Theory and Methodology

A new procedure for identifying the frame of the convex hull of a finite collection of points in multidimensional space

J.H. Dulá *, R.V. Helgason

Department of Computer Science and Engineering, Southern Methodist University, Dallas, TX 75275, USA

Received June 1993; revised December 1994

Abstract

Consider a set, $\mathcal{A}$, of $n$ points in $m$-dimensional space. The convex hull of these points is a polytope, $\mathcal{P}$, in $\mathbb{R}^m$. The frame, $\mathcal{F}$, of these points in the set of extreme points of the polytope $\mathcal{P}$ with $\mathcal{F} \subseteq \mathcal{A}$. The problem of identifying the frame plays a central role in optimization theory (redundancy in linear programming and stochastic programming), economics (data envelopment analysis), computational geometry (facial decomposition of polytopes) and statistics (Gastwirth estimators). The standard approach for finding the elements of $\mathcal{F}$ consists of solving linear programs with $m$ rows and $n - 1$ columns; one for each element of $\mathcal{A}$, differing only in the right-hand side vectors. Although enhancements to reduce the total number of linear programs which must ultimately be solved as well as to reduce the number of columns in the technology matrix are known, the utility of this approach is severely limited by its laboriousness and computational demands. We introduce a new procedure also based on solving linear programs but with an important and distinguishing difference. The linear programs begin small and grow larger, but never have more columns than the number of extreme points of $\mathcal{P}$. Experimental results indicate that the time to find the frame using the new procedure is between about one-third and two-thirds that of an enhanced implementation of the established method currently in use.

Keywords: Convex hull problem; Frame; Linear programming; Data envelopment analysis; Redundancy

0. Introduction

The problem of identifying the frame of a finite collection of points in multidimensional space can be more properly defined as follows. Given a set $\mathcal{A}$ of $n$ points in $m$-dimensional Euclidean space, identify the extreme points (or vertices) of the convex hull of these points. The convex hull of these points is the smallest convex set containing them. It is a bounded polyhedron; i.e., a polytope with dimension at most $m$. Denote this polytope by $\mathcal{P}$. The identification of the points that are extreme points of $\mathcal{P}$ provides a minimal description of the polytope. We denote this minimal set of extreme points as the 'frame', $\mathcal{F}$, of $\mathcal{A}$. Notice that the frame is composed of points that are elements of $\mathcal{A}$.

Finding the frame of a finite collection of points is a problem that appears in equivalent forms in several applications. The problem appears directly in two important areas of optimiza-
Theory and Methodology

A new procedure for identifying the frame of the convex hull of a finite collection of points in multidimensional space

J.H. Dulá *, R.V. Helgason

Department of Computer Science and Engineering, Southern Methodist University, Dallas, TX 75275, USA

Received June 1993; revised December 1994

Abstract

Consider a set, \( \mathcal{A} \), of \( n \) points in \( m \)-dimensional space. The convex hull of these points is a polytope, \( \mathcal{P} \), in \( \mathbb{R}^m \). The frame, \( \mathcal{F} \), of these points in the set of extreme points of the polytope \( \mathcal{P} \) with \( \mathcal{F} \subseteq \mathcal{A} \). The problem of identifying the frame plays a central role in optimization theory (redundancy in linear programming and stochastic programming), economics (data envelopment analysis), computational geometry (facial decomposition of polytopes) and statistics (Gastwirth estimators). The standard approach for finding the elements of \( \mathcal{F} \) consists of solving linear programs with \( m \) rows and \( n - 1 \) columns; one for each element of \( \mathcal{A} \), differing only in the right-hand side vectors. Although enhancements to reduce the total number of linear programs which must ultimately be solved as well as to reduce the number of columns in the technology matrix are known, the utility of this approach is severely limited by its laboriousness and computational demands. We introduce a new procedure also based on solving linear programs but with an important and distinguishing difference. The linear programs begin small and grow larger, but never have more columns than the number of extreme points of \( \mathcal{P} \). Experimental results indicate that the time to find the frame using the new procedure is between about one-third and two-thirds of an enhanced implementation of the established method currently in use.

Keywords: Convex hull problem; Frame; Linear programming; Data envelopment analysis; Redundancy

0. Introduction

The problem of identifying the frame of a finite collection of points in multidimensional space can be more properly defined as follows. Given a set \( \mathcal{A} \) of \( n \) points in \( m \)-dimensional Euclidean space, identify the extreme points (or vertices) of the convex hull of these points. The convex hull of these points is the smallest convex set containing them. It is a bounded polyhedron; i.e., a polytope with dimension at most \( m \). Denote this polytope by \( \mathcal{P} \). The identification of the points that are extreme points of \( \mathcal{P} \) provides a minimal description of the polytope. We denote this minimal set of extreme points as the 'frame', \( \mathcal{F} \), of \( \mathcal{A} \). Notice that the frame is composed of points that are elements of \( \mathcal{A} \).

Finding the frame of a finite collection of points is a problem that appears in equivalent forms in several applications. The problem appears directly in two important areas of optimiza-
tion: redundancy in linear programming and stochastic programming. The frame problem is also linked with the methodology for measuring the comparative efficiency among many economic firms known as ‘data envelopment analysis’ (DEA). The frame problem plays a role in one of the classical problems in computational geometry, that of finding the hyperplanes which define the facets of the convex hull of a finite set of points. Finally, the problem of identifying the frame appears in statistics in the evaluation of Gastwirth estimators. In Section 1 we present in more detail the role of the frame problem in these applications.

Until now, the standard deterministic approach for finding the frame of a collection of points has been a procedure based on the solution of linear programs. The approach relies on the principle that the solution of an $m$ by $n - 1$ linear program can conclusively determine the ‘status’ of a particular point; that is, a linear program can be solved to establish whether a specified point is an element of the frame or not. To see this, note that a given point in $\mathcal{A}$ is an element of the frame if and only if it cannot be expressed as a convex combination of the other $n - 1$ elements of $\mathcal{A}$. Since this can be done for each of the points in the set, the problem of finding the frame can be resolved conclusively by solving $n$ very similar linear programs. However, results have been available for some time that may be used to reduce the total number of linear programs required by preprocessing the data and by applying opportunistic tests to intermediary basic feasible solutions of linear programs that may reveal the status of some of the other points in the set (see Dulá, 1991, and Dulá et al., 1992). The fact remains that current techniques must contend with the possibility of having to solve as many as $n$ large, dense, and very similar $m$ by $n - 1$ linear programs.

In the section that follows we motivate this problem by discussing in detail the different applications where the problem of finding the frame of a finite collection of points plays a central role. Following this, we present the background and theory necessary to understand the new procedure for identifying the frame of the convex hull.

1. Applications of the frame problem

The version of the convex hull problem we propose to study here is central to several important problems in operations research, economics, computer science, and statistics. We now present a detailed description of the role of the convex hull problem in five areas.

Linear programming: redundancy

Redundancy is a topic of general interest in linear programming. Redundancy is usually studied in the context of a linear program where the set of feasible solutions is defined by a collection of linear inequality constraints. A constraint is redundant if it can be omitted from the model without affecting the feasible set. Redundant constraints in a linear program are a common occurrence. These constraints make problems larger than necessary; eliminating them reduces the model size which in turn reduces the cost of solution. This is particularly important when linear programs must be solved repeatedly, e.g. when they are subproblems as in a nonlinear programming algorithm or when they are a component of a real-time control system or a decision support expert system. Also, redundant constraints may be manifestations of modeling inefficiencies and inconsistencies; their identification permits better insight into the problem. Redundancy may lead to numerical instability in the simplex method (Tomlin and Welch, 1986). Finally, redundancy and degeneracy are dualy linked.

The ‘dual’ to identifying redundancies in a system of linear inequalities corresponds to identifying extraneous variables in a linear program in standard form. Dulá (1994) established the equivalence between the problem of identifying extraneous variables in a linear program in standard form and solving a frame problem.

Stochastic optimization

In the paper by Wallace and Wets (1992) it is shown how the frame problem for cones plays a role in the standard two-stage stochastic linear program with recourse. The idea proposed in this work is to analyze and understand intrinsic problem attributes of the actual formulation in order
to extract from it essential modeling and numerical information which may permit a simplification by reformulating equivalent versions of the problem which may have a better chance of being solved. Wallace and Wets state that "there is a lot to be gained by a more efficient implementation [of an algorithm to find the frame of the convex hull, than one based on solving linear program's]."

**Economics: Data Envelopment Analysis (DEA)**

DEA is a non-parametric estimation method introduced by Charnes, Cooper and Rhodes (1978) which is used to measure the relative efficiency of a collection of firms (referred to as 'Decision Making Units' (DMUs)) in transforming a common list of inputs into outputs. The data defines a frontier that can then be used to evaluate the efficiency of each of the DMUs responsible for the observed input and output quantities. A firm is efficient if the point defined by its level of inputs and outputs lies on the efficient frontier. This frontier is on the boundary of a convex polyhedron. DMUs that do not lie on the boundary are termed inefficient and the analysis provides a measure of relative efficiency.

It is well known that the production possibility set of the, so called, 'CCR' model in DEA is a pointed polyhedral cone (Charnes et al., 1986). This implies that there exists a hyperplane that can 'cut' (intersect) the cone such that it intersects all its extreme rays. This intersection defines a bounded polyhedron on the hyperplane. The extreme points, boundary points and interior points of this polytope correspond directly to extreme rays, boundary rays and interior rays of the original cone. Therefore, all DEA data points corresponding to extreme points of the bounded polyhedron are extreme-efficient. Finding the frame of the bounded polyhedron on the hyperplane would yield this critical set of DEA efficient points.

**Computational geometry**

Preparata and Shamos (1985, Section 3.2) classify the convex hull problem into two fundamental versions. The first is the facial decomposition of the polytope described by the convex hull of a finite collection of points in multidimensional space. This problem requires the complete description of the boundary of this polyhedron. The second is the identification of the extreme points, or vertices, of this convex hull. In the first problem we say that we find the envelope of the convex hull; the second problem is the frame problem.

A by-product of finding the envelope of the convex hull of points in multidimensional space is an enumeration of the extreme points. An efficient procedure for finding the frame of the convex hull can provide a check for algorithms that are addressing the envelope problem, based on an examination of the vertex set produced. More importantly, however, is the role that an efficient procedure for finding the frame can have in recent important developments for finding the envelope as in that proposed by Seidel (1986, Algorithm 4.1). In Seidel's algorithm, it is necessary to find the optimal basic solution of a special linear program for each point in the list that is a vertex of the convex hull. The efficiency of the algorithm is adversely affected by the presence of points that are not vertices since a linear programming solution is wasted on them. Certainly, considerable effort could be eliminated if the frame problem can be efficiently solved first and no linear program solutions need be attempted for points known not to be vertices.

**Statistics**

In applied statistics, many estimators of population parameters are very sensitive to the presence of observations that lie unusually far from most other observations. Such anomalous observations are called outliers. Chatterjee and Chatterjee (1990) address the role of the frame problem in identifying outliers in multivariate statistical data. An important property of good estimators is robustness or insensitivity to deviations from an assumed distribution. An important class of robust estimators are the Gastwirth estimators (Gastwirth, 1966), which discount the effect of outliers by producing estimators after simply removing the elements of the frame. If \( P = P_0 \) is the original set of observations, let \( P_{i+1} \) be P_i
with its extreme points removed. Thus a chain of subsets,

\[ P_0 \supseteq P_1 \supseteq \cdots \supseteq P_L \neq \emptyset, \]

known as the onion of \( P \), is produced. Used in the estimate are only the first \( P_K \) subset of \( P \) having less than a \( (1 - 2\alpha) \)-fraction of the points of \( P \). This approach may be implemented recursively; each time it is required to find the frame of the convex hull of the surviving points as each successive \( P_{L+1} \) is produced. Thus, a procedure for the frame needs to be applied repeatedly.

2. Background

The frame problem has an important supporting role in each of the applications discussed in the previous section and each area contributes a body of work to solve it. In every case, though, the approaches proposed for solving the frame problem are essentially the same; namely, to solve many, slightly modified, linear programs.

There are a few works that address directly the frame problem in its general form. Perhaps the first work in this area is by Wets and Witzgall (1967) in the context of the related problem of identifying the generating elements of a convex polyhedral cone. The approach taken by Wets and Witzgall to find the ‘frame’ of the cone is essentially based on simplex method iterations. A more formal algorithm presented in Wallace and Wets (1992) is also based on the solution of linear programs.

A more recent work in this area by Rosen et al. (1992) again proposes an algorithm for identifying the extreme points of the convex hull based entirely on linear programs. Rosen et al. also report on numerical results using a parallelization scheme, apparently the first attempt at implementing in parallel. Finally, the work by Dulá, Helgason and Hickman (1992) presents the first deterministic results on identifying the extreme points of the convex hull of a finite set of points without applying linear programming; this approach is based entirely on strongly geometrical preprocessing schemes in conjunction with a specialization of the Frank–Wolfe algorithm for a quadratic program.

We now proceed to formalize the statement of the problem and to present some theoretical background. We will assume that the number of points \( n \) is greater than the dimensions \( m \) with at least one subset of \( m \) vectors being linearly independent. It is also assumed that the convex hull \( \mathcal{P} \) contains the origin in its interior (if not, the points can be translated to satisfy this condition). These assumptions are necessary to establish that the polytope \( \mathcal{P} \) has dimension \( m \). Finally, we denote by ‘con’ the convex hull of a set and by ‘int’ the interior of a set.

The following linear program can be used to determine if the element \( a^k \neq 0 \) of the set \( \mathcal{A} = \{a^1, \ldots, a^n\} \) constitutes an element of \( \mathcal{P} \):

\[
\text{LP1) }
\begin{align*}
z_1 &= \min \sum_{j=1}^{n} \lambda_j \\
& \quad \text{s.t. } \sum_{j=1}^{n} a^j \lambda_j = a^k, \\
& \quad \lambda_j \geq 0, \quad j = 1, \ldots, n
\end{align*}
\]

We have the following result regarding the solution to LP1.

**Result 1.** Assume LP1 is feasible. Then the point \( a^k \neq 0 \) is an element of the frame \( \mathcal{F} \) if and only if the optimal objective function value of LP1, \( z_1^* \), is greater than 1.

**Proof.** We will prove the contrapositive for both implications in this theorem. Suppose that LP1 is feasible and \( z_1^* \leq 1 \). We show this implies that

\[ a^k \in \text{con} \{a^j; \ j = 1, \ldots, n, \ j \neq k\}. \]

To see this consider the two cases: \( z_1^* = 1 \) and \( z_1^* < 1 \). In the first case this is clear. In the second case, let \( \lambda_1, \ldots, \lambda_m \) be the basic variables at optimality and \( a^1, \ldots, a^m \) the corresponding
optimal basic columns to LP1 when \( a_k \) is the right-hand side.

Then

\[
\begin{align*}
    a_k &= \sum_{i=1}^{m} \lambda_j \cdot a_i, \\
    \frac{1}{\sum_{i=1}^{m} \lambda_j} a_k &= \sum_{i=1}^{m} \frac{\lambda_j}{\sum_{i=1}^{m} \lambda_j} a_i, \\
    \frac{1}{z^*_i} a_k &= \sum_{i=1}^{m} \hat{\lambda}_j \cdot a_i,
\end{align*}
\]

where

\[
    \hat{\lambda}_j = \lambda_j/z^*_i, \quad i = 1, \ldots, m,
\]

so that \( \sum_{i=1}^{m} \hat{\lambda}_j = 1 \). Therefore,

\[
(1/z^*_i) a_k \in \text{con}\{a_i; \ i = 1, \ldots, m\} \subseteq \mathcal{P}.
\]

Recall that the origin is in the polytope \( \mathcal{P} \). This means that, for any point \( \bar{a} \in \mathcal{P}, \alpha \bar{a} \in \mathcal{P} \) for \( 0 \leq \alpha \leq 1 \). Since \((1/z^*_i) a_k \in \mathcal{P} \) and \( 1/z^*_i > 1 \) we have that \( a_k \) is in the interior of \( \mathcal{P} \) and therefore not an extreme point of \( \mathcal{P} \).

To prove the converse, suppose that \( a_k \neq 0 \) is not an element of \( \mathcal{F} \), so \( a_k \) is not an extreme point of \( \mathcal{P} \). By Carathéodory's Theorem, we know that \( a_k \) can be expressed as the convex combination of \( m + 1 \) extreme points of \( \mathcal{P} \). These extreme points constitute a feasible solution to LP1 such that the objective function value is 1 and \( z^* \neq 1 \). \( \Box \)

A corollary of Result 1 is that if LP1 is infeasible, we may conclude that the point \( a_k \neq 0 \) is an element of the frame \( \mathcal{F} \). This corollary follows directly from the 'only if' part of the proof.

The linear program formulation LP1 is a generic form which can be used to resolve whether or not the point \( a_k \in \mathcal{A} \) belongs to \( \mathcal{F} \). Notice how it is possible to identify conclusively the status of all the points in the set \( \mathcal{A} \) by solving this linear program \( n \) times over all right-hand side vectors \( a^1, \ldots, a^n \).

Linear programming formulations for solving the frame problem currently proposed are equivalent to LP1 and the approach of repeated solutions of linear programs such as the one here are standard. For example, in Rosen et al. (1992) the approach is to add the constraint

\[
    \sum_{j=1}^{n} \lambda_i = 1 \\
    \lambda_i \neq 1
\]

to formulation LP1, discard the objective function and then apply Phase 1 to verify if the set of \( m + 1 \) equalities has a nonnegative solution. The approach presented in Wallace and Wets (1992) is also based on verifying feasibility but since their formulation is for finding the extreme rays of the positive cone generated by the elements of \( \mathcal{A} \), the constraint \( \sum_{j=1, i \neq k}^{n} \lambda_j = 1 \) is not needed. The linear programming formulation applied in DEA introduces extra variables, one to measure 'efficiency' and the rest used as slacks.

3. Results

We will now present a collection of new results related to linear programming formulations of the LP1 type. The data consists of the \( n \) points in \( \mathbb{R}^m: a^1, \ldots, a^n \) which constitute the set \( \mathcal{A} \). We assume that the polyhedron

\[
\mathcal{P} = \text{con}\{a^1, \ldots, a^n\}
\]

contains the origin in its interior, and has full dimension. The following definitions are required.

**Definition 1.** \( \hat{0} = (0, \ldots, 0) \) denotes the zero vector with dimension defined by the context.

**Definition 2.** The set

\[
H(a, \beta) = \{ y \in \mathbb{R}^m | \langle a, y \rangle = \beta \}
\]

is the hyperplane in \( \mathbb{R}^m \) with orthogonal vector \( a \) and level value \( \beta \).

\[
H^{+}(a, \beta) = \{ y \in \mathbb{R}^m | \langle a, y \rangle > \beta \}
\]

and

\[
H^{-}(a, \beta) = \{ y \in \mathbb{R}^m | \langle a, y \rangle < \beta \}
\]

are its associated open halfspaces.

**Definition 3.** \( H(a, \beta) \) is a supporting hyperplane of \( \mathcal{P} \) if \( \langle a, x \rangle = \beta \) for some \( x \in \mathcal{P} \) and \( \langle a, z \rangle \leq \beta \) for all \( z \in \mathcal{P} \).
Definition 4. A face of \( \mathcal{P} \) is the intersection of \( \mathcal{P} \) and a supporting hyperplane.

Definition 5. A facet of \( \mathcal{P} \) is an \( m - 1 \) dimensional face of \( \mathcal{P} \).

We propose to investigate the following linear program:

\[
\text{(LP')} \\
\quad z' = \min \sum_{j=1}^{n} \lambda_j \\
\text{s.t. } \sum_{j=1}^{n} a_j^i \lambda_j = b, \\
\quad \lambda_j \geq 0, \quad j = 1, \ldots, n,
\]

where \( b \) is an arbitrary nonzero vector in \( \mathbb{R}^m \) and not necessarily one of the elements of \( \mathcal{A} \). Notice also that the index \( j \) is defined over all its possible values without excluding any as in the original expression for LP1. Finally, observe that LP' is always feasible and its solution bounded since, by assumption, \( \mathcal{A} \) has full dimension and contains the origin in its interior. Denote by \( z'^* \) the optimal objective function value of LP'.

Proposition 1. If \( z'^* \) is the optimal solution to LP' for some \( b \neq 0 \), then

(i) \( z'^* < 1 \) if and only if \( b \) is interior to \( \mathcal{P} \).

(ii) \( z'^* = 1 \) if and only if \( b \) is on the boundary of \( \mathcal{P} \).

(iii) \( z'^* > 1 \) if and only if \( b \) is exterior to \( \mathcal{P} \).

Proof. (i): Let \( z'^* < 1 \). Denote by \( \lambda_{j_1}, \ldots, \lambda_{j_m} \) the basic variables of an optimal basic solution and by \( a_{j_1}, \ldots, a_{j_m} \) the corresponding basic columns of LP' when \( b \neq 0 \) is the right-hand side. Then, as in the proof to Result 1,

\[
\frac{1}{z'^*} b = \sum_{i=1}^{m} \hat{\lambda}_i a_i,
\]

where

\[
\hat{\lambda}_i = \lambda_{j_i}/z'^*, \quad i = 1, \ldots, m.
\]

Note that

\[
\sum_{i=1}^{m} \hat{\lambda}_i = 1,
\]

and therefore, \( (1/z'^*)b \) belongs to the polytope \( \text{con}(a_i^i; i = 1, \ldots, m) \subset \mathcal{P} \). Since the origin is in the interior of the polytope, the point \( b \) which is ‘shorter’ than \( (1/z'^*)b \) is necessarily interior to the polytope.

To prove the converse of (i) assume \( b \in \text{int} \mathcal{P} \) and employ the following lemma.

Lemma. For \( b \) interior to \( \mathcal{P} \), there exist a collection of points, \( a_i^1, \ldots, a_i^L \), in \( \mathcal{A} \) and two sets of corresponding multipliers, \( \hat{\lambda}_1 > 0, \ldots, \hat{\lambda}_L > 0 \) and \( \lambda_1 > 0, \ldots, \lambda_L > 0 \), such that

\[
b \in \text{con}(a_i^1, \ldots, a_i^L) = \sum_{k=1}^{L} a_i^k \hat{\lambda}_k, \quad \sum_{k=1}^{L} \hat{\lambda}_k = 1, \quad (1)
\]

and

\[
0 \in \text{con}(a_i^1, \ldots, a_i^L) = \sum_{k=1}^{L} a_i^k \hat{\lambda}_k, \quad \sum_{k=1}^{L} \hat{\lambda}_k = 1. \quad (2)
\]

Proof of Lemma. Consider the following procedure. Generate the line segment, \( \mathcal{L}_1 \), connecting the two points \( \hat{0} \) and \( b \); with endpoints on the boundary of the polytope \( \mathcal{P} \). Name the two endpoints of \( \mathcal{L}_1 \), \( p_1^{1,1} \) and \( p_1^{1,2} \). If both \( p_1^{1,1} \) and \( p_1^{1,2} \) are extreme points of \( \mathcal{P} \), stop. Otherwise, for any nonextreme point, say \( p_1^{1,1} \), generate a line segment, \( \mathcal{L}_2 \), on the same face of \( \mathcal{P} \) containing \( p_1^{1,1} \) in its relative interior, extending from one boundary of the face to the other to generate two new points \( p_1^{1,1}_{1,2} \) and \( p_1^{1,2}_{1,2} \). Every nonextreme point generated in this fashion defines a new line segment on a face of \( \mathcal{P} \) with fewer dimensions. If we repeat this procedure for every nonextreme point at the end of any generated line segment, we will produce a sequence of segments \( \mathcal{L}_i, \ldots, \mathcal{L}_K, i = 1, 2, \) and corresponding points \( p_1^{1,1}_{i,2}, \ldots, p_1^{1,2}_{i,2}, i = 1, 2 \). Note that \( K \leq m + 1 \) because the number of superscripts is incremented whenever the dimension of the face is decremented. When this procedure ends, every endpoint of the last generated line segment will necessarily be an
If \( \rho \) is an optimal basic feasible solution to LPI and \( \pi \) is a complementary optimal dual solution, then the basis is composed of points \( a^{i_1}, \ldots, a^{i_m} \) that are elements of the same \((m-1)\)-dimensional facet of \( \mathcal{P} \).

**Proof.** Suppose that the \( m \) points \( a^{i_1}, \ldots, a^{i_m} \) constitute a complementary optimal basis to LPI but are not all on the same facet of \( \mathcal{P} \). Then, by complementarity, the system
\[
\pi^T a^{i_h} = 1, \\
\vdots \\
\pi^T a^{i_m} = 1
\]
has a solution \( \hat{\pi} \) that defines a unique hyperplane in \( \mathbb{R}^m \), \( H(\hat{\pi}, 1) \). Since \( a^{i_1}, \ldots, a^{i_m} \) are not all on the same face of \( \mathcal{P} \), \( H(\hat{\pi}, 1) \) does not support \( \mathcal{P} \). This means that
\[
H^{++}(\hat{\pi}, 1) \cap \mathcal{P} \neq \emptyset
\]
and
\[
H^{--}(\hat{\pi}, 1) \cap \mathcal{P} \neq \emptyset.
\]
This implies that there is at least one element in \( \mathcal{A} \), say \( a^{i'} \), such that \( \langle \hat{\pi}, a^{i'} \rangle > 1 \) violating the dual-feasibility of the basis. 

The condition that the optimal solution to LPI be such that it generates a complementary optimal dual solution is required since not any optimal solution to LPI satisfies this condition. A degenerate solution may be optimal but its complementary dual may not. In this case, the optimal solution to LPI is achieved via \( m \) points of \( \mathcal{A} \) that are not on the same face of \( \mathcal{P} \). Note that an optimal basic feasible solution to LPI arrived
Proposition 3. The optimal basis to LP1', if unique, is composed of points $a^1, \ldots, a^m$ which are elements of the frame $\mathcal{F}$.

Proof. A unique optimal solution to LP1' must necessarily generate a complementary optimal dual solution. Therefore, from Proposition 2, $a^1, \ldots, a^m$ are necessarily on the same facet of $\mathcal{P}$. Again by uniqueness, these $m$ points must be the $m$ points required to define the $m-1$ dimensional facet of $\mathcal{P}$. □

The results presented here can be directly applied to the formulation LP1 presented in the previous section. There are two important consequences of these results. The first is that any time the original linear program LP1 is solved and a unique optimal basis is obtained, the status of $m$ points in $\mathcal{A}$ (those which constitute the basis) is instantly revealed as elements of the frame. The second is that every time a point is discovered not to be an element of the frame it can be removed from subsequent applications of the linear program formulation. These ideas can be used to enhance the procedure for identifying the frame of $\mathcal{A}$ by reducing the total number of linear programs that have to be solved as well as by reducing their size by removing columns from the matrix of coefficients.

Using the formulation LP1 and the results accompanying it means that it is required that both the objective function value and the basic feasible solution be known to determine whether a point belongs to the frame. The fact that, eventually, an accurate optimal basic feasible solution to LP1 is required is one reason why interior point methods are not used. Another reason is that the input-output matrix in LP1 is dense with many more columns than rows. This is a particularly unattractive structure for interior point methods since these are very sensitive to the number of columns.

4. New procedure

Before we proceed we need to present more notation and preliminary discussion. We will assume that we know some of the elements of the frame. This can actually be achieved by applying simple preprocessing schemes to the set $\mathcal{A}$ (see Dulá et al., 1992). We will employ the following symbols and operations:

At any given point in the procedure, the set $\mathcal{A}$ is partitioned into three subsets: $\mathcal{A}^E, \mathcal{A}^U, \mathcal{A}^N$, where:

$\mathcal{A}^E = \text{The set of all currently known elements of } \mathcal{F}$.

$\mathcal{A}^N = \text{The set of all currently known non-extreme points of } \mathcal{P}$.

$\mathcal{A}^U = \text{The set of all other points of } \mathcal{A}, \text{whose status is yet to be assigned}$.

It is also convenient to introduce the following intermediary polytope:

$\mathcal{P}^E = \text{The convex hull of the points in } \mathcal{A}^E, \text{itself a polytope}. \text{Note: } \mathcal{P}^E \subset \mathcal{P}$.

The new procedure for finding the frame of the convex hull of a finite set of points in $\mathbb{R}^m$ is based on the following scheme.

Procedure BUILD

Step 1. Select a point from $\mathcal{A}^U$ and determine if it belongs to $\text{con } \mathcal{A}^E$ (the ‘current’ convex hull). If so, the point is termed interior otherwise it is termed exterior and identified as $a^*$. If the point is interior, remove it from from $\mathcal{A}^U$ and add it to $\mathcal{A}^N$, select another point from $\mathcal{A}^U$ and repeat Step 1; otherwise proceed to Step 2.

Step 2. Generate a direction $v \in \mathbb{R}^m$ such that it is perpendicular to a hyperplane $H(v, \beta)$ which separates $a^*$ and $\text{con } \mathcal{A}^E$ and such that

$\mathcal{A}^E \subset \{ y | v^T y \leq \beta \}$.

Proceed to Step 3.

Step 3a. Calculate the maximum of the inner product $\langle v, a^* \rangle$ and $\langle v, a^k \rangle$, $\forall a^k \in \mathcal{A}^U$. The maximum is attained at an extreme point of $\mathcal{P}$.

Step 3b. Calculate the minimum of the inner product $\langle v, a^k \rangle$, $\forall a^k \in \mathcal{A}^U \cup \mathcal{A}^E$. The minimum, if it is attained at a point in $\mathcal{A}^U$, is an extreme point of $\mathcal{P}$.

Return to Step 1.
The following result establishes the validity of this procedure:

**Proposition 4.** Every pass through procedure BUILD's Step 3a (or 3b) resulting in a unique point generates a new element of $\mathcal{P}$.

**Proof.** The result hinges on the fact that the optimal solution to the linear program
\[
\max_{x \in P \subset \mathbb{R}^m} \langle c, x \rangle,
\]
or
\[
\min_{x \in P \subset \mathbb{R}^m} \langle c, x \rangle,
\]
for some $c \neq \mathbf{0}$, an arbitrary vector in $\mathbb{R}^m$, and $P$ any nonempty polytope, occurs at extreme points of $P$. Let us limit the discussion to the case of the maximum in Step 3a; the arguments for the case of minimum (Step 3b) follow directly. When the maximum is unique, the point where this value is attained is necessarily an extreme point of the polytope $P$. If $H(v, \beta)$ is a supporting hyperplane separating the exterior point, $a^*$, from the current polytope such that the polytope belongs to $H^{-}(v, \beta)$ then, since $\langle v, a^* \rangle > \beta$, either $\langle v, a^* \rangle$ is the maximum value for the inner product or the maximum is attained at some other point in $\mathcal{A}^U$. In any case, the maximum is attained and, if unique, it is necessarily an extreme point of $\mathcal{P}$ and an element of the frame. □

This result indicates that the resolution of ties reduces to a smaller version of our original frame problem. The resolution of ties is an implementation problem. Note, though, that if just two points are involved in a tie they are both necessarily extreme points of $\mathcal{P}$.

This procedure for finding the elements of the frame works by identifying at least one new extreme point at every pass. Therefore, it is guaranteed to finish in, at most, as many passes as there are points in $\mathcal{A}^U$. The difference between the traditional approach described in Section 2 and this new procedure is that, in the first one, every iteration works with the 'whole' polytope, extracting at least one point at each iteration, and applies a check to determine if such a point is extreme or not. In the new procedure proposed here, the polyhedron steadily 'grows' one extreme point at a time until $\mathcal{P}$ is completely generated.

In the first implementation (Dulá et al., 1992) which had as a declared objective not to use linear programming, a special streamlined version of the Frank–Wolfe method was used to take care of Steps 1 and 2 simultaneously for every point with undefined status. Whether a point is interior or exterior to a convex hull and, if exterior, determining a direction $v$ defined by the point and its projection, can be formulated as a quadratic program. Applying the Frank–Wolfe method to this problem is relatively efficient since each iteration requires only finding the minimum of a list of values (there is no need to solve a linear program since the formulation has a single linear constraint). One problem with the Frank–Wolfe method in this application resides in con-
structing the quadratic objective function which requires evaluating all pairwise inner products of the points involved in the current convex hull. This difficulty increases with every iteration. Convergence is another problem associated with the Frank–Wolfe method. This is particularly serious when the point in question happens to be interior. Finally there is a problem with the numerical detection of boundary points. This problem is related to convergence since lack of precision may result in the classification of a point as 'interior' when, in fact, it is marginally outside the convex hull.

An alternative to using a Frank–Wolfe approach is suggested by the following result. Let $\hat{a}^1, \ldots, \hat{a}^\hat{n}$ be the elements of $\mathcal{E}$. Suppose that $\hat{n} \geq m + 1$; assume that the origin is in the interior of their convex hull; and assume also that the dimension of $\mathcal{E}$ is $m$ (a discussion on how these conditions are attained is presented later). Let $a^k$ be a point with undefined status. Consider the following linear program formulation:

(LP2)

$$
\begin{align*}
\hat{n} &= \min \sum_{j=1}^{\hat{n}} \lambda_j \\
\text{s.t.} & \quad \sum_{j=1}^{\hat{n}} \hat{a}^j \lambda_j = a^k, \\
& \quad \lambda_j \geq 0, \quad j = 1, \ldots, \hat{n}.
\end{align*}
$$

A solution to this linear program yields the following information.

**Proposition 5.** An optimal basis for LP2 with a complementary optimal dual solution for an exterior point $a^k$ defines a supporting hyperplane for $\mathcal{E}$ that separates it from $\hat{a}^k$. Moreover, this hyperplane is given by $H(\hat{\pi}^*, 1)$ where $\hat{\pi}^*$ is the corresponding optimal dual solution.

**Proof.** If $a^k$ is exterior to $\mathcal{E} = \text{con}(\hat{a}^1, \ldots, \hat{a}^\hat{n})$, then a basis to LP2 with a complementary optimal dual solution is composed of elements belonging to the same facet of $\mathcal{E}$ (Proposition 2). Let us call the elements of the basis $\hat{a}^{i_1}, \ldots, \hat{a}^{i_m}$.

Therefore, the unique vector $\hat{\pi} \in \mathbb{R}^m$, which solves the system of $m$ equations

$$
\hat{\pi}^T \hat{a}^{i_1} = 1,
$$

$$
\vdots
$$

$$
\hat{\pi}^T \hat{a}^{i_m} = 1,
$$

defines the hyperplane $H(\hat{\pi}, 1)$ that supports the current polytope at the facet defined by the $m$ points $\hat{a}^{i_1}, \ldots, \hat{a}^{i_m}$. To see this, notice that since the basic solution is optimal, the system of linear equations is a necessary condition for complementary slackness. The remaining complementarity conditions imply that $\hat{\pi}^T \hat{a}^i \leq 1$ for all the other points $\hat{a}^i \in \mathcal{E}$. Since at optimality $\hat{z}^2 > 1$ (since $a^k$ is exterior, see Proposition 1) it follows that $\hat{\pi}^T \hat{a}^k > 1$ (strong duality), and we may conclude that the point $a^k$ is in $H^{++}(\hat{\pi}^*, 1)$ while $\mathcal{E}$ is on the other side of the hyperplane. □

The formulation LP2 along with the result in Proposition 5 suggests the following procedure for identifying the frame. For Step 1 of Procedure BUILD use a given point from $\mathcal{E}^U$ as the right-hand side of LP2 to check if it is 'interior' or 'exterior' to the 'current' polytope. Proposition 1 guarantees that if the optimal objective function value of LP2 formulated such that its columns are the elements of $\mathcal{E}$ is less than or equal to 1, the point in question is interior and it must be transferred to the set $\mathcal{E}$, otherwise the point is exterior to the 'current polytope'. Proposition 5 provides the vector $v$ required in Step 2. This is, of course, the vector of dual multipliers of the optimal basic feasible solution. This way the new procedure is fully specified.

The only remaining issue is the initialization. The application of linear program LP2 requires that there be at least $m + 1$ affinely independent columns, which should be elements of the frame, and which contain the origin in their convex hull. Otherwise the linear program is not guaranteed to be feasible for any right-hand side and the powerful inferences possible from Proposition 1 are invalid. There are several ways to assure that these initialization conditions are attained. We propose that the procedure be initialized as follows: Find the vector in $\mathcal{F}$ with greatest norm. This point is necessarily an element of $\mathcal{E}$ (see
Result 2 in Dulá et al., 1992). Take the negative of this ‘max-norm’ vector and use it as the right-hand side element of the linear program LP1. The resultant basic feasible solution, if unique, is composed of \( m \) more elements of the frame; this from Proposition 3. (If not unique select another right-hand side which is the negative of some other element of the frame until one is found which generates a unique optimum.) These \( m \) vectors contain the right-hand side in their positive cone; therefore, by applying Farkas’ Lemma we may conclude that the \( m \) vectors in conjunction with the negative of the right-hand side vector constitute an affinely independent set of \( m + 1 \) vectors that positively span the space. Moreover, the convex hull of these vectors necessarily contain the origin (apply Stiemke’s Theorem of Alternative). Note that this initialization scheme essentially identifies \( m + 1 \) points from the frame of \( \mathcal{F} \) the convex hull of which is an \( m \)-dimensional simplex which contains the origin in its interior. Also, by selecting the ‘max-norm’ vector as the ‘seed’ for the right-hand side of LP1 we may suppose that the resultant simplex is, in some sense, large (for more on how theorems of alternatives play a role in these ideas and on how this initialization scheme generates a ‘large’ simplex (see Dulá, 1993).

This means we may proceed as follows. Formulate LP2 with the points in \( \mathcal{F}^E \) as it columns until an exterior point is found. By Proposition 5, the optimal dual vector is the vector \( v \) of Step 2 of procedure BUILD. Now we can apply Step 3 to generate a new element of the frame.

Notice that the procedure based on the linear program formulation LP2 is fundamentally different from the approach discussed in the previous section. Here we ‘build-up’ the polytope. The procedure using LP2 generates linear programs that grow by one column every time a new vertex of \( \mathcal{P} \) is identified. In the case of LP1 the size of the linear program starts at \( m \) by \( n - 1 \) and, if enhancements are implemented, the number of columns may be reduced by removing points that are discovered not to belong to the frame. Since the columns of LP2 are always elements of the frame, the size of the final linear program is determined by the total number of extreme points of \( \mathcal{P} \). On the other hand, a difference which favors the approach based on LP1 is the necessity of calculating and comparing inner product values as required in Steps 3a and 3b. As we will see from our results, this difference is not enough to offset the advantages of the new procedure.

An important concern in the method used in this second phase is the complication that arises from the presence of ties in Steps 3a and 3b of the scheme above. Ties among three or more points are resolved by finding the frame of the points participating in the tie. However, finding just one element of this nested frame problem is sufficient to be able to proceed. A simple sorting as in ‘Preprocessor 1’ of Dulá et al. (1992) will yield such a point. The inclusion of a point in \( \mathcal{F}^E \) means that the current polytope changes its shape and that procedure BUILD is applied to a different object.

5. Computational results

We wish to compare the new procedure we have developed for solution of the frame problem (based on procedure BUILD and LP2) versus the currently used procedure based on repeated application of LP1. Two FORTRAN 77 test codes were constructed utilizing the XMP linear programming routines written and disseminated by Marsten (1981). The code designated OLD (old method) was developed to implement the procedure employing LP1 and the code designated NEW (new method) was developed to implement the procedure based on BUILD and LP2, with the exception that NEW omits the inner product minimization used to identify extreme points (Step 3b of BUILD). The inner product minimization was found to be ineffective in our early testing.

We also designed and implemented an extreme point test problem generator specifically for testing purposes. The problem generator produces problem data having a specified number of points, specified dimension, and specified density of extreme points. The basic generation technique is to first randomly generate points on the unit sphere of specified dimension. All such points will be extreme points of the convex hull they define. These points on the sphere are then ‘de-
formed' by applying to each point its own random multiplicative scale factor from a uniform distribution between 1 and 100. This will likely result in several of the deformed points becoming interior points of the resulting convex hull. In general, the lower the dimension, the higher the percentage that will become interior. To insure that exactly the right density of extreme points is achieved, we initially generate a multiple (2 × to 6 ×, depending on the dimension) of the needed extreme points. Embedded in the generator is a subroutine version of NEW which is used to determine exactly which deformed points are still extreme. Excess extreme points are randomly re-
moved until the exact number needed remain. Non-extreme points are then produced from those remaining extreme points using random convex coefficients obtained by normalizing combining coefficients from a uniform distribution between 0 and 1. The points are translated by subtracting their barycenter and then ordered by distance from the origin, after which the points are randomly shuffled, with the exception that the furthest point from the origin is the last data point (so that it will be a known extreme point and also may be used as the starting right hand side for the initialization scheme used with NEW).

For comparison purposes we have used a test suite of 48 random extreme point test problems produced by our generator. We have utilized all combinations of extreme point densities at 20%, 40%, and 60%, dimensions (m) of 5, 10, 15, and 20, and number of points (n) at 125, 250, 375, and 500. All tests were performed using the SEQUENT SYMMETRY S81 making use of the Weitek 1167 floating-point accelerator. We report on our currently fully instrumented versions of these codes which collect extensive statistics. (We also have earlier streamlined versions of these codes which collect minimal statistics and run somewhat faster.) Both codes identified the same points as extreme or as nonextreme and agreed with the generator.

Overall solution times after the data has been read are shown in Fig. 1 and Tables 1–3. The timings given are each the average of three timing runs. No provisions for special machine loading situations were made.

We used an indexing loop to move sequentially through the data points (an obvious implementation) for the selection of points from $\mathcal{A}$ (Step 1 of BUILD). This also provided two natural alternatives in the choice of direction the loop uses (low to high or high to low). It turns out that the choice of direction has a decided effect on the speed of the algorithms, depending on how many non-extreme points are encountered early in each test case. We have chosen to make all runs using both directions and have reported the average time for the best direction for NEW and the average time for the best direction for OLD in each data combination independently.

### Table 1
20% Extreme point density

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Points</th>
<th>Old method</th>
<th>New method</th>
<th>Ratio new/old</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>125</td>
<td>406.16</td>
<td>160.27</td>
<td>0.39</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>689.34</td>
<td>367.41</td>
<td>0.53</td>
</tr>
<tr>
<td></td>
<td>375</td>
<td>1275.75</td>
<td>634.73</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>2277.65</td>
<td>1244.90</td>
<td>0.55</td>
</tr>
<tr>
<td>15</td>
<td>125</td>
<td>163.62</td>
<td>82.31</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>346.90</td>
<td>180.98</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>375</td>
<td>720.50</td>
<td>385.83</td>
<td>0.54</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>1162.67</td>
<td>628.85</td>
<td>0.54</td>
</tr>
<tr>
<td>10</td>
<td>125</td>
<td>42.26</td>
<td>22.62</td>
<td>0.54</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>156.92</td>
<td>80.55</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td>375</td>
<td>303.85</td>
<td>155.44</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>490.64</td>
<td>269.59</td>
<td>0.55</td>
</tr>
<tr>
<td>5</td>
<td>125</td>
<td>14.84</td>
<td>8.71</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>52.85</td>
<td>29.8</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td>375</td>
<td>101.90</td>
<td>64.36</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>161.20</td>
<td>104.74</td>
<td>0.65</td>
</tr>
</tbody>
</table>

On this set NEW was consistently faster than OLD with time ratios varying from 39% to 65% for problems with 20% extreme point density, from 46% to 67% for problems with 40% extreme point density, and from 38% to 67% for problems

### Table 2
40% Extreme point density

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Points</th>
<th>Old method</th>
<th>New method</th>
<th>Ratio new/old</th>
</tr>
</thead>
<tbody>
<tr>
<td>m</td>
<td>n</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>125</td>
<td>282.25</td>
<td>130.64</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>1001.23</td>
<td>483.71</td>
<td>0.48</td>
</tr>
<tr>
<td></td>
<td>375</td>
<td>1787.50</td>
<td>933.99</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>2911.96</td>
<td>1582.80</td>
<td>0.54</td>
</tr>
<tr>
<td>15</td>
<td>125</td>
<td>145.05</td>
<td>67.28</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>510.81</td>
<td>257.88</td>
<td>0.50</td>
</tr>
<tr>
<td></td>
<td>375</td>
<td>922.69</td>
<td>501.08</td>
<td>0.54</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>1465.33</td>
<td>810.56</td>
<td>0.55</td>
</tr>
<tr>
<td>10</td>
<td>125</td>
<td>60.39</td>
<td>31.95</td>
<td>0.53</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>215.62</td>
<td>121.06</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td>375</td>
<td>392.92</td>
<td>222.35</td>
<td>0.57</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>584.80</td>
<td>339.34</td>
<td>0.58</td>
</tr>
<tr>
<td>5</td>
<td>125</td>
<td>17.67</td>
<td>10.92</td>
<td>0.62</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>65.50</td>
<td>38.60</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td>375</td>
<td>123.04</td>
<td>79.05</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>189.65</td>
<td>126.36</td>
<td>0.67</td>
</tr>
</tbody>
</table>
Table 3
60% Extreme point density

<table>
<thead>
<tr>
<th>Dimension m</th>
<th>Points n</th>
<th>Old method</th>
<th>New method</th>
<th>Ratio new/old</th>
</tr>
</thead>
<tbody>
<tr>
<td>20</td>
<td>125</td>
<td>362.65</td>
<td>138.36</td>
<td>0.38</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>1164.43</td>
<td>556.53</td>
<td>0.48</td>
</tr>
<tr>
<td></td>
<td>375</td>
<td>2185.03</td>
<td>1145.54</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>3434.50</td>
<td>1806.97</td>
<td>0.53</td>
</tr>
<tr>
<td>15</td>
<td>125</td>
<td>193.07</td>
<td>89.67</td>
<td>0.46</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>605.40</td>
<td>284.24</td>
<td>0.47</td>
</tr>
<tr>
<td></td>
<td>375</td>
<td>1119.96</td>
<td>583.74</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>1760.00</td>
<td>937.88</td>
<td>0.53</td>
</tr>
<tr>
<td>10</td>
<td>125</td>
<td>74.12</td>
<td>38.12</td>
<td>0.51</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>256.00</td>
<td>132.60</td>
<td>0.52</td>
</tr>
<tr>
<td></td>
<td>375</td>
<td>473.90</td>
<td>265.83</td>
<td>0.56</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>728.68</td>
<td>422.19</td>
<td>0.58</td>
</tr>
<tr>
<td>5</td>
<td>125</td>
<td>21.42</td>
<td>12.61</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td>250</td>
<td>77.18</td>
<td>45.64</td>
<td>0.59</td>
</tr>
<tr>
<td></td>
<td>375</td>
<td>146.16</td>
<td>93.83</td>
<td>0.64</td>
</tr>
<tr>
<td></td>
<td>500</td>
<td>229.07</td>
<td>151.35</td>
<td>0.66</td>
</tr>
</tbody>
</table>

with 60% extreme point density. The ratios typically increase as number of points increases and decrease as dimension increases. The ratios appear to be relatively insensitive to the extreme point density, with the highest amount of variability appearing across extreme point density at high dimension and low number of points. As might be expected, the total time appears to increase exponentially as both the number of points and the dimension increases.

We have chosen six of these problems to use as illustrations in analyzing why NEW is more efficient than OLD. We have chosen to use the three problems with 125 data points and dimension 20 at extreme point densities of 20%, 40%, and 60%, which gave the best time ratios (38–46%) for NEW versus OLD, and the three problems with 500 data points and dimension 5 at extreme point densities of 20%, 40%, and 60%, which gave the worst time ratios (65–67%) for NEW versus OLD.

Recall that OLD and NEW differ markedly in the size of the LP problems which must be solved. OLD solves n LP problems, starting with a large-size problem having n – 1 columns and as the iterations proceed the problem size steadily decreases. Each time a point not of the frame is identified, the column associated with that data point is eliminated from all subsequent LP problems. NEW is initialized with a single large-size LP problem having n – 1 columns. The point with maximum norm is used as the right-hand side and the remaining n – 1 data points constitute the structural data for this problem. With the solution of this problem, a starting set of m + 1 points of the frame (see Proposition 3) are available. Thereafter NEW performs procedure BUILD until all frame elements are determined. The BUILD portion of NEW solves n – ( M + 1) LP problems, starting with a small-size problem having m + 1 columns and as the iterations proceed the problem size steadily increases. Each time a point of the frame is identified, the data for that point becomes an additional column for the subsequent LP problems. Table 4 shows that the average number of columns per LP solved for NEW versus OLD is 46–59% for the problems with m = 20 and n = 125 and 22–44% for the problems with m = 5 and n = 500.

NEW is uniformly faster than OLD because the time savings for solution of linear programming problems more than offsets the time taken for the sets of inner products which are required by NEW in Step 3a of BUILD. Table 5 shows that only a few thousand additional inner products were required for the problems with m = 20.

Table 4
Average LP column size

<table>
<thead>
<tr>
<th>Problem</th>
<th>Extreme point density</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>20%</td>
</tr>
<tr>
<td></td>
<td>Old</td>
</tr>
<tr>
<td>m = 20, n = 125</td>
<td>122.7</td>
</tr>
<tr>
<td>m = 5, n = 500</td>
<td>296.8</td>
</tr>
</tbody>
</table>
and \( n = 125 \), while 40 K to 102 K thousand additional inner products were required for the problems with \( m = 5 \) and \( n = 500 \). One reason the relative performance degrades for problems with few dimension and many points is the substantial increase in the number of inner product operations.

The relative size of the LP problems solved by NEW and OLD have an effect on the number of simplex iterations needed. Table 6 shows the average number of simplex iterations per LP problem for NEW versus OLD, with ratios in the range 51–66\% for the problems with \( m = 20 \) and \( n = 125 \) and 60–76\% for the problems with \( m = 5 \) and \( n = 500 \).

The difference in the number of simplex iterations per LP solved manifested between the two methods can be explained by the fact that there are actually two types of linear programs being solved. NEW often performs an LP solution which is a reoptimization. When a frame element is identified by use of inner products, it is sometimes not the point whose data is the right-hand side. When this occurs, a single new column is added to the previous problem data and the LP resolved. Such a solution is termed a reoptimization and often requires only a few iterations. Table 7 shows that reoptimizations occurred on 13–40\% of the LP solutions for the problems with \( m = 20 \) and \( n = 125 \) and on 8–19\% of the LP solutions for the problems with \( m = 5 \) and \( n = 500 \). Further, the average number of simplex iterations per reoptimization varied from only 1.8 to 4.8 over all of the problems.

Preprocessing schemes can result in substantial speedups for both NEW and OLD (see Dulá et al., 1992) but were not implemented here, since the objective was to compare the new method vs. the established method. There is no reason to believe that comparable computational savings would not be obtained for both methods if preprocessing schemes were to be employed in both codes.
6. Concluding remarks

We have presented a new procedure for finding the frame of the convex hull of a finite set of points. The principal advantage of this approach over previous ones is that the size of the linear program begins small and grows in columns to no more than the number of elements in the frame. This is in contrast to solving the same number of linear programs with one less column than that there are points. This work represents a fundamental contribution in many areas and will enhance their usefulness by increasing performance limits. In some areas such as DEA the possible impact can be substantial since there are currently large problems which are beyond the reach of current resources available to most analysts in this area.

References


