Procedure

The Piecewise Linear Line Search is an important procedure in the algorithm since it allows a rapid decrease in the objective function, on the first iterates.

Consider $\mathbf{d}^k$ a feasible descent direction at an interior point $\mathbf{x}^k$. An intermediary sequence $y^0, y^1, \ldots$ will be generated.

Step 0: Let $y^0 = x^k$ *and* $r=0$.

Step 1: For $j=1,\ldots,n$, *compute:*

$$\alpha_j = \begin{cases} \dfrac{\beta_j^{i_j} - y_j^r}{d_j^k}, & \text{if } d_j^k > 0, \\ \dfrac{\beta_j^{i_j-1} - y_j^r}{d_j^k}, & \text{if } d_j^k < 0, \\ \infty, & \text{if } d_j^k = 0. \end{cases}$$

and determine the index j^*, *such that:* $\alpha_{j^*} = min\{\alpha_j,\ j = 1,\cdots,n\}$.

Step 2: For $y=y^r+\varepsilon d^k$, *with* ε *slightly greater than* α_{j^*} *the gradient of* f *changes according to the following:*

$$\nabla f(y) = \begin{cases} \nabla f(y^r) + \left(c_{j^*}^{i_{j^*}+1} - c_{j^*}^{i_{j^*}} \right) e_{j^*}, & \text{if } d_{j^*}^k > 0 \\ \nabla f(y^r) + \left(c_{j^*}^{i_{j^*}-1} - c_{j^*}^{i_{j^*}} \right) e_{j^*}, & \text{if } d_{j^*}^k < 0 \end{cases}$$

where e_j *is* j^{th} *row of the identity matrix.*

Step 3: If ($\nabla f(y)d^k \geq 0$) or ($i_{j^*} = 0$ or $i_{j^*} = m_{j^*}$, i.e., a real boundary):

3.1. Then: $x^{k+1} = y^r + \delta_r d^k$, with $\delta_r = 0.985\ \alpha_{j^*}$.. END.

{The piecewise linear line search is finished with the steplength $\varepsilon_k = \delta_0 + \ldots + \delta_r$, and a new moving direction d^{k+1} should be calculated}.

3.2 Else: $y^{r+1} = y^r + (1.002\alpha_{j^*})\, d^k$,

$$i_{j^*} \leftarrow \begin{cases} i_{j^*} + 1, & \text{if } d_{j^*} > 0, \\ i_{j^*} - 1, & \text{if } d_{j^*} < 0. \end{cases}$$

{This updates the indices of y^{r+1} according to Definition 2}.

$r \leftarrow r+1$ and repeat Step 1.

Remarks: Sherali *et al.* [19] have proven that the parameter in Step 3.1 for avoiding the boundary (i.e., the number 0.985) can be chosen in the interval [0.97, 0.99] in order to guarantee the convergence of the sequence $\{x^k\}$. The steplength $(1.002\alpha_{j^*})$ for the intermediary sequence in Step 3.2 was chosen based on computational experiments. The generated sequence of interior points will coincide with that generated by the algorithm of Sherali *et al.*, only if no virtual boundary is crossed (i.e., r=0, for all k). This occurs at the tail end of the sequence generated by the proposed algorithm.

Next, an interior point algorithm for solving proposed the piecewise linear program is outlined..

4.4 The Piecewise Linear Interior Point Algorithm

Step 0: Consider an initial interior point $\mathbf{x}^0$, with $\beta_j^{i_j - 1} < x_j^k < \beta_j^{i_j}$, j=1,...,n. Let k = 0.

Step 1: The Moving Direction

Let $D_k = \left\{d_1^k, d_2^k, \ldots, d_n^k\right\}$, where $d_j^k = \min\left\{\beta_j^{i_j} - x_j^k, x_j^k - \beta_j^{i_j - 1}\right\}$. Compute:

$$\lambda^k = \left(AD_k^2 A^t\right)^{-1} AD_k\left(D_k c^k - \mu v\left(x^k\right)\right) \{\textit{dual estimate}\}$$

$$\mathbf{g_k} = \mathbf{c^k} - \mathbf{A^t}\lambda_\mathbf{k} - \mu\mathbf{D_k^{-1}}\mathbf{v}(\mathbf{x_k}) \{\textit{Lagrangian gradient}\}$$

$$d^k = -D_k g^k \{\textit{search direction}\}$$

Step 2: Stopping Criterion

If $\left\| d^k \right\| < \varepsilon$ then STOP. {$\mathbf{x}^k$ solves the problem}.

Step 3. *The Steplength*:

Apply the Piecewise Linear Line Search (Section 4.3) to determine:

$x^{k+1} = x^k + \varepsilon_k d^k$, $k = k+1$. Repeat Step 1.

4.5. How the method works on the equivalent linear program

As already mentioned a method of linear programming can be used to solve piecewise linear programming by considering its equivalent linear version, Problem (5). Problem (5) has some particularities that allow convenient simplifications, such as skipping some basis exchange in the simplex method, that correspond to a piecewise linear line search.

If an interior method is applied to the Problem (5), it is possible to identify, through Transformation (3), the behavior in the original problem (1).

Consider, in the transformed space, a feasible descent direction $\tilde{\mathbf{d}}$, feasible at $\tilde{\mathbf{x}}^{\mathbf{k}}$, obtained by any method:

$$\tilde{d} = \left(d_1^1 \ldots d_1^{m_1} d_2^1 \ldots d_2^{m_2} \ldots d_n^1 \ldots d_n^{m_n}\right)^t . \qquad (10)$$

The next iterate will be:

$\tilde{x}^{k+1} = \tilde{x}^k + \alpha \tilde{d}$, that is: $\left(x_j^i\right)^{k+1} = \left(x_j^i\right)^k + \alpha d_j^i$, $i=1,\ldots,m_j$, $j=1,\ldots,n$.

By definition:

$$x_j^{k+1} = \sum_{i=1}^{m_j} \left(x_j^i\right)^{k+1} ,$$

and so, in the original problem we have:

$$x_j^{k+1} = \left[\sum_{i=1}^{m_j} (x_j^i)^k\right] + \alpha \left[\sum_{i=1}^{m_j} d_j^i\right] = x_j^k + \alpha d_j ,$$

where

$$d_j = \sum_{i=1}^{m_j} d_j^i \qquad (11)$$

is the j^{th} coordinate of the direction **d** in the original space. This shows that any method applied to the transformed problem is well determined in the original problem (the reverse is not true). Moreover, note that an interior point in Problem (5) satisfies:

$$0 < x_j^i < \beta_j^{i+1} - \beta_j^i,$$

for all i and j, and therefore, Condition (4) will never hold, although knowing *a priori* that most of the components x_j^i in the optimal solution will be at their lower or upper bounds. Actually, an optimal basic solution has at most one component out of its bound. This direct way of extending an LP interior point method for PLP (i.e., using Problem (5)) seems to be inadequate, causing a great number of iterates when near an optimal solution (Graphic 8 illustrates this drawback). This occurs because the virtual frontiers in the original problem become true frontiers in the equivalent problem.

However, a hybrid interior point method for Problem (5) can be developed satisfying Condition (4). For this, consider a piecewise linear interior solution, **x** according to Definition (2), which defines indices i_j such that:

$$\beta_j^{i_j-1} < x_j < \beta_j^{i_j}, \; j = 1,2,\ldots,n,$$

and the only corresponding solution, $\tilde{x}$, in the transformed problem which satisfies Condition (4) is given by: For $j=1,\ldots,n$,

$$\begin{aligned} x_j^i &= \beta_j^i - \beta_j^{i-1}, && i = 1,\ldots,i_j - 1, \\ x_j^i &= 0, && i = i_j + 1,\ldots,n, \\ x_j^{i_j} &= x_j - \beta_j^{i_j-1} && \in \left(0, \beta_j^{i_j} - \beta_j^{i_j-1}\right). \end{aligned} \qquad (12)$$

It should be emphasized that although **x** is an interior point of the original problem, $\tilde{x}$ is not an interior point of Problem (5). Any feasible direction $\mathbf{d} = (d_1, d_2, \ldots, d_n)^t$ at **x** (original problem) has a corresponding feasible direction at $\tilde{x}$ (transformed problem according to (12)), given by:

$$d_j^i = \begin{cases} 0, & i \neq i_j, \\ d_j, & i = i_j. \end{cases} \qquad (13)$$

It follows that:

Theorem 4: If **d** is a feasible descent direction at **x**, then $\tilde{\mathbf{d}}$, given by (13), will be a feasible descent direction at $\tilde{x}$, given by (12).

Moving in the direction $\tilde{\mathbf{d}}$, only n components $x_j^{i_j}$, $j=1,...,n$ are changed, causing a component to reach one of its bound. For example, suppose that $x_1^1 \in \left(0, \beta_1^1 - \beta_1^0\right)$, and x_1^1 reaches its upper bound before the other components ($\tilde{\mathbf{d}} = (\underbrace{d_1\, 0\, 0...0}_{comp.\, x_1}, ...)$, $d_1>0$). Then the maximum step is given by:

$$\varepsilon_1 = ((\beta_1^1 - \beta_1^0) - x_1^1)/d_1^1,$$

producing a new feasible solution, $\tilde{x} \leftarrow \tilde{x} + \varepsilon_1 \tilde{d}$, in which the direction $\tilde{d}$ is no longer feasible, although in the original problem it can still be feasible and descending. In this case, it is advantageous to cross break-points (i.e., piecewise linear line search). It is worth noting that the most expensive step in optimization methods consists of computing feasible directions. To cross break-points corresponds, in the equivalent linear program, to changing the direction $\tilde{d}$ by:

$$\tilde{d} \leftarrow (\underbrace{0\, d_1 0...0}_{comp.\, x_1}, ...).$$

The new direction with $i_j \leftarrow i_j + 1$ is feasible at the new point, and Theorem 4 holds. The process is repeated, finding a new step, and so on.

In other words, the method presented corresponds to working on the transformed problem with the solutions according to (12), feasible directions according to (13), and changes of feasible directions in agreement with the procedure explained above. Thus, a method of interior points applied to the equivalent linear problem does not produce the interior piecewise linear method developed, because an interior solution in PLP can be seen in the equivalent problem as admitting x_j^i components at their bounds (and therefore at the frontier).

6. Computational Experiments

In this section we present some simple examples to illustrate the performance of the proposed algorithm. We consider the following situations:

- the only optimal solution is on a vertex (Example 1),
- the only optimal solution is interior (Example 2),
- multiple optimal solutions (Examples 3 and 4). and
- a convex and separable nonlinear program is approximated by a piecewise linear program and the number of breakpoints is increased (Example 5).

For all these instances, the Phase-1 problem (i.e., how to determine the initial interior point) was overcome by including penalty functions in their objective functions, allowing unfeasible initial solutions (see, for example, Table 3). These penalty functions are piecewise linear functions with only one breakpoint; for example, in the case of inequalities, the slack variables have high costs if they are negative, otherwise the cost is zero. The initial solutions are randomly generated.

The Stopping Criterion was: $\|d^k\| < \varepsilon$, $\varepsilon = 10^{-10}$, because, in this kind of algorithm, the convergence proof requires: $\lim_{k\to\infty} \|d^k\| = 0$.

Example 1. Consider the piecewise linear objective function given by: $f(x) = f_1(x_1) + f_2(x_2)$ with

$$f_1(x_1) = \begin{cases} -3x_1 + 9 & 0 \le x_1 \le 1 \\ -2x_1 + 8 & 1 \le x_1 \le 2 \\ -0.5x_1 + 5 & 2 \le x_1 \le 3 \\ x_1 + 0.5 & 3 \le x_1 \le 4 \end{cases} \qquad f_2(x_2) = \begin{cases} -3x_2 + 3 & 0 \le x_2 \le 1 \\ -2x_2 + 2 & 1 \le x_2 \le 2 \\ x_2 - 4 & 2 \le x_2 \le 3 \\ 2x_2 - 7 & 3 \le x_2 \le 4 \end{cases}$$

and the restrictions given by:

$$\begin{cases} -x_1 + x_2 \le 2 \\ 2x_1 + x_2 \le 8 \\ 2x_1 - x_2 \le 4 \end{cases}$$
$$x_1 \ge 0,\ x_2 \ge 0.$$

Figure 4 depicts the trajectories of the generated sequence of the algorithm, by initiating it with six different solutions, in a case where the only optimal solution is on a vertex of the feasibility region.

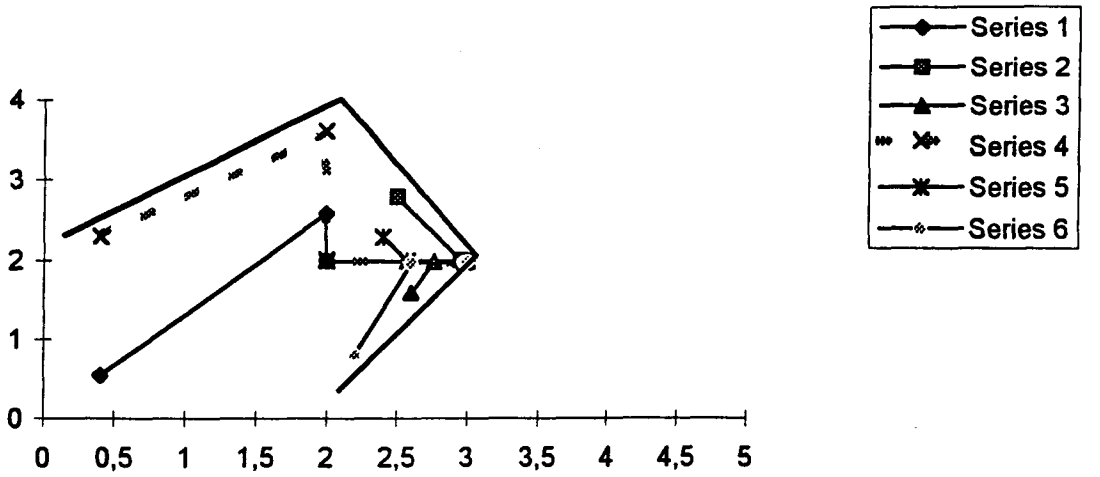


Figure 4

The number of intermediary iterations made in each main iteration for the series 1, 3, 4 and 6 (represented by $r+1$ in the procedure of Section 4) are shown in Figure 5. As can be observed in the graph obtained from Series 1, the algorithm performs four intermediary iterations in the first main iteration, and three non-differentiability regions are crossed in the piecewise linear search. However, in the second iteration, it does not abandon the face; in the third, one face is changed, and then converges on the optimal solution from the last visited face. Series 4 shows that with the first calculated direction, the algorithm visits three faces, changes direction and moves to two other faces. Then the direction is updated again, but no non-differentiability line is crossed. A new direction updating is made, in which three faces are visited, and then a face is reached from which the optimal solution is achieved.

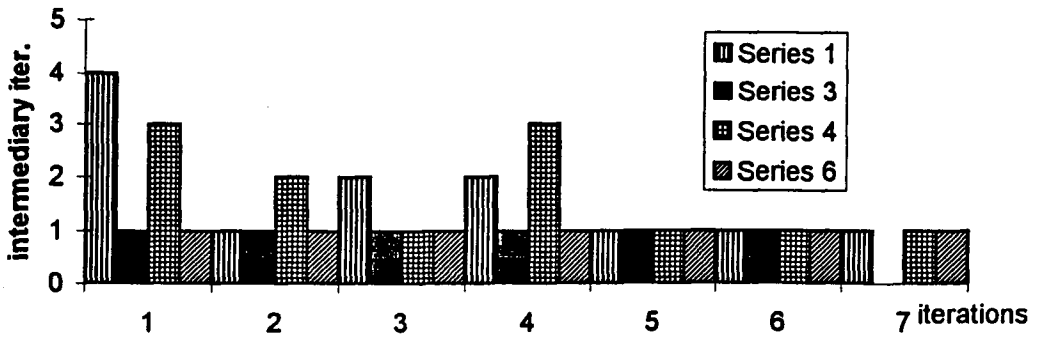


Figure 5

To illustrate, the results of Series 1 are shown in Table 1, indicating the number of intermediary iterations in each main iteration, that is, the way the piecewise linear line search behaves. Note that after the fifth iteration, since there is no more changing in the linearity region, it behaves just like the method in Sherali *et al.*.

Iter	x1	x2	iter	x1	x2
0	0.40400000000	0.54400000000	5	2.99968403888	1.99976528818
	0.75835422481	1.00000009120	6	2.99977217683	1.99999647932
	1.00000004833	1.31096158072	7	2.99999516864	1.99999711917
	1.53544670387	2.00000013781	8	2.99999635158	1.99999995679
1	1.99303170056	2.58884256049	9	2.99999635158	1.99999995679
2	1.99343458577	2.00883263841	10	2.99999992700	1.99999996281
	2.00000000131	2.00289108208	11	2.99999994435	1.99999999944
	2.00314672532	2.00004336623	12	2.99999999892	1.99999999951
	2.11731018435	1.99999999999	13	2.99999999916	1.99999999999
4	2.98659454917	1.99966979281	14	2.99999999998	1.99999999999

Table 1

Table 2 exhibits how the moving direction of Series 1 behaves in the first nine iterations.

1st iteration	2nd iteration	3rd iteration	4th iteration	5th iteration
d[1]=1.15E-1	d[1]=2.25E-2	d[1]=6.29E-6	d[1]=9.07E-9	d[1]=8.81E-10
d[2]=-1.94E-1	d[2]=3.31E-2	d[2]=9.43E-6	d[2]=2.12E-9	d2]=1.31E-9
d[3]=-7.36E-1	d[3]=-1.31E-3	d[3]=-2.68E-8	d[3]=-2.29E-8	d[3]=-5.1E-12
d[4]=4.68E-1	d[4]=-1.22E-1	d[4]=-3.45E-5	d[4]=-1.54E-8	d[4]=-4.83E-9
d[5]=-5.41E-1	d[5]=-3.44E-2	d[5]=-9.45E-6	d[5]=-2.51E-8	d[5]=-1.32E-9

6th iteration	7th iteration	8th iteration	9th iteration
d[1]=1.63E-12	d[1]=1.33E-13	d[1]=3.31E-16	d[1]=3.06E-17
d[2]=2.98E-13	d[2]=1.99E-13	d[2]=7.03E-17	d[2]=4.53E-17
d[3]=-4.31E-12	d[3]=-1.05E-15	d[3]=-8.52E-16	d[3]=-1.31E-18
d[4]=-2.53E-12	d[4]=-7.31E-13	d[4]=-5.42E-16	d[4]=-1.66E-16
d[5]=-4.61E-12	d[5]=-4.61E-12	d[5]=-2.00E-13	d[5]=-9.22E-16

Table 2

Example 2. Consider the feasible region defined by

$$\begin{cases} -x_1 + x_2 \le 2 \\ x_1 + 3x_2 \le 14 \\ 3x_1 - x_2 \le 12 \\ x_1 \ge 0,\ x_2 \ge 0, \end{cases}$$

and the piecewise linear objective function given by: $f(x) = f_1(x_1) + f_2(x_2)$ with

$$f_1(x_1) = \begin{cases} -3x_1 + 9 & 0 \le x_1 \le 1 \\ -2x_1 + 8 & 1 \le x_1 \le 2 \\ -0.5x_1 + 5 & 2 \le x_1 \le 3 \\ x_1 + 0.5 & 3 \le x_1 \le 4 \end{cases} \qquad f_2(x_2) = \begin{cases} -3x_2 + 3 & 0 \le x_2 \le 1 \\ -2x_2 + 2 & 1 \le x_2 \le 2 \\ x_2 - 4 & 2 \le x_2 \le 3 \\ 2x_2 - 7 & 3 \le x_2 \le 4 \end{cases}$$

Figure 6 exhibits the trajectories for the optimal solution when the single optimal solution is interior to the feasibility region, departing from six randomly generated solutions.

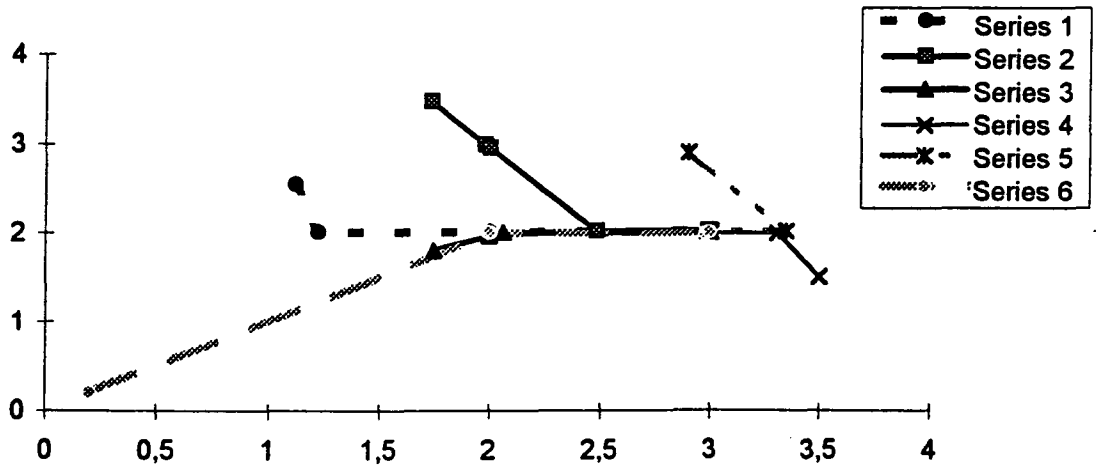


Figure 6

Example 3. Now the restrictions of Example 1 and the objective function of Example 1 produce a problem with multiple optimal solutions. These solutions are such that $2 \le x_1 \le 3$ and $x_2 = 2$. Figure 7 illustrates the behavior of the algorithm when started from different solutions, stopping at different optimal solutions.

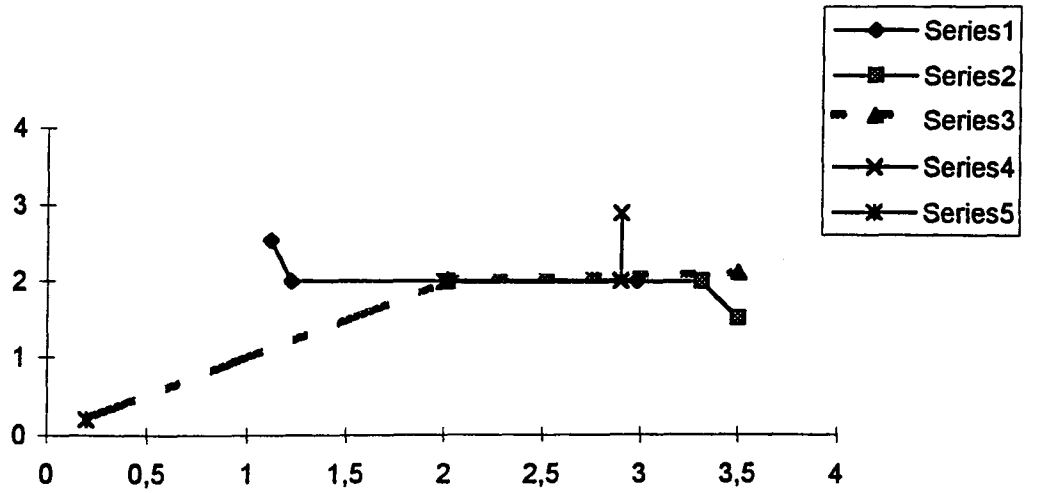


Figure 7

Example 4. (Golstein and Youdine, p. 518)

$$\min F(x) = \sum_{i=1}^{8} f_j(x_j)$$
$$s.t.\ \ Ax \le b$$
$$x \in \Omega,$$

where

$$f_1(x_1) = \begin{cases} -8x_1 & 0 \le x_1 \le 1, \\ -5x_1 - 3 & 1 \le x_1 \le 2, \\ -4x_1 - 5 & 2 \le x_1 \le 3, \\ -2x_1 - 11 & 3 \le x_1 \le 4. \end{cases} \qquad f_2(x_2) = \begin{cases} -9x_2 & 0 \le x_2 \le 1, \\ -7x_2 - 2 & 1 \le x_2 \le 2, \\ -6x_2 - 4 & 2 \le x_2 \le 3, \\ -5x_2 - 7 & 3 \le x_2 \le 4, \\ -3x_2 - 15 & 4 \le x_2 \le 5. \end{cases}$$

$$f_3(x_3) = \begin{cases} -8x_3 & 0 \le x_3 \le 2, \\ -4x_3 - 8 & 2 \le x_3 \le 5, \\ -x_3 - 28 & 5 \le x_3 \le 6. \end{cases} \qquad f_4(x_4) = \{-6x_4 \qquad 0 \le x_4 \le 8.$$

$$f_5(x_5) = \begin{cases} -8x_5 & 0 \le x_5 \le 1, \\ -5x_5 - 3 & 1 \le x_5 \le 2. \end{cases} \qquad f_6(x_6) = \begin{cases} -9x_6 & 0 \le x_6 \le 2, \\ -8x_6 - 2 & 2 \le x_6 \le 20. \end{cases}$$

$$f_7(x_7)=\begin{cases}-8x_7 & 0\le x_7\le 2,\\ -7x_7-2 & 2\le x_7\le 4,\\ -5x_7-10 & 4\le x_7\le 7.\end{cases}\qquad f_8(x_8)=\begin{cases}-11x_8 & 0\le x_8\le 1,\\ -10x_8-1 & 1\le x_8\le 2,\\ -8x_8-5 & 2\le x_8\le 3,\\ -5x_8-14 & 3\le x_8\le 4,\\ -4x_8-18 & 4\le x_8\le 5.\end{cases}$$

subject to:

$$\begin{aligned} x_1-2x_2+3x_3-x_4+5x_5-3x_6+x_7-x_8 &\le 10\\ 3x_1+x_2-x_3+2x_4+2x_5-2x_6-3x_7+4x_8 &\le 8\\ -2x_1+x_2+x_3+x_5+x_6-2x_8 &\le 7\\ 5x_1-4x_2+3x_4-2x_5-x_7 &\le 5\end{aligned}$$

and Ω given by:

$$0\le x_1\le 4,\ 0\le x_2\le 5,\ 0\le x_3\le 6,\ 0\le x_4\le 8,\ 0\le x_2\le 2,\ 0\le x_6\le 20,$$
$$0\le x_7\le 7,\ 0\le x_8\le 5.$$

If the transformation technique were applied to obtain a linear problem (see Murty [16]), instead of 8 variables we would have 25, one for each subinterval.

Below, some results are presented when Example 4 is solved. Table 3 shows 5 randomly generated initial solutions and the corresponding sequences of objective function values, which are also illustrated in Figure 8. Note the rapid decrease at the initial iterations.

initial solutions					
x[1]=	6.880E-01	6.360E-01	1.304E+00	3.480E-01	6.840E-01
x[2]=	1.150E+00	2.040E+00	4.800E+00	6.100E-01	1.050E-01
x[3]=	5.556E+00	9.420E-01	2.640E+00	3.516E+00	4.290E+00
x[4]=	6.160E+00	4.800E-02	6.328E+00	5.920E+00	7.776E+00
x[5]=	1.610E+00	1.942E+00	1.078E+00	1.528E+00	3.500E-01
x[6]=	5.760E+00	1.304E+01	1.444E+01	1.856E+01	1.120E+00
x[7]=	4.193E+00	5.775E+00	2.982E+00	8.400E-02	3.024E+00
x[8]=	3.450E+00	1.810E+00	3.550E-01	1.130E+00	2.750E+00
x[9]=	9.591E+00	3.611E+01	5.201E+01	5.533E+01	5.768E+00
x[10]=	5.101E+00	3.718E+01	2.352E+01	2.782E+01	-5.807E+00
x[11]=	1.200E+00	-6.072E+00	-1.264E+01	-1.426E+01	8.003E+00
x[12]=	-4.907E+00	1.949E+01	3.834E+00	-8.921E+00	-1.760E+01

Table 3

iterations	objective function values				
	initial sol.1	initial sol. 2	inital sol.3	initial sol.4	initial sol.5
1	-1.280E+02	3.179E+02	8.780E+02	-1.511E+02	9.893E+02
2	-1.950E+02	-2.179E+01	4.820E+02	-1.777E+02	-8.538E+01
3	-2.520E+02	-2.216E+02	-1.480E+02	-2.386E+02	-1.477E+02
4	-2.630E+02	-2.793E+02	-1.980E+02	-2.607E+02	-1.591E+02
5	-2.750E+02	-2.873E+02	-2.060E+02	-2.684E+02	-2.081E+02
6	-2.840E+02	-2.927E+02	-2.170E+02	-2.735E+02	-2.432E+02
7	-2.840E+02	-3.154E+02	-2.470E+02	-3.084E+02	-2.852E+02
8	-2.900E+02	-3.203E+02	-2.480E+02	-3.088E+02	-2.867E+02
9	-2.970E+02	-3.229E+02	-2.490E+02	-3.090E+02	-2.976E+02
10	-3.010E+02	-3.230E+02	-2.490E+02	-3.090E+02	-3.015E+02
11	-3.010E+02	-3.230E+02	-2.500E+02	-3.092E+02	-3.020E+02
12	-3.020E+02	-3.230E+02	-2.960E+02	-3.190E+02	-3.020E+02
13	-3.030E+02	-3.230E+02	-3.040E+02	-3.230E+02	-3.022E+02
14	-3.080E+02	-3.230E+02	-3.040E+02	-3.230E+02	-3.044E+02
15	-3.150E+02	-3.230E+02	-3.040E+02	-3.230E+02	-3.079E+02
16	-3.150E+02	-3.230E+02	-3.040E+02	-3.230E+02	-3.080E+02
17	-3.150E+02	-3.230E+02	-3.040E+02	-3.230E+02	-3.080E+02
18	-3.170E+02	-3.230E+02	-3.070E+02	-3.230E+02	-3.080E+02
19	-3.190E+02	-3.230E+02	-3.090E+02	-3.230E+02	-3.080E+02
20	-3.190E+02	-3.230E+02	-3.090E+02	-3.230E+02	-3.080E+02
21	-3.220E+02		-3.090E+02	-3.230E+02	-3.080E+02
22	-3.220E+02		-3.090E+02	-3.230E+02	-3.190E+02
23	-3.220E+02		-3.230E+02		-3.195E+02
24	-3.220E+02		-3.230E+02		-3.220E+02
25	-3.220E+02		-3.230E+02		-3.220E+02
26	-3.220E+02		-3.230E+02		-3.220E+02
27	-3.230E+02		-3.230E+02		-3.220E+02
28	-3.230E+02		-3.230E+02		-3.220E+02
29	-3.230E+02		-3.230E+02		-3.220E+02
30	-3.230E+02		-3.230E+02		-3.220E+02
31	-3.230E+02		-3.230E+02		-3.220E+02
32	-3.230E+02				-3.220E+02
33	-3.230E+02				-3.230E+02

34					-3.230E+02
35					-3.230E+02
36					-3.230E+02
37					-3.230E+02

Table 4

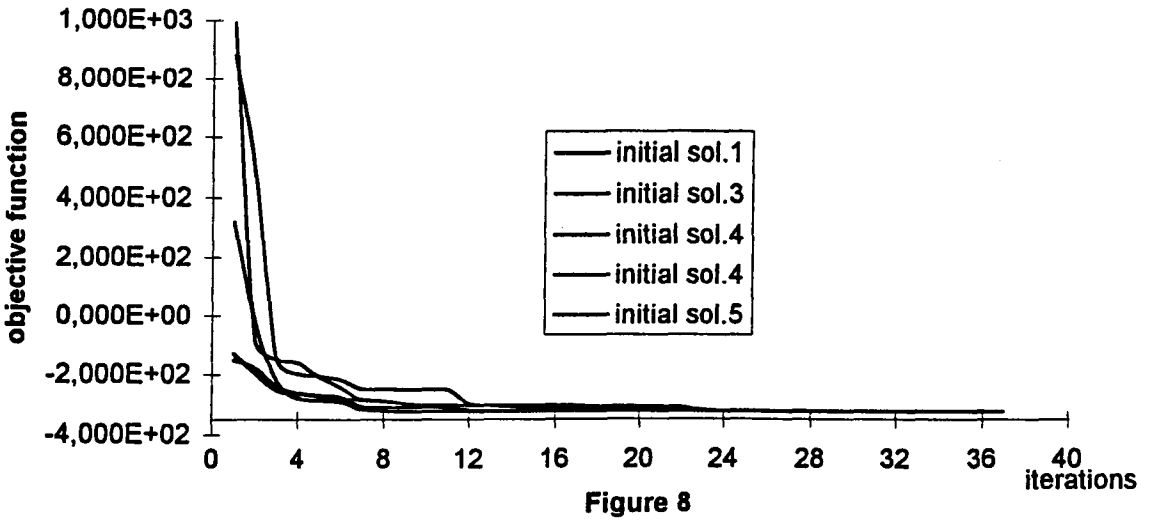


Figure 8

Example 5. Consider the following convex and separable non-linear program:

$$Min\ g(x_1, x_2) = 4x_1^2 + 2x_2^2 - 6x_1 - 3x_2$$

$$s.t.\ \begin{cases} -x_1 + x_2 \le 2 \\ 2x_1 + x_2 \le 8 \\ 2x_1 - x_2 \le 4 \end{cases}$$

$$0 \le x_1 \le 3,\ 0 \le x_2 \le 4,$$

whose feasible region is that presented in Figure 4.

The optimal solution is: $(x_1,x_2) = (0.75,\ 0.75)$, in the interior of the feasibility region. Note that the objective function is separable,and $g(x_1, x_2) = g_1(x_1) + g_2(x_2)$, where $g_1(x_1) = 4x_1^2 - 6x_1$ and $g_2(x_2) = 2x_2^2 - 3x_2$.

The non-linear program above is then approximated by a piecewise linear program:

$$Min\ f(x_1,x_2) = f_1(x_1)+f_2(x_2)$$

$$s.t.\ \begin{cases} -x_1 + x_2 \le 2 \\ 2x_1 + x_2 \le 8 \\ 2x_1 - x_2 \le 4 \end{cases}$$

$$0 \le x_1 \le 3,\ 0 \le x_2 \le 4,$$

where f_1 and f_2 are piecewise linear functions that interpolate g_1 and g_2, respectively, at the break-points: $\beta_j^{i+1} = \beta_j^i + h_j$, with $h_j = \frac{u_j}{K}$ and $\beta_j^0 = 0$, $j=1, 2$, $i=0,\ldots,K-1$, where K is the number of subintervals.

In order to compare the effects of directly solving the piecewise linear program with the equivalent linear program the interior point algorithm in Section 4.4 was used. For each *K=4, 8, 16, 32, 64, 128*, four sequences were generated starting from four different random initial solutions, and Fig. 9 depicts the average of the iteration numbers.

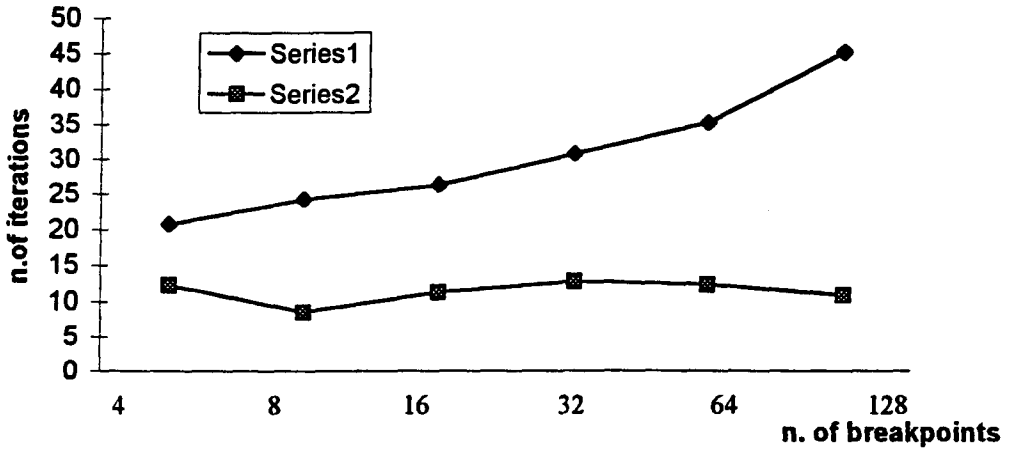


Figure 9: Evolution of iteration numbers by using interior point algorithm to solve the piecewise linear (Series 2) and equivalent linear (Series 1) programs

Figures 10 to 13 show the behavior of the objective function values.

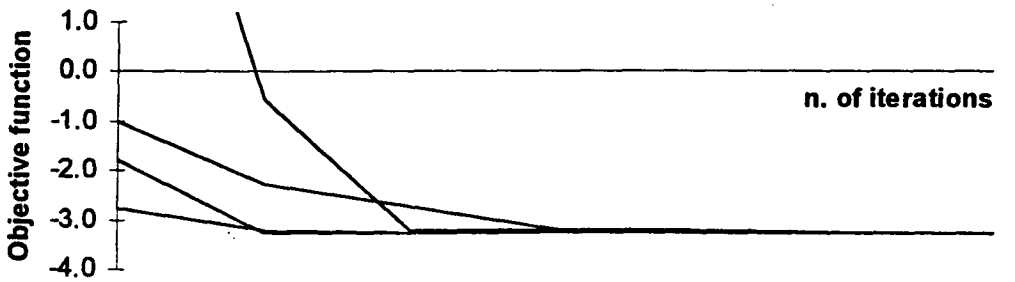


Figure 10: Behavior of obj. function solving the piecewise linear program, with k=128

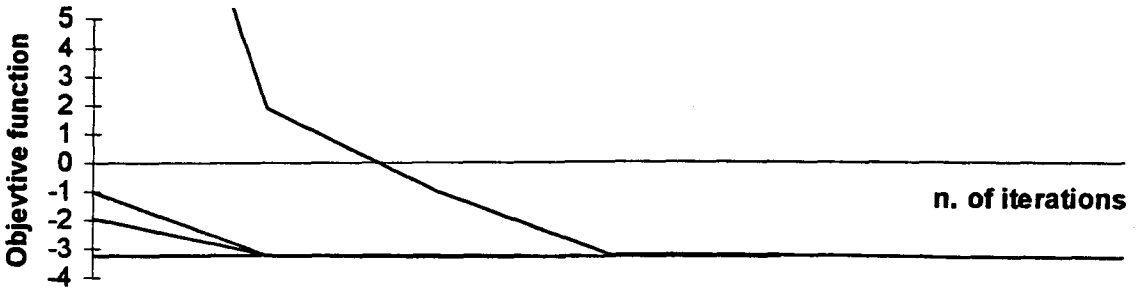


Figure 11:Behavior of obj. function solving the piecewise linear program, with k=64

Figure 11

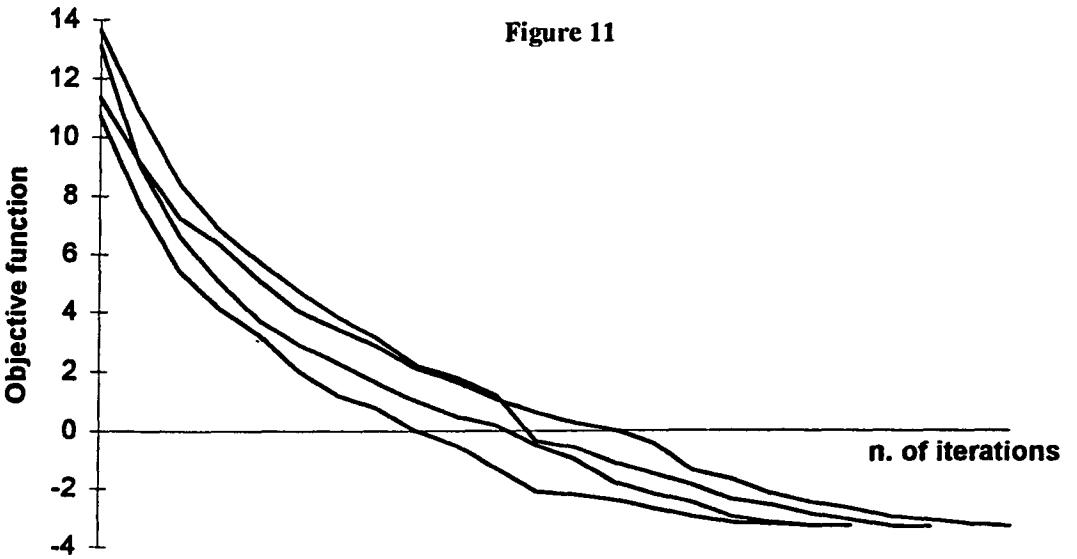


Figure 12: Behavior of obj. function solving the equivalent linear program, with k=128

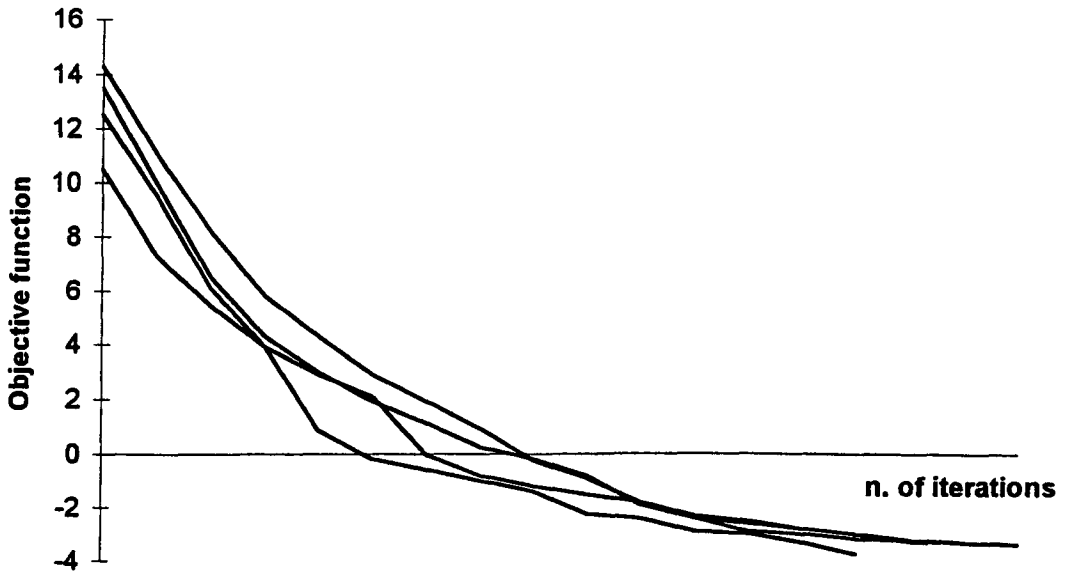


Figure 13: Behavior of obj. function solving the equivalentlinear program, with k=64

7. Conclusions and Perspectives

In this work we studied the problem of Piecewise Linear Programming, whose objective function is separable, convex and restricted to linear constraints. Methods of interior points were used showing, how methods originally developed for linear programming can be extended to piecewise linear programs. In order to do this, we defined the linearity regions of the piecewise linear function (by a virtual consideration of some boundaries) on which the moving directions were determined, following a piecewise linear line search. In this way, the piecewise linear structure was explored. The method of Sherali *et al.*, for instance, was used in order to obtain the moving direction, and some computational experiments were presented, suggesting a fertile field of research.

It was also shown that the technique proposed in this work for extending methods of interior points, cannot be obtained by the simple application of the method to the equivalent linear program and exploring its particularities. This technique can be used with simplex type methods, without losing generality. The employment of interior point methods to the equivalent linear program does not allow the components of the original variables to reach their bounds,

although only one component for each variable can be out of its bound. However, the technique proposed here has incorporated this property and the sequence obtained can be seen as a hybrid in the equivalent linear program, in which exactly **n** components (one component for each original variable) remain in the interior of their intervals and the others at one of their bounds.

The strategy used in this work for determining the search direction can be further extended to solve other piecewise linear programs, such as the well-known *minimax* problem:

$$min\ max\{f_1(x), f_2(x), \ldots, f_m(x)\}, \qquad (15)$$

where $f_i(x) = a_i^t x - b_i$, $\mathbf{x} \in \Re^n$, as illustrated in Figure 14, and the objective function given by $f(x) = max\{f_1(x), f_2(x), \ldots, f_m(x)\}$ is the bold line.

A common approach in the treatment of the *minimax* problem is its transformation into a linear program, and then the specialization of the methods of linear programming (a practice quite usual in the treatment of most piecewise linear problems, as previously observed).

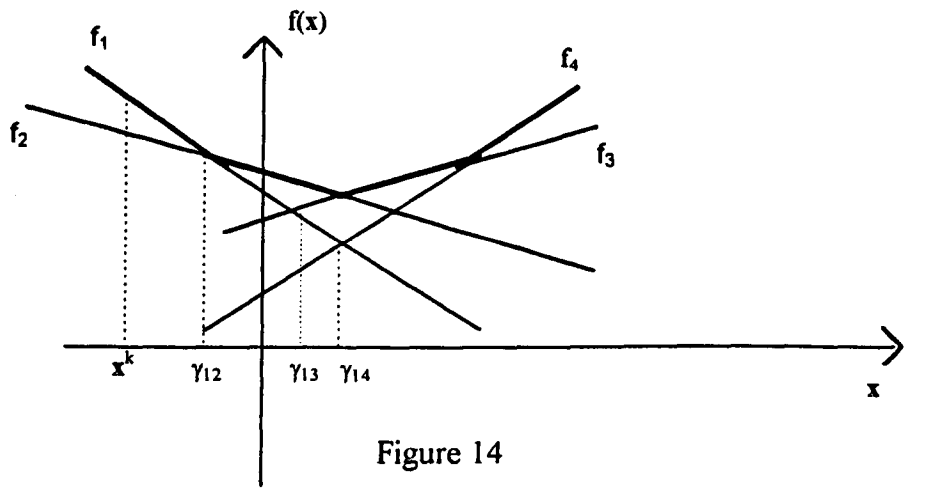


Figure 14

Observe that now the break-points are not available *a priori* as in the problem addressed in this paper. Despite this, we can extend interior point methods by using the general principles in section 4.1, that is, by defining the linear region of f(**x**) in order to choose the descent direction, and then performing piecewise linear line search. Here, as before, an interior point is such that it does not coincide with a break point of f(**x**).

Let $\mathbf{x}^k$ be an interior point. Then there is only an index r such that:

$$f_r(x^k) = max\{f_1(x^k), f_2(x^k),..., f_m(x^k)\}$$

and for $\mathbf{x}$ sufficiently near $\mathbf{x}^k$, we have $f(x) = f_r(x) = a_r^t x - b_r$.

The linearity region of f(x) around $\mathbf{x}^k$ is given by:

$$\Omega_L = \{x \text{ such as } a_i^t x - b_i \le a_r^t x - b_r,\ i = 1,\cdots,m,\ i \ne r\}.$$

To obtain the moving direction we consider the following problem:

$$\begin{aligned} &min\, f(x) = a_r^t x - b_r \\ &s.t.\ (a_i - a_r)^t x \le b_i - b_r,\ \ i=1,2,...,n,\ \ i \ne r. \end{aligned} \tag{16}$$

Note that $\mathbf{x}^k$ is an interior point to Problem (16) for which any interior point method can be used to provide a moving direction $\mathbf{d}^k$, such as, the method of Adler *et al.* [1] which is devised for inequality constraints. After determining a moving direction, we performed a line search. It is worth observing that every time a break-point is crossed, Problem (16) changes, but the direction $\mathbf{d}^k$ is maintained while it is a descent direction. In order to perform a line search, first we identify candidates for break points, denoted by γ_{ir}, which are such that (see Figure 14):

$$f_r(x^k + \gamma_{ir} d^k) = f_i(x^k + \gamma_{ir} d^k),$$

that implies

$$\gamma_{ir} = \frac{f_i(x^k) - f_r(x^k)}{a_r^t d^k - a_i^t d^k} > 0.$$

The first break point is given by: $\gamma_{k_1 r} = \min_i \{\gamma_{ir}\}$ and now we have

$$f_r(x^k + \gamma_{k_1 r} d^k) = f_{k_1}(x^k + \gamma_{k_1 r} d^k).$$

If $a_{k_1}^t d^k < 0$, then d^k is still a descent direction at $y^1 = y^0 + \gamma_{k_1 r} d^k$, with $y^0 = x^k$. The next candidates for break points are:

$$\gamma_{ik_1} = \frac{f_i(y^1) - f_{k_1}(y^1)}{a_{k_1}^t d^k - a_i^t d^k} > 0, \qquad i \ne k_1.$$

The second break point is given by: $\gamma_{k_2k_1} = \min_i \{\gamma_{ik_1}\}$. Again, if $a_{k_2}^t d^k < 0$, then d^k is still a descent direction at $y^2 = y^1 + \gamma_{k_2k_1} d^k$. The calculations above are repeated until $a_{k_j}^t d^k \geq 0$, at which point the line search ends, and

$$x^{k+1} = y^{j-1} + \gamma d^k, \ \gamma = \alpha\gamma_{k_jk_{j-1}}, \text{ with } 0 < \alpha < 1.$$

A more detailed study of the minimax problem is a theme for future research.

Acknowledgents

This research was partially supported by FAPESP and CNPq.

8 References

[1]-Adler, I., Karmarkar, N., Resende M.G.C. and Veiga, G., An implementation of Karmarkar's algorithm for linear programming, Mathematical Programming 44(1989)297-335.

[2]-Armstrong, R.D., Frome, E.L. and Kung, D.S., A revised simplex algorithm for the absolute deviation curve fitting problem , Commun. Stat., B.8(1979)175 -190.

[3]-Barrodale, I. and Roberts, F.D.K. , An improved algorithm for discrete L_1 linear approximation , SIAM J. Numer. Anal.10(1973)839 - 848.

[4]-Barrodale, I. and Young, A., Algorithms for best L_1 and L_∞ linear approximations on a discrete set, Numer. Math.,8,1966.

[5]-Cavichia, M.C., Resolução de problemas de programação linear por partes via pontos interiores, Faculdade de Engenharia Elétrica e de Computação-Univ. Campinas, Ph.D. Thesis (1997).

[6]-Fourer, R., Notes on semi-linear programming, informal report, Department of Industrial Engineering and Management Sciences, Northwestern University(Evanston, IL,1981).

[7]-Fourer, R. A, Simplex algorithm for piecewise-linear programming I - Derivation and proof., Mathematical Programming 33(1985)204-233.

[8]-Fourer, R. A, Simplex algorithm for piecewise-linear programming II - Finiteness, feasibility and degeneracy , Mathematical Programming 41(1988)281-315.

[9]-Fourer, R. A, Simplex algorithm for piecewise-linear programming III - Computational analysis and applications , Mathematical Programming 53(1992)213-235.

[10]-França, P.M., Fernandes, J.F.R. e Tavares,H.M.F., Expansão de redes telefônicas. SBA: Controle & Automação, vol.1, nº 3, julho, 1987.

[11]-Golstein, E. and Youdine, E.T., Problèmes particuliers de la programmation linéaire, Éditions MIR, Moscou,1966.

[12]-Gonzaga, C.C., Search Directions for Interior Linear Programming Methods, Algorithmica 6(1991)153-181.

[13]-Hertog, D D. and Roos, C., A survey of search directions in interior point methods for linear programming, Mathematical Programming 52(1991)481-509.

[14]-Karmarkar, N.K., A new polynomial-time algorithm for linear programming , Combinatorica4(1984)373-395

[15]-Monteiro, R.D.C. and Adler, I., Interior path following primal-dual algorithms, part II-Convex quadratic programming - Mathematical Programming 44(1989)43-46.

[16]-Murty, K.G., Linear and combinatorial programming, John Wiley and Sons, INC, New York (1976).

[17]-Ruzinski, S.A. and Olsen, E.T., L_1 and L_∞ minimization via a Karmarkar 's algorithm, IEEE Transations on Ascoustics Speech and Signal Processing, 37(1989)245 - 253.

[18]-Shanno, D.F. and Bagchi, A., A unified view of interior point methods for linear programming, Annals of Operations Research 22(1990)55-70.

[19]-Sherali, H.D., Skarpness, B.O. and Kim, B., An assumption-free convergence analysis for a perturbation of the scaling algorithm for linear programs with application to the L_1 estimation problem ,Naval Res. Logist. ,35(1988)473 - 492.

[20]-Tapia, R.A., On the formulation and theory of the Newton interior-point method for non-linear programming, Journal of Optimization Theory and Applications 89(1996)507-541.

[21]-Zhang, Y., A primal, Dual interior point Approach for computing the L_1 and L_∞ solutions of overdetermined linear systems , Tech . Report, Dept. of Math. Stat., University of Maryland, Baltimore County Campus, Baltimore, 1990.

NOTAS DO ICMSC

SÉRIE COMPUTAÇÃO

037/97 REZENDE, S. O. ; PUGLIESI, J.B. - Aquisição de conhecimento explícito ou manual - versão 1.0

036/97 SOSSOLOTE, C.R.C; RINO, L.H.M; ZAVAGLIA, C.; NUNES, M.G.V. - As manifestações morfossintáticas da linguagem UNL no português do Brasil.

035/97 FRANCO, N.B. - A volterra integral equation arising from the propagation of non-linear waves

034/97 MARTINS, A.L.; OLIVEIRA, M.C.F.; MINGHIM, R. - Visualização Científica em mecânica dos fluidos.

033/97 NUNES, M.G.V.; HASEGAWA, R.; VIEIRA, F.M.C.; SANTOS, G.H.R.; FORTES, R.P.M. - SASHE: Sistema de autoria e suporte hipermídia para ensino [1].

032/97 DIOGO, T.; FRANCO, N..F. - High order product integration methods for Volterra integral equation with weakly singular Kernel.

031/96 ARENALES, M.N.; MORABITO, R. - A theory of the generation of cutting and packing patterns Part I: foundations.

030/96 TOMÉ, M..F.; CASTELO FILHO, A. ; MURAKAMI, J.; CUMINATO, J.A.; MINGHIM, R.; OLIVEIRA, M.C.F. - Numerical simulation of axisymmetric free surface flows.

029/96 TOMÉ, M. F.; CASTELO FILHO, A.; CUMINATO, J.A.; McKEE, S. - GENSMAC3D: Implementation of the Navier-Stokes equations and Boun dary conditions for 3D free surface flows.

028/96 MARTINS, TERESA B.F.; GHIRALDELO, CLAUDETE M.; OLIVEIRA JR., O.N. - Readability formulas applied to textbooks in brazilian portuguese.