Available online at www.sciencedirect.com
SCIENCE DIRECT®

PERGAMON

Computers & Operations Research 31 (2004) 101 – 122

computers & operations research

www.elsevier.com/locate/dsw

On finding global optima for the hinge fitting problem

M. Queiroz[a,*], C. Humes Jr.[a], J. Júdice[b]

[a] *Computer Science Department, University of São Paulo, R. do Matão 1010, 05508-090 São Paulo, SP, Brazil*
[b] *Mathematics Department, University of Coimbra, 3000 Coimbra, Portugal*

Received 1 March 2002; received in revised form 1 August 2002

Abstract

This paper considers the data fitting of n given points in $\mathbb{R}^m$ by a hinge function, as it appears in Breiman (IEEE Trans. Inform. Theory 39(3) (1993) 999) and Pucar and Sjöberg (IEEE Trans. Inform. Theory 44(3) (1998) 1310). This problem can be seen as a mathematical programming problem with a convex objective function and equilibrium constraints. For the euclidean error, an enumerative approach is proposed, which is a polynomial method in the sample size n, for a fixed dimension m. An alternative formulation for the l_1 error is also introduced, which is processed by a Sequential Linear Complementarity Problem approach. Some numerical results with both algorithms are included to highlight the efficiency of those procedures.

Scope and purpose

The problem addressed here is found in applications of neural networks, nonlinear regression and data classification. Due to its nonconvex character, the application of classical nonlinear programming tools lead only to (usually suboptimal) local minima. This work proposes a global optimization method for the Hinge Fitting problem, combining combinatorial optimization and quadratic programming techniques. By achieving global optimality in reasonable time, the method compares favorably to previously proposed approaches.

MSC: 62J02; 90C33; 90C27

Keywords: Least-squares problems; Hinge functions; MPEC problems

1. Introduction

Least-squares data fitting problems abound in the literature, and have important applications wherever statistical methods are needed, as in Dynamical Systems, Signal Processing and so on. Linear

[*] Corresponding author.
E-mail addresses: mqz@ime.usp.br (M. Queiroz), chumes@usp.br (C. Humes Jr.), joaquim.judice@co.it.pt (J. Júdice).

0305-0548/04/$ - see front matter © 2003 Elsevier Ltd. All rights reserved.
PII: S0305-0548(02)00179-X

and nonlinear function approximation problems can be solved by a variety of algorithms, with solution methods ranging from Discrete and Continuous Optimization to Neural Networks, Wavelets and Computational Geometry [1–3].

The problem of nonlinear function approximation requires the choices of an error measurement and of the basis functions. The euclidean norm of the differences between the original function and the approximating function is the most common form of error measurement; the l_1 and l_∞ norms are possible alternatives that define different approximation goals. The choice of basis functions depends on the class of functions being approximated, and may range from indicator functions, polynomials and splines to trigonometric and sigmoidal functions.

The hinging hyperplanes (HH) model has recently received attention as an alternative to the sigmoidal functions of Neural Network models for nonlinear function approximation [4,5]. The HH model uses hinge functions, i.e., functions of the form $h(a) = \max\{a^t\bar{x} + x_0, a^t\bar{y} + y_0\}$ or $h(a) = \min\{a^t\bar{x} + x_0, a^t\bar{y} + y_0\}$, as basis functions in expansions such as

$$f^k(a) = \sum_{i=1}^{k} h_i(a),$$

where $f^k(\cdot)$ is an approximation of an unknown nonlinear function $f(\cdot)$. If the function $f(\cdot)$ is sufficiently smooth, then [4] there exists an $\alpha > 0$ such that for any $k \in \mathbb{N}$ there exist k hinge functions $h_1, \ldots, h_k$ satisfying

$$\left\| f - \sum_{i=1}^{k} h_k \right\|^2 \leqslant \frac{\alpha}{k}.$$

An approximation scheme has been proposed in [4], which consists of fitting the given data for the function f to a hinge h_1, then refitting the difference $f - h_1$ to a hinge h_2, then refitting the difference $f - h_1 - h_2$ to a hinge h_3, and so on. When k hinges have been computed, one might redefine each of the h_j's in turn as the hinge fitting of $(f - \sum_{i \neq j} h_i)$, and keep refining the approximation until no significant improvement is obtained.

An iterative procedure for the Hinge Fitting problem called the Hinge Finding Algorithm (HFA) has also been proposed in [4]. It is a fixed-point approach that partitions the given data in two sets and solves independently one linear least-squares problem for each set. The two linear functions obtained are used to define a new partition of the original data, which is used as input for the next iteration. The algorithm is said to converge if the partitions in two consequent iterations are the same. When the algorithm converges, the solution found is a local minimizer of the least-squares problem. Moreover, if the given data represents exact measures of a hinge function, and the initial partition is not very far from the correct one, then the algorithm is guaranteed to converge [4]. Despite its simplicity and ease of implementation, this approach has severe drawbacks:

(1) If at some iteration a trivial partition, i.e., one with an empty set at one side, is obtained, the algorithm converges prematurely. The last solution is the same as the linear least-squares solution for the original data.
(2) The algorithm may fail to converge, in which case it cycles between a finite number of hinges, and no local optimality is guaranteed.

A modification of this method is proposed in [5], in order to guarantee global convergence to a (local) optimal solution. The HFA is seen as a Newton method, and a line search is introduced to guarantee decrease of the error function at each iteration. The proposed modification corresponds to a damping of the Newton method.

As is usual when applying local optimization methods to nonconvex problems, the quality of the local minima obtained depends heavily on the starting solution. For practical applications, one is always recommended to run these methods from multiple starting points. In [6], a technique for generating a sequence of starting points has been proposed, in the context of data fitting using piecewise linear sigmoidal functions. In this way one obtains a collection of local minima, but no guarantee of global optimality.

The large variation in the quality of local minima observed in practical experiments leads one to consider a global optimization method for the Hinge Fitting problem. This paper addresses this topic by exploring two reformulations of the problem as mathematical programs with equilibrium (complementarity) constraints (MPECs). The Hinge Fitting problem with the euclidean norm (Euclidean Hinge Fitting problem) is formulated as a quadratic MPEC and the Hinge Fitting problem with the l_1-norm (Linear Hinge Fitting problem) is formulated as a linear MPEC.

An enumerative method for the global minimum of the quadratic MPEC is proposed, which is polynomial in the sample size n, for fixed space dimension m. It is shown that the only complementary cones that are enumerated are the ones associated to separable partitions of the given data (Section 2). The size of the branching tree is shown to be a polynomial function of the sample size n, for fixed space dimension m. Some computational experiments with this enumerative algorithm for medium-sized problems have been performed. The results reported in this paper show that the quality of the fittings is highly improved over those obtained by the damped HFA algorithm.

A sequential linear complementarity problem (SLCP) algorithm for processing linear MPECs has been introduced in [7]. The performance of this algorithm for processing the MPEC associated to the l_1-norm Linear Hinge Fitting problem is also investigated. The computational experience presented in this paper shows that this method is able to process efficiently the same test problems with the l_1-norm.

The structure of the paper is as follows. The formulation and general properties related to the Euclidean Hinge Fitting problem are presented in Section 2. The HFA and the damped HFA algorithm are briefly reviewed in Section 3. The global solution algorithm is discussed in Section 4. Section 5 includes the formulation for the l_1-norm Linear Hinge Fitting problem, and describes the SLCP algorithm. Computational experience is reported in Section 6 and some conclusions are presented in Section 7.

2. The Hinge Fitting problem

Let $(a_i, b_i) \in \mathbb{R}^{m-1} \times \mathbb{R}$, $i = 1, \ldots, n$ be given measurements of a function $b = f(a)$. The goal is to approximate $f(\cdot)$ by a hinge function $h(a) = \max\{a^t\bar{x} + x_m, a^t\bar{y} + y_m\}$ minimizing the squared euclidean norm of the approximation errors. For notational convenience the given points are written as $A_i = (a_i^t, 1) \in \mathbb{R}^m$ and correspondingly $x = (\bar{x}, x_m)$, so that $A_i x = a_i^t \bar{x} + x_m$. The problem is

$$\min_{x,y} \sum_{i=1}^{n} (\max\{A_i x, A_i y\} - b_i)^2.$$

By modelling the error inside the summation above as a variable $\sigma_i = \max\{A_i x - b_i, A_i y - b_i\}$ the problem becomes

$$\begin{aligned} \min \quad & \sum_{i=1}^{n} \sigma_i^2 \\ \text{s.t.} \quad & \sigma_i \geqslant A_i x - b_i, \\ & \sigma_i \geqslant A_i y - b_i, \qquad\qquad i = 1, \ldots, n, \\ & (\sigma_i - A_i x + b_i)(\sigma_i - A_i y + b_i) = 0 \end{aligned}$$

which is an MPEC problem with a convex quadratic objective function. Note that the complementarity condition guarantees that σ_i is either equal to $A_i x - b_i$ or $A_i y - b_i$. For future reference this problem is written in matrix notation as

$$\text{(HF)} \qquad \begin{aligned} \min \quad & \sigma^t \sigma \\ \text{s.t.} \quad & r = \sigma - Ax + b, \\ & s = \sigma - Ay + b, \\ & r^t s = 0, \\ & r, s \geqslant 0 \end{aligned}$$

and is referred to as the (Euclidean) Hinge Fitting problem.

2.1. Feasible solutions and separable partitions

It is clear, by construction, that $\sigma = \max\{Ax, Ay\} - b$ holds for any feasible solution of (HF), where the max above is taken componentwise. This allows a solution to be expressed simply as (x, y), since the values of σ, r and s follow uniquely. Any solution (x, y) of the problem above defines a hinge function

$$h(a) = \max\{a^t x, a^t y\}$$

and also a partition of the original data in the index sets

$$P = \{k \mid A_k x \geqslant A_k y\},$$

$$Q = \{k \mid A_k x < A_k y\}.$$

By convention, indices k such that $A_k x = A_k y$ belong to the set P.

Fig. 1 shows the hinge function $h(a)$ defined by a solution (x, y), as well as the induced partition of the data index set $\{1, 2, \ldots, n\}$. For the sake of simplicity, such a partition of the data index set is also referred to as a partition of the data.

Definition 1. A partition (P, Q) of the data is called a *separable partition* if there is a hyperplane $\{a \mid a^t z = 0\}$ such that

$$P = \{k \mid A_k z \geqslant 0\},$$

$$Q = \{k \mid A_k z < 0\}.$$

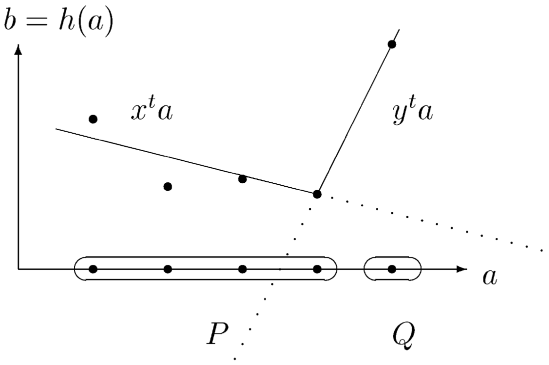

Fig. 1. Hinges and separable partitions.

In the last example, the partition (P,Q) is defined by the hyperplane $\{a \mid a^t(x-y)=0\}$. It is easy to see that separable partitions correspond to strictly separable convex sets, by considering the convex hulls $\mathrm{conv}\{A_k\}_{k\in P}$ and $\mathrm{conv}\{A_k\}_{k\in Q}$ of the partitioned data. It is easy to see, by a perturbation argument, that (P,Q) is a separable partition if and only if (Q,P) is also a separable partition. This accounts for the fact that (P,Q) and (Q,P) are considered equivalent partitions for the purpose of this paper.

The concept of separable partitions has appeared explicitly in [8,9], under the name "partitions induced by hyperplanes", and is closely related to the concept of "partitions of space induced by hyperplanes" [10]. A recent treatment of generalized separable partitions can be found in [11], for instance.

Given a partition (P,Q), the problem

$$(\mathrm{HF}_{PQ}) \qquad \begin{array}{ll} \min & \sigma^t\sigma \\ \text{s.t.} & r = \sigma - Ax + b, \\ & s = \sigma - Ay + b, \\ & r_P = 0, \\ & s_Q = 0, \\ & r, s \geqslant 0 \end{array}$$

is intimately connected to (HF). The constraints $r_P = 0$ and $s_Q = 0$, together with $r, s \geqslant 0$, define a cone in $\mathbb{R}^n_+ \times \mathbb{R}^n_+$ which is called a complementarity cone, since the complementarity condition $r^t s = 0$ automatically holds. If (P,Q) is separable, then this complementarity cone is associated to the closure of the hinge functions $h(a) = \max\{a^t x, a^t y\}$ defining the same separable partition (P,Q). This is a consequence of the fact that, under complementarity, $r_P = 0 \Leftrightarrow A_P x \geqslant A_P y$ and $s_Q = 0 \Leftrightarrow A_Q x \leqslant A_Q y$.

For every feasible point of (HF_{PQ}) the identities $\sigma = \max\{Ax, Ay\} - b$, $\sigma_P = A_P x - b_P$ and $\sigma_Q = A_Q y - b_Q$ hold. Furthermore, any feasible solution of (HF) is also a feasible solution of (HF_{PQ}) for $P = \{k \mid A_k x \geqslant A_k y\}$ and $Q = \{k \mid A_k x < A_k y\}$. This accounts for the fact that only separable partitions are worth considering from the practical viewpoint, and only these are considered by the algorithmic approaches here presented. Nevertheless, some analysis of nonseparable partitions is presented in the end of this section.

Lemma 2. *The problem* (*HF*) *admits an optimal solution. If* $\bar{b} = 1/n \sum_{i=1}^{n} b_i$, *one has*

$$0 \leqslant \textit{Optimal value}(HF) \leqslant \sum_{i=1}^{n} (b_i - \bar{b})^2.$$

Proof. The existence of an optimal solution of (HF_{PQ}) for any separable partition (P, Q) is guaranteed by the so-called Frank–Wolfe Theorem [12, Theorem 2.8.1], since each of these problems is a convex quadratic programming problem with a nonnegative objective function. Therefore the original problem (HF) admits an optimal solution, which is one of the optimal solutions of the problems (HF_{PQ}).

The optimal value of (HF) is in the range

$$0 \leqslant \text{Optimal value(HF)} \leqslant \sum_{i=1}^{n} (b_i - \bar{b})^2$$

since the objective function is bounded below by 0, and the upper bound is given by the feasible solution $r = s = 0$, $x = y = (0, \ldots, 0, \bar{b})^t$ and $\sigma = (\bar{b}, \ldots, \bar{b})^t - b$, where $\bar{b} = 1/n \sum_{i=1}^{n} b_i$. □

In fact, the solution given above is also feasible for any (HF_{PQ}), and therefore the optimal values of these problems also lie within the same range.

The above considerations motivate the following question: how many separable partitions of n points in $\mathbb{R}^m$ do exist? The general worst-case answer is 2^{n-1}, since the partitions (P, Q) and (Q, P) are equivalent. This upper bound is achieved for instance when $n = m$ and the points are chosen as $e^i = (0, \ldots, 1, 0, \ldots, 0)^t$. Nevertheless, for fixed dimension m the number of separable partitions is polynomially bounded as a function of n. This is a classical result, slightly rephrased for the present context.

Theorem 3 (Schläfli [10]). *The maximum number of separable partitions of n points in* $\mathbb{R}^m$ *is*

$$v_m(n) = \sum_{i=0}^{m} \binom{n-1}{i}.$$

Furthermore, this upper bound is achieved when the given points are in general position (*i.e.*, *no subset of* $m + 1$ *points lie in the same* $(m - 1)$-*dimensional linear manifold*).

Remark 4. The original theorem [10] considers the partition of space by n hyperplanes in general position, and proves the result with equality. This is equivalent to considering separable partitions of n points in general position [8].

The inductive proof establishes the validity of the recurrence relation

$$v_1(n) = n \quad \forall n,$$

$$v_m(n) = v_{m-1}(n) \quad \text{if } n \leqslant m,$$

$$v_m(n) = v_m(n-1) + v_{m-1}(n-1),$$

where the “general position” assumption is a technical hypothesis to ensure that the otherwise natural relationship

$$v_m(n) \leqslant v_m(n-1) + v_{m-1}(n-1)$$

holds as an equality. The solution of this recurrence relation is the expression given by the theorem above.

The present reformulation is more readily accessible in the recent works [8,9].

Note that $v_m(n) = \Theta(n^m)$, i.e., $v_m(n)$ is a polynomial in n of degree m, for fixed m. This remark entails the polynomiality of the enumerative method presented in Section 4 with respect to the number n of complementarity pairs of variables, which is the size of the given sample set.

Since the number of possible partitions of $\{1, 2, \ldots, n\}$ is 2^{n-1}, it is seen that at least

$$2^{n-1} - v_m(n) = \sum_{i=m+1}^{n-1} \binom{n-1}{i}$$

partitions are nonseparable, and this number is $\Theta(2^{n-1})$ for fixed m, i.e., it grows exponentially with n, for fixed m. The remaining part of this section analyzes nonseparable partitions and properties of the corresponding subproblems (HF$_{PQ}$).

2.2. Nonseparability and degeneracy

It is natural to consider what happens to the problem (HF$_{PQ}$) if (P, Q) is taken to be a generic (nonseparable) partition. In the sequel, the relationship between nonseparable partitions and degenerate (HF$_{PQ}$) subproblems is established.

Lemma 5. *(P, Q) is a nonseparable partition if and only if the problem (HF$_{PQ}$) is degenerate, in the sense that there is a nonempty set $\bar{Q} \subset Q$ such that $r_{\bar{Q}} = s_{\bar{Q}}$ for all feasible solutions in (HF$_{PQ}$).*

Proof. As seen before, all the problems (HF$_{PQ}$) are feasible and have optimal solutions. Moreover, for every feasible solution (x, y) of (HF$_{PQ}$), the inequalities $A_P x \geqslant A_P y$ and $A_Q x \leqslant A_Q y$ hold.

If, on the one hand, (P, Q) is a nonseparable partition of the given points, then there exists no hyperplane of the form $\{a \mid a^t x = a^t y\}$ such that $A_P x \geqslant A_P y$ and $A_Q x < A_Q y$. Together with the previous inequalities, it follows that for any feasible solution there exists at least one index $j \in Q$ such that $A_j x \geqslant A_j y$, which together with $A_Q x \leqslant A_Q y$ implies $A_j x = A_j y$.

Since the solution set of (HF$_{PQ}$) is convex, and Q is finite, it follows that the set

$$\bar{Q} = \{j \in Q \mid A_j x = A_j y \text{ holds for every feasible solution of (HF}_{PQ})\}$$

is nonempty. Suppose, for the sake of contradiction, that $\bar{Q} = \emptyset$. Then for every $j \in Q$ there would be a feasible pair (x^j, y^j) such that $A_j x^j < A_j y^j$. This implies that the convex combination $(x, y) = 1/|Q| \sum_{j \in Q} (x^j, y^j)$ would satisfy $A_P x \geqslant A_P y$ and $A_Q x < A_Q y$, contradicting the nonseparability of (P, Q).

It follows from the above remark that all feasible solutions of the problem (HF$_{PQ}$) satisfy $r_j = s_j = 0$, for all $j \in \bar{Q}$, and are thus degenerate.

Suppose, on the other hand, that there is a nonempty set $\bar{Q} \subset Q$ such that $r_{\bar{Q}} = s_{\bar{Q}} = 0$ holds for all feasible solutions in (HF_{PQ}). Then, for any feasible solution (x, y) in (HF_{PQ}), one has $A_{\bar{Q}}x = A_{\bar{Q}}y$, and therefore there exists no (x, y) feasible in (HF_{PQ}) such that $A_Q x < A_Q y$. This proves that (P, Q) is not separable. □

One possible consequence of degeneracy, in the case where $A_{\bar{Q}}$ has full rank, is the collapse of the problem (HF_{PQ}) into an ordinary linear least-squares problem, since in this case $A_{\bar{Q}}x = A_{\bar{Q}}y \Rightarrow x = y$. The nontrivial ($\mathrm{HF}_{PQ}$) subproblems, i.e., those that do not reduce to a linear least-squares problem, are crucially determined by the geometrical configuration of the points A_P and A_Q.

The degeneracy disappears as soon as one defines $P' = P \cup \bar{Q}$ and $Q' = Q \backslash \bar{Q}$. Then one obtains a separable partition (P', Q'), by the convexity argument above. The feasible region of (HF_{PQ}) is trivially contained in that of ($\mathrm{HF}_{P'Q'}$), so that only the nondegenerate problem ($\mathrm{HF}_{P'Q'}$) is worth considering.

It is important to remember that every feasible solution of (HF) is also immediately a feasible solution of at least one (HF_{PQ}) defined by a separable partition (P, Q). In the remaining of the paper, only separable partitions are considered.

3. The Hinge Finding Algorithm for (HF)

The first approach (HFA) to solve the Hinge Fitting problem has been proposed by Breiman [4], and tries to find a separable partition which is a fixed point of the function

$$\varphi(P, Q) = (P^\varphi, Q^\varphi),$$

where $P^\varphi = \{k \mid A_k x^\varphi \geqslant A_k y^\varphi\}$, $Q^\varphi = \{k \mid A_k x^\varphi < A_k y^\varphi\}$, x^φ is the solution of the (linear) least-squares problem applied to the data $\{(A_i, b_i)\}_{i \in P}$, and y^φ is the solution of the (linear) least-squares problem applied to the data $\{(A_j, b_j)\}_{j \in Q}$.

3.1. The Hinge Finding Algorithm (HFA)

```
(0) Let $(P, Q)$ be a separable partition, and let $(P^\varphi, Q^\varphi) \leftarrow \varphi(P, Q)$
(1) While $(P, Q) \neq (P^\varphi, Q^\varphi)$ set
            $(P, Q) \leftarrow (P^\varphi, Q^\varphi)$,
    and
            $(P^\varphi, Q^\varphi) \leftarrow \varphi(P, Q)$.
```

If (x^*, y^*) is a hinge corresponding to a fixed point (P^*, Q^*) of $\varphi(\cdot, \cdot)$, then (x^*, y^*) is a local minimum of the problem

$$\min_{x,y} \sum_{i=1}^{n} (\max\{A_i x, A_i y\} - b_i)^2.$$

This has been shown in [4]. A drawback of this algorithm is the presence of an awkward fixed point, namely the trivial partition $(\{1, 2, \ldots, n\}, \emptyset)$. It is not an infrequent situation to arrive at the trivial partition when approximating functions which are not hinges, or even when approximating hinges, starting with a bad initial solution.

A more serious problem is the possibility of cycling, and the difficulty of identifying this case. To overcome these problems, the use of line searches for the natural merit function $\|\max\{Ax, Ay\} - b\|$ is recommended in [5]. If the current hinge corresponding to (P, Q) is given by (x, y) and the new hinge defined by φ is given by $(x^{\varphi}, y^{\varphi})$, then the line search aims at a new solution $(x^{\lambda}, y^{\lambda}) = \lambda(x, y) + (1 - \lambda)(x^{\varphi}, y^{\varphi})$ such that

$$\|\max\{Ax^{\lambda}, Ay^{\lambda}\} - b\| < \|\max\{Ax, Ay\} - b\|.$$

The proposed strategy is to try $\lambda = 1, \frac{1}{2}, \frac{1}{4}, \ldots$ and take the first one that satisfies the above descent criterion. This is one of the many choices that guarantees global convergence, i.e., convergence regardless of the initial point, to a local optimal solution of the (HF) [5].

4. An enumerative method for (HF)

The purpose of this section is to introduce a new enumerative method for the Euclidean Hinge Fitting problem, which focuses on subproblems (HF_{PQ}) for separable partitions (P, Q) of the given points.

Definition 6. If (P, Q) is a separable partition and $I \subseteq P$ and $J \subseteq Q$, the pair (I, J) is called a *separable subpartition*. For a subpartition (I, J) the complement set $K(I, J)$ is defined as $K(I, J) = \{1, 2, \ldots, n\} \backslash (I \cup J)$. When the context is clear, the notation $K \equiv K(I, J)$ is adopted.

The definition of the problem (HF_{PQ}) is extended naturally to separable subpartitions (I, J), by considering the complementarity constraint restricted to the indices in $K(I, J)$, as follows:

$$(\mathrm{HF}_{IJ}) \qquad \begin{array}{ll} \min & \sigma^t \sigma \\ \text{s.t.} & r = \sigma - Ax + b, \\ & s = \sigma - Ay + b, \\ & r_I = 0, \\ & s_J = 0, \\ & r, s \geqslant 0, \\ & r_K^t s_K = 0 \quad \text{where } K = K(I, J). \end{array}$$

A general enumerative approach to solve (HF) is stated next for reference. The algorithm constructs an enumeration tree as follows: the interior nodes correspond to problems (HF_{IJ}) for separable subpartitions (I, J), and the leaves of the tree correspond to separable partitions. It is clear that $(\mathrm{HF}) = (\mathrm{HF}_{\emptyset\emptyset})$, so the first node to be processed is $(\emptyset, \emptyset)$. The discussion of important steps of the algorithm is left to the sequel.

4.1. Enumerative algorithm for (HF)

```
(0) Set
        $\mathscr{L} \leftarrow \{(\emptyset,\emptyset)\}$: List of nodes to investigate, and
        UB $\leftarrow \infty$: best known Upper Bound.
(1) If $\mathscr{L} = \emptyset$ stop! The optimal value is given by UB.
        Otherwise, choose $(I,J) \in \mathscr{L}$, and set
        $\mathscr{L} \leftarrow \mathscr{L} \backslash (I,J)$.
(2) (optional) Set
        $I \leftarrow I \cup \{k \in K \mid r_k \equiv 0$ in $(\mathrm{HF}_{IJ})\}$ and
        $J \leftarrow J \cup \{k \in K \mid s_k \equiv 0$ in $(\mathrm{HF}_{IJ})\}$.
(3)  Compute a lower bound LB for $(\mathrm{HF}_{IJ})$, by solving the relaxed problem

        $(\mathrm{LB}_{IJ})\quad \begin{array}{ll} \min & \sigma^t\sigma \\ \text{s.t.} & r = \sigma - Ax + b, \\ & s = \sigma - Ay + b, \\ & r_I = 0, \\ & s_J = 0, \\ & r, s \geqslant 0. \end{array}$

     If the lower bound obtained satisfies $r_K^t s_K = 0$, set
     UB $\leftarrow \min\{\mathrm{UB}, \mathrm{LB}\}$.
     If LB $\geqslant$ UB go back to step (1).
(4)  (optional) Compute new upper bounds $\mathrm{UB}_1, \ldots, \mathrm{UB}_p$ for $(\mathrm{HF}_{IJ})$, for some $p \in \mathbb{N}$. Set
        UB $\leftarrow \min\{\mathrm{UB}, \mathrm{UB}_1, \ldots, \mathrm{UB}_p\}$.
(5)  Branch: choose $k \in K$ such that $(I \cup \{k\}, J)$ and $(I, J \cup \{k\})$ are separable subpartitions, and set
        $\mathscr{L} \leftarrow \mathscr{L} \cup \{(I \cup \{k\}, J), (I, J \cup \{k\})\}$.
        Go back to step (1)
```

The next result establishes the soundness and computational complexity of the Enumerative Method.

Proposition 7. *The above-presented method is correct, in the sense that it always finds the global optimal solution of* (HF). *Besides, the total number of nodes in the enumeration tree is bounded above by* $2v_m(n) - 1 = \Theta(n^m)$, *and the method is polynomial in n for fixed m.*

Proof. The soundness is a consequence of the facts that the optimal solution corresponds to some separable partition, and that all separable partitions are implicitly enumerated by the method. The upper bound on the number of nodes is a consequence of Theorem 3 and the fact that a binary tree with $v_m(n)$ leaves has exactly $v_m(n) - 1$ internal nodes. Each node solves a fixed number of convex quadratic programming problems, each of which is polynomially solvable [13]. □

The method presented above follows a branch-and-bound strategy, so the main issues are the choice of the next node of the tree to be processed, the computation of lower and upper bounds, and the branching strategy. Another important point is the possibility of fixing additional variables in step (2), which depends on the geometry of the separable subpartitions. Each step of the algorithm is now addressed in turn.

4.2. Choice of next node

One important implementational issue is the choice of a new node of the tree to be processed after a branching step. The first-in-first-out policy treats the list of nodes as a queue, and leads to a breadth-first exploration of the tree; this is a good choice for obtaining good lower bounds quickly. The last-in-first-out policy corresponds to a stack implementation of the list, and leads to a depth-first search, which aims rapidly at upper bounds. Other mixed strategies might also be used. In the present computational experience, the breadth-first strategy led to better performance, and is used by default.

4.3. The geometry of separable subpartitions

Step (2) of the algorithm tries to enlarge the sets I and J that form a separable subpartition, by identifying indices $k \in K$ such that $r_k = 0$ or $s_k = 0$ for every feasible solution of (HF_{IJ}). Consider the sets

$$\mathcal{R}_{IJ} = \{a \mid a^t x \geqslant a^t y, \ \forall (x, y) \text{ feasible in } (\mathrm{HF}_{IJ})\},$$

$$\mathcal{S}_{IJ} = \{a \mid a^t y \geqslant a^t x, \ \forall (x, y) \text{ feasible in } (\mathrm{HF}_{IJ})\}.$$

The next result relates these sets to the indices $k \in K$ such that $r_k = 0$ or $s_k = 0$ necessarily in (HF_{IJ}).

Lemma 8. *Let (r, s, x, y, σ) be a feasible solution of (HF_{IJ}). If $A_k \in \mathcal{R}_{IJ}$, then $r_k = 0$; if $A_k \in \mathcal{S}_{IJ}$, then $s_k = 0$.*

Proof. If $A_k \in \mathcal{R}_{IJ}$, then $A_k x \geqslant A_k y$, and therefore $s_k \geqslant r_k$, which implies $r_k = 0$ due to the complementarity constraint $r_k s_k = 0$. The implication $A_k \in \mathcal{S}_{IJ} \Rightarrow s_k = 0$ follows analogously. □

A direct consequence of the previous result is the following.

Proposition 9. (HF_{IJ}) *is completely equivalent to* $(\mathrm{HF}_{\hat{I}\hat{J}})$, *where*

$$\hat{I} = \{k \mid A_k \in \mathcal{R}_{IJ}\},$$

$$\hat{J} = \{k \mid A_k \in \mathcal{S}_{IJ}\}.$$

By using the above result, step (2) of the algorithm may be reformulated as

$$(2') \quad \text{Set } I \leftarrow \hat{I} \quad \text{and} \quad J \leftarrow \hat{J}.$$

The definition of $\mathscr{R}_{IJ}$ and $\mathscr{S}_{IJ}$ leads to a natural membership test: if the problem

$$\begin{aligned} \min \quad & A_k x - A_k y \\ \text{s.t.} \quad & A_I x \geqslant A_I y, \\ & A_J y \geqslant A_J x \end{aligned}$$

has a nonnegative optimal value, then $A_k \in \mathscr{R}_{IJ}$. On the other hand, if $A_k \notin \mathscr{R}_{IJ}$, then there is some feasible (x, y) such that $A_k x - A_k y < 0$, and the above problem is unbounded. Substituting $z = x - y$ leads to the following property.

Proposition 10. *$A_k \in \mathscr{R}_{IJ}$ if and only if the problem*

$$\begin{aligned} \min \quad & A_k z \\ \text{s.t.} \quad & A_I z \geqslant 0, \\ & A_J z \leqslant 0 \end{aligned}$$

has a nonnegative optimal value. An analogous test for $\mathscr{S}_{IJ}$ is obtained by replacing min *for* max *in the linear problem above.*

Step (2) of the algorithm can then be carried as follows. First an LP is solved to test whether $A_k \in \mathscr{R}_{IJ}$, and the set I is updated if necessary. If $A_k \notin \mathscr{R}_{IJ}$, and only in this case, an LP is solved to test whether $A_k \in \mathscr{S}_{IJ}$, and the set J is updated if necessary. This prevents the same index k from being introduced in both sets I and J, in the special case where $A_k \in \mathscr{R}_{IJ} \cap \mathscr{S}_{IJ}$. Then at most two LPs are solved for each $k \in K$.

From the practical viewpoint, step (2) of the algorithm represents a large overhead when I and J are small, due to the fact that in this case most of the points A_k are neither in $\mathscr{R}_{IJ}$ nor in $\mathscr{S}_{IJ}$, and the time needed to solve the LPs above is wasted. On the other hand, if K is small compared to I and J, this test might spare some unnecessary branching in the nodes near the leaves.

It is easy to see that the dual-based representations of the sets $\mathscr{R}_{IJ}$ and $\mathscr{S}_{IJ}$ are

$$\begin{aligned} \mathscr{R}_{IJ} &= \operatorname{Conv}\{A_i\}_{i \in I} + \operatorname{Cone}\{A_i - A_j\}_{i \in I,\ j \in J}, \\ \mathscr{S}_{IJ} &= \operatorname{Conv}\{A_j\}_{j \in J} + \operatorname{Cone}\{A_j - A_i\}_{j \in J,\ i \in I}, \end{aligned}$$

where $\operatorname{Cone}\{\mathscr{X}_i\}_{i \in \mathscr{I}}$ and $\operatorname{Conv}\{\mathscr{X}_i\}_{i \in \mathscr{I}}$ denote the cone and convex hull generated by the rows of matrix $\mathscr{X}$ with indices in $\mathscr{I}$. This representation can illustrate better the result of Proposition 9, as in the next example (Fig. 2).

4.4. Computation of lower and upper bounds

A natural choice for lower bounds of the subproblems (HF$_{IJ}$) is the relaxed (HF$_{IJ}$) obtained by dropping the complementarity constraint. This is a convex quadratic programming problem that can be solved by a variety of methods.

It is immediate that the relaxed (HF), or (LB$_{\emptyset\emptyset}$) has a trivial solution with $\sigma = 0$, by setting $x = y = (0, \ldots, 0, \bar{b})^t$ and $r = s = b - (\bar{b}, \ldots, \bar{b})^t$. however, the same is not true in general about the relaxed (HF$_{IJ}$) problems. For instance, if $I = \{1, 2, \ldots, n\}$ and $J = \emptyset$, then the problem (LB$_{IJ}$)=(HF$_{IJ}$)

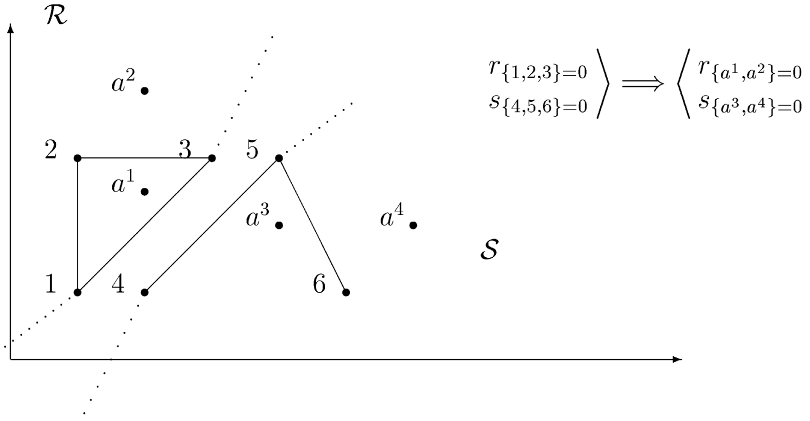

Fig. 2. The sets $\mathscr{R}_{IJ}$ and $\mathscr{S}_{IJ}$.

reduces to an ordinary linear least-squares problem which, in general, does not have a solution with the approximation error $\sigma = 0$.

The solutions of (LB$_{IJ}$) are in general infeasible for (HF$_{IJ}$); when feasible, they are also optimal for (HF$_{IJ}$), as can easily be seen. This accounts for the update of UB in step (3) of the algorithm, which leads necessarily to the verification of the condition LB $\geqslant$ UB, and to the pruning of the tree.

The condition LB $\geqslant$ UB prunes the tree, by identifying subproblems which are not worthy to solve, because the optimal solution of (HF$_{IJ}$) is in this case worse than some feasible solution already available. The efficiency of any enumerative method depends heavily on how many times this condition holds during the course of the algorithm. This number depends on finding tight lower and upper bounds.

Since the subproblems (HF$_{IJ}$) are generally as hard to solve as the original problem, finding good upper bounds is usually done by heuristic methods.

Consider an optimal solution of the problem (LB$_{IJ}$), and one of the indices $k \in K = \{1, 2, \dots, n\} \setminus (I \cup J)$. Because (LB$_{IJ}$) minimizes σ_k^2 and the only restriction applying to σ_k is $\sigma_k \geqslant \max\{A_k x, A_k y\} - b_k$, it follows that there are only two possible outcomes:

(1) $\max\{A_k x, A_k y\} - b_k < 0$; in this case $\sigma_k = 0$ and $r_k s_k > 0$.
(2) $\max\{A_k x, A_k y\} - b_k \geqslant 0$; in this case $\sigma_k = \max\{A_k x, A_k y\} - b_k$ and $r_k s_k = 0$.

Therefore, a number of the original complementarity constraints not imposed by the relaxed problem (LB$_{IJ}$) are implicitly satisfied because of the form of the objective function. This remark proves the following fact.

Lemma 11. *Let* (r, s, x, y, σ) *be an optimal solution of* (LB$_{IJ}$), *and define*

$$I' = I \cup \{k \mid r_k = 0\},$$

$$J' = J \cup \{k \mid s_k = 0\}.$$

Then (r, s, x, y, σ) *is also an optimal solution of* (LB$_{I'J'}$).

The solution of ($\mathrm{LB}_{I'J'}$) as above can easily be used to obtain an upper bound for both ($\mathrm{HF}_{I'J'}$) and (HF_{IJ}), by setting $\sigma_{K'} = \max\{A_{K'}x, A_{K'}y\} - b_{K'}$, where $K' = \{1,2,\dots,n\}\backslash(I' \cup J')$. The upper bound thus obtained is $\mathrm{LB}_{IJ} + \sigma^t_{K'}\sigma_{K'}$.

A better upper bound can be efficiently obtained by using the previous upper bound as input to the HFA algorithm of Pucar and Breiman, presented in Section 3. In fact, the initial upper bound in step (0) can be taken as the solution of the HFA with any chosen input.

Depending on the quality of these upper and lower bounds, additional upper bounds might be obtained by solving new quadratic programming problems. The assignments $I'' = I' \cup K'$ and $J'' = J' \cup K'$ leads to two possible choices for upper bounds, namely the solutions of the convex quadratic problems ($\mathrm{HF}_{I''J'}$) and ($\mathrm{HF}_{I'J''}$).

4.5. Branching

The last step of the algorithm consists in choosing some variable $k \in K$, and adding two new subproblems to the list. From the theoretical viewpoint, any choice of the branching variable is enough for the soundness of the algorithm. However, the performance of the algorithm is sensitive to this choice. A good heuristic is to choose $k = \mathrm{argmax}\{r_k s_k \mid k \in K\}$ in the optimal solution of (LB_{IJ}). This has shown to perform well in practice.

Some care must be taken to guarantee that the new nodes generated by the branching choice still correspond to separable subpartitions. This holds if $A_k \notin \mathscr{R}_{IJ} \cup \mathscr{S}_{IJ}$, which can be tested by solving one or two linear programming problems, as shown before. It is clear that usually only one linear problem is necessary, according to whether $A_k x > A_k y$ or $A_k x < A_k y$ in the solution of (LB_{IJ}); in the first case the above test reduces to $A_k \notin \mathscr{R}_{IJ}$, in the second it reduces to $A_k \notin \mathscr{S}_{IJ}$. In the unlikely event that $A_k x = A_k y$ in the (LB_{IJ}) solution, then the solution of two linear problems might be needed.

In the case $A_k \in \mathscr{R}_{IJ}$ (or $A_k \in \mathscr{S}_{IJ}$) then the corresponding sets are updated, $I \leftarrow I \cup \{k\}$ (or $J \leftarrow J \cup \{k\}$, respectively) and $K \leftarrow K\backslash\{k\}$, and the next $k = \mathrm{argmax}\{r_k s_k \mid k \in K\}$ is chosen. If K becomes empty, then (I,J) is a partition of $\{1,2,\dots,n\}$; the corresponding quadratic problem (HF_{IJ}) is solved and the node is fathomed.

If k is a valid branching choice, then two new nodes are added to the branching tree, corresponding to the problems ($\mathrm{HF}_{I'J}$) and ($\mathrm{HF}_{IJ'}$), where $I' = I \cup \{k\}$ and $J' = J \cup \{k\}$.

5. The Linear Hinge Fitting problem

In this section an alternative formulation of the Hinge Fitting problem is introduced, which is based on the l_1-norm instead of the euclidean norm. This leads to an MPEC with a linear objective function, which can be solved by a variety of methods.

The original problem with the euclidean norm replaced by the l_1-norm can be formulated as

$$\min \|\max\{Ax, Ay\} - b\|_1.$$

By modelling the difference $\max\{Ax, Ay\} - b$ as a variable σ and the $\|\cdot\|_1$ as a variable γ, the problem becomes

$$
\text{(LHF)}\qquad
\begin{aligned}
\min\quad & e^t\gamma\\
\text{s.t.}\quad & r = \sigma - Ax + b,\\
& s = \sigma - Ay + b,\\
& \gamma \geqslant \sigma,\\
& \gamma \geqslant -\sigma,\\
& r, s \geqslant 0,\\
& r^t s = 0.
\end{aligned}
$$

A possible algorithmic solution is to adapt the enumerative approach of the previous section to the Linear Hinge Fitting problem. Since all the geometric considerations in Sections 2 and 4 are related to the feasible region of the (HF), and are therefore independent of the euclidean norm, the main difference is the computation of lower and upper bounds. The computation of a lower bound for node (I, J) of the branching tree can be done by solving the following linear programming problem:

$$
\text{(LLB)}_{IJ}\qquad
\begin{aligned}
\min\quad & e^t\gamma\\
\text{s.t.}\quad & r = \sigma - Ax + b,\\
& s = \sigma - Ay + b,\\
& \gamma \geqslant \sigma,\\
& \gamma \geqslant -\sigma,\\
& r_I = 0,\\
& s_J = 0,\\
& r, s \geqslant 0.
\end{aligned}
$$

The HFA (Section 3) can also be adapted to this formulation, and can be used in the computation of upper bounds. Instead of defining x^φ and y^φ as the least-squares approximations of $\{(A_i, b_i)\}_{i\in I}$ and $\{(A_j, b_j)\}_{j\in J}$ in the main step of the HFA algorithm, x^φ and y^φ are defined, respectively, as the solutions of the following linear programming problems:

$$
\begin{aligned}
\min\quad & e^t\gamma_I\\
\text{s.t.}\quad & \gamma_I \geqslant A_I x - b_I,\\
& \gamma_I \geqslant -A_I x + b_I,
\end{aligned}
$$

$$
\begin{aligned}
\min\quad & e^t\gamma_J\\
\text{s.t.}\quad & \gamma_J \geqslant A_J x - b_J,\\
& \gamma_J \geqslant -A_J x + b_J.
\end{aligned}
$$

The heuristic upper bounds can also be adapted, either the solution obtained directly from the lower bound solution, or the solution of subproblems corresponding to separable partitions, which are also linear programs.

A drawback of this algorithmic approach is the fact that the size of the subproblems raises significantly. An efficient procedure for Linear MPECs with constraints given by a general linear complementarity problem (GLCP) has been proposed in [7]. The GLCP takes the form

$$\begin{aligned} & w = q + Mz + Ny, \\ & z, w, y \geqslant 0, \\ & z^t w = 0. \end{aligned}$$

To rewrite the (LHF) as a GLCP, the unrestricted variables x, y and σ are replaced by nonnegative variables u, u_{n+1}, v, v_{n+1}, α and β, by using the following substitutions:

$$\begin{aligned} & x = u - u_{n+1} e, \\ & y = v - v_{n+1} e, \\ & \sigma = \alpha - \beta, \end{aligned}$$

where $e = (1, \dots, 1)^t$. Furthermore, by imposing the complementarity condition $\alpha^t \beta = 0$ it is possible to express $|\sigma_i|$ as $\alpha_i + \beta_i$. Then, after rearranging terms the following equivalent problem is obtained:

$$\begin{aligned} \min \quad & e^t(\alpha + \beta) \\ \text{s.t.} \quad & \beta = b + \alpha - r - Au + (Ae)u_{n+1}, \\ & \beta + s = b + \alpha - Av + (Ae)v_{n+1}, \\ & r, s, \alpha, \beta, u, u_{n+1}, v, v_{n+1} \geqslant 0, \\ & r^t s = \alpha^t \beta = 0. \end{aligned}$$

By substituting $\beta = b + \alpha - r - Au + (Ae)u_{n+1}$ in the second equation and in the objective function, and grouping the variables as

$$w = \begin{bmatrix} \beta \\ s \end{bmatrix}, \quad z = \begin{bmatrix} \alpha \\ r \end{bmatrix} \text{ and } y = \begin{bmatrix} u \\ u_{n+1} \\ v \\ v_{n+1} \end{bmatrix},$$

the problem becomes

$$(\text{LHF}) \quad \begin{aligned} \min \quad & e^t b + c^t z + d^t y \\ \text{s.t.} \quad & w = q + Mz + Ny, \\ & z, w, y \geqslant 0, \\ & z^t w = 0, \end{aligned}$$

where

$$c = \begin{bmatrix} 2e \\ -e \end{bmatrix}, \; d = \begin{bmatrix} -A^t e \\ c^t e \\ 0 \\ 0 \end{bmatrix}, \; q = \begin{bmatrix} b \\ 0 \end{bmatrix}, \; M = \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix}$$

and

$$N = \begin{bmatrix} -A & Ae & 0 & 0 \\ A & -Ae & -A & Ae \end{bmatrix}.$$

The SLCP algorithm aims to find a global minimum of the LHF and is outlined in the sequel. The details of the algorithm can be found in [7].

The SLCP algorithm solves a sequence of GLCPs obtained from the linear MPEC in order to guarantee a monotonic decrease in the value of the objective function. The subproblem solved in each step is parameterized by λ, as below

$$(\mathrm{LHF}_\lambda) \quad \begin{aligned} & v_0 = \lambda - e^t b - c^t z - d^t y, \\ & w = q + Mz + Ny, \\ & z, w, y, v_0 \geqslant 0, \\ & z^t w = 0. \end{aligned}$$

The algorithm is stated next. The relative accuracy for the solution depends on a given parameter $\gamma > 0$.

5.1. SLCP algorithm for the Linear Hinge Fitting Problem

(0) Let γ be a small positive number. Solve the GLCP corresponding to the feasible set of the LHF, and let (z^*, w^*, y^*) be the solution. Set
$\mathrm{UB} \leftarrow e^t b + c^t z^* + d^t y^*$.

(1) Let $\lambda = \mathrm{UB} - \gamma|\mathrm{UB}|$, and solve the (LHF_λ). If this problem has no solution, stop! Otherwise, let (z^*, w^*, y^*) be the solution obtained. Set
$\mathrm{UB} \leftarrow e^t b + c^t z^* + d^t y^*$
and go back to step (1).

The last step of the algorithm consists of showing that the subproblem (LHF_λ) is infeasible, with λ slightly below the optimal value of the original MPEC. This is a very difficult problem, as is discussed in [7].

The termination criterion corresponding to the infeasibility of the subproblem (LHF_λ) leads immediately to the fact that the last upper bound found by the algorithm is an ε-optimal value of the LHF problem, where ε is given by

$$\varepsilon = \gamma|e^t b + c^t z^* + d^t y^*|.$$

Furthermore the corresponding solution (z^*, w^*, y^*) is an ε-global minimum for the Linear Hinge Fitting problem.

It is also interesting to note that the matrix M in the subproblem (LHF$_\lambda$) is positive definite, since for any x and y,

$$\begin{bmatrix} x \\ y \end{bmatrix}^t \begin{bmatrix} I & -I \\ 0 & I \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{2}[x^t x + y^t y + (x-y)^t(x-y)],$$

which is always nonnegative, and equals to zero if and only if $x = y = 0$.

The efficiency of the method is directly related to the efficiency of the procedure used to solve the (LHF$_\lambda$) subproblems. Efficient methods available to solve LCP problems with positive definite matrices [12] cannot be applied to this GLCP. These subproblems are therefore solved by a hybrid enumerative method [7], which has no restrictions on the classes of matrices involved in the GLCP.

6. Computational experience

The damped HFA and the enumerative algorithms described before have been implemented in C, and have been applied to 10 randomly generated medium-sized test problems, in a Pentium III 1.2 GHz running Linux 2.2.19.

For the sake of comparison, the same data has been submitted to the damped HFA algorithm of Pucar, Sjöberg and Breiman, to the enumerative methods of Sections 4 and 5, and to the SLCP algorithm.

The damped HFA algorithm is used as initialization to the enumerative method, so these methods are not intended to be compared against each other. Rather, they are used to show typical initialization values (local minima) against optimal values of these nonconvex problems.

To initialize the damped HFA algorithm, a balanced partition is used as input, to avoid convergence to the trivial partition. The value of the squared approximation error of the HFA solution is given, as well as the number of iterations and running time.

The Linear and Quadratic Programming subproblems in the enumerative method are solved with Lemke's Complementary Pivot Algorithm [12]. The initial upper bound in step (0) is the solution of the HFA as above. The upper bound computed in step (4) is the HFA solution using as starting point the lower bound solution at each node. The process for fixing additional variables has not been used in step (2). These options empirically minimize the running time for the tests presented, and are the default settings.

The table below summarizes the computational experience for the Euclidean Hinge Fitting problem. The sizes of MPEC formulation of the (HF) problem are given; the total number of variables is therefore $3n + 2m$, the number of complementary pairs is n and the number of linear constraints is $2n$. The performance parameters for the enumerative method are the number of quadratic problems solved (which is equal to the size of the branching tree), the number of linear problems, the running time and the optimal value. The last column of this table gives the improvement in the quality of the solution of the enumerative method with respect to the damped HFA solution, computed by

$$\frac{\text{Best HFA value}}{\text{Optimal value}}.$$

Table 1
Computational tests for the Euclidean Hinge Fitting problem

Tests					Initial sol. (HFA)			Final sol. (enumerative)				
Test #	m (space dim.)	n (# points)	# Vars. (total)	# Lin. constrs.	# Iters.	Time (s)	Value	# QPs	# LPs	Time (s)	Value	Improvement
1	5	30	100	60	20	0.01	1.3028	84	45	3.13	1.2023	1.08
2	10	35	125	70	11	0.01	0.9905	354	178	27.80	0.5306	1.87
3*	15	40	150	80	19	0.02	0.6212	1000	655	167.25	0.1256	4.94
4*	20	45	175	90	30	0.06	0.5404	1000	911	296.96	0.0166	32.40
5	25	50	200	100	16	0.06	2.5227	26	27	11.61	0	∞
6	30	55	225	110	16	0.09	3.7060	56	57	53.39	0	∞
7	35	60	250	120	29	0.23	4.9130	51	52	56.16	0	∞
8	40	65	275	130	59	0.62	0.2089	19	20	25.73	0	∞
9	45	70	300	140	41	0.58	0.3080	106	107	231.01	0	∞
10	50	75	325	150	73	1.36	0.2446	105	106	284.93	0	∞

To avoid very long runs, a maximum number of nodes has been chosen for the size of the branching tree. In the following tests, enumeration was interrupted after 1000 nodes. The test numbers are shown starred whenever this ocurred. Final objective function values for the enumerative method are globally optimal in all nonstarred tests.

In Table 1, an infinite value in the “Improvement” column corresponds to a perfect fit not achieved by the HFA algorithm. It can be seen that in those cases the number of nodes enumerated is surprisingly small. This is due to the fact that 0 is a known lower bound for the original problem, so the algorithm stops once a solution with value 0 is detected. If, in addition, the input is known a priori to correspond to a hinge, then the algorithm can be accelerated by noting that every node (I,J) with a lower bound greater than 0 can be pruned.

The relative effort of the enumerative method with respect to a complete enumeration of the separable partitions is a measure of how effective the pruning process is. This relation can be computed for each test by

$$\frac{\#\ \text{QPs}}{v_m(n)},$$

where $v_m(n)$ is given by Theorem 3. For the above tests, this measure starts with the values 0.003017, 0.000005, and from test # 3 onwards it is identically 0. It might be inferred from these tests that this relation goes to 0 as $m,n \to \infty$, and this would mean that the size of the branching tree is actually $o(v_m(n))$.

The same 10 medium-sized problems were reformulated as Linear Hinge Fitting problems and solved by the adapted enumerative method, and by the SLCP algorithm applied to the linear MPEC formulation of Section 5.

The size of the branching trees for the adapted enumerative method are shown, as well as the number of LPs solved in the branching step. The measures for the SLCP algorithm are the number of iterations and the running time. Since the last GLCP solved has no solution, any enumerative

Table 2
Computational tests for the Linear Hinge Fitting problem

Tests			Enumerative				SLCP			
Test #	# Vars. (total)	# Lin. constr.	# Nodes	# LPs	Time (s)	Value	# Iters	Time Sol. (s)	Time (s)	Optimal value
1	132	60	412	328	222.79	4.3626	6	0.23	7.49	4.3626
2*	162	70	1000	507	1419.68	2.6287	10	6.62	21.47	2.6287
3*	192	80	1000	504	2776.37	1.4140	21	0.59	21.85	1.1230
4*	222	90	1000	505	5039.91	0.3051	17	16.10	31.69	0.1456
5	252	100	691	354	5333.28	0	8	0.43	0.43	0
6	282	110	58	36	638.27	0	11	0.22	0.22	0
7	312	120	51	33	753.91	0	5	2.12	2.12	0
8	342	130	56	38	1157.89	0	4	2.80	2.80	0
9	372	140	56	39	1509.92	0	1	3.76	3.76	0
10	402	150	57	43	2410.00	0	1	4.19	4.19	0

approach has to explore the entire branching tree to detect this property. To compare the relative effort of the SLCP until it finds the optimal solution, with that needed to show that the last subproblem is infeasible, the table displays both the running time until the solution is found (marked “Time Sol.” in Table 2) and the total running time.

The performance of the enumerative method for the Linear Hinge Fitting problem is impaired by two main reasons: the size of the subproblems, and the number of nodes needed to be explored. The size of the branching tree depends on the quality of the lower and upper bounds. Apparently, the lower bound obtained by relaxation and the upper bound obtained by the adapted HFA are not as tight in the linear case as in the former euclidean case. Additionally, the adapted HFA solves two LPs in each step, and is therefore computationally more expensive than the original HFA, which solves two least-squares problems, i.e., two linear systems, in each step.

On the other hand, the SLCP algorithm (implemented as in [7]) has performed very well, taking less than 10 s to find the solution in the majority of the tests. The total number of iterations, corresponding to the number of GLCPs solved, is also small.

The Euclidean Hinge Fitting and the Linear Hinge Fitting problems have the same input, and differ in the type of approximation obtained. If, in a practical application, the choice of the norm for error measurements is not pre-determined by the specific model, then the choice of the l_1-norm leads to a simpler MPEC problem, and the algorithmic solution via the SLCP is recommended.

7. Conclusion and future work

In this paper, an enumerative algorithm for the global optimization of the Hinge Fitting problem has been introduced. Its implementational issues and computational tests have been presented, showing that it is suited to approximate nonlinear functions with a hinge function, which is the fundamental subproblem of nonlinear function approximation within the HH approximation model.

A reformulation of the problem with the l_1-norm as a linear MPEC has also been discussed and has been efficiently processed by a Sequential LCP approach.

It should be noted that the work presented here can easily be transported into a more general setting where, instead of hinge functions, one considers the maximum of $k \in \mathbb{N}$ linear functions. In this case, the branch-and-bound method would explore a k-ary branching tree, where in each node a structure of k-partitions is maintained, such that each pair of subsets within the k-partition form a separable subpartition in the sense discussed here.

It is known that positive definite matrices play a fundamental role in complementarity problems. For instance, the classical LCP with a positive definite matrix is equivalent to a convex quadratic programming problem, and can be solved very efficiently. The subproblems solved by the SLCP algorithm are actually GLCPs, but the positive definiteness of the matrix M associated with the complementary part of the subproblems may be exploited to speed up their solution.

Acknowledgements

The research of the first author was supported by FAPESP Grant Nos. 97/06227-2 and 02/01351-7. The research of the third author was supported by FCT project POCTI/35059/MAT/2000.

References

[1] Bertsekas DP, Tsitsiklis JN. Neuro-dynamic programming. Belmont, MA: Athena Scientific, 1996.

[2] Cichocki A, Unbehauen R. Neural networks for optimization and signal processing. New York: Wiley, 1996.

[3] Hassoun MH. Fundamentals of artificial neural networks. Cambridge, MA: MIT Press, 1995.

[4] Breiman L. Hinging hyperplanes for regression, classification and function approximation. IEEE Transactions on Information Theory 1993;39(3):999–1013.

[5] Pucar P, Sjöberg J. On the hinge finding algorithm for hinging hyperplanes. IEEE Transactions on Information Theory 1998;44(3):1310–8.

[6] Hush DR, Horne B. Efficient algorithms for function approximation with piecewise linear sigmoidal functions. IEEE Transactions on Neural Networks 1998;9(6):1129–41.

[7] Júdice JJ, Faustino AM. A sequential LCP method for bilevel linear programming. Annals of Operations Research 1992;34:89–106.

[8] Cover TM. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. IEEE Transactions on Electronic Computers 1965;14:326–34.

[9] Harding EF. The number of partitions of a set of n points in k dimensions induced by hyperplanes. Proc Edinburgh Math Soc 1967;15(2):285–9.

[10] Schläfli L. Theorie der vielfachen Kontinuität (1852). In: Steiner–Schläfli–Komitee der Schweizerischen Naturforschenden Gesellschaft (Eds) Gesammelte Mathematische Abhandlungen I. Basel: Birkhäuser Verlag, 1950. p. 209–12.

[11] Alon N, Onn S. Separable partitions. Discrete Applied Mathematics 1999;91(1):39–51.

[12] Cottle RW, Pang JS, Stone RE. The linear complementarity problem. New York: Academic Press, 1992.

[13] Kozlov M, Tarasov S, Hacijan L. Polynomial solvability of convex quadratic programming. Soviet Math Dokl 1979;20:1108–11.

Marcelo Queiroz graduated in Computer Science at the Instituto de Matemática e Estatística da Universidade de São Paulo, where he got his MS degree and his Ph.D. Currently he is Assistant Professor at the Computer Science Department of the Instituto de Matemática e Estatística da Universidade de São Paulo. His areas of interest are Mathematical Programming and Operations Research, with emphasis to Global Optimization, Complementarity Problems and applications.

Carlos Humes Jr. graduated in Electrical Engineering at Escola Politécnica da Universidade de São Paulo where he got his Master of Engineering. After obtaining his Ph.D. at UnC-Berkeley, he worked at USP and several computing companies in Brazil. Currently he is Full Professor at the Computer Science Department of the Instituto de Matemática e Estatística da Universidade de São Paulo. His areas of interest comprise Mathematical Programming and Control, with emphasis to Scheduling in Manufacturing Systems.

Joaquim João Júdice graduated in Mathematics at the Faculdade de Ciências e Tecnologia da Universidade de Coimbra, and obtained his Ph.D. in Mathematics and Statistics at Brunel University in 1982. Currently, he is Full Professor at the Department of Mathematics of the Faculdade de Ciências e Tecnologia da Universidade de Coimbra. His main areas of interest are Numerical Linear Algebra and Mathematical Programming with special emphasis in Large-scale Problems, Complementarity, Global Optimization and applications.