Identifying the edges of a convex hull

Francisco J. López¹ and Art M. Duval²

 francisco.lopez@maconstate.edu. School of Business 100 College Station Dr, Macon, GA 31206

2. artduval@math.utep.edu. Department of Mathematical Sciences University of Texas at El Paso, El Paso TX 79968

Abstract. We present three new algorithms for identifying all one-dimension faces of convex hulls. They emerge from theoretical and geometric characteristics of the hulls. The computational efficiency of these procedures is compared to the best currently available algorithm (to the best knowledge of the authors) that solves the same problem. Two of the methods are efficient, and can be further accelerated with speed-up techniques, as we verified on one of them.

Key words. computational geometry, computing methodologies and applications, convex hulls, polyhedra

1. Introduction.

An important research field involving convex hulls (or other types of hulls) refers to the identification of their faces; *e.g.*, Wets and Witzgall (1967); Dulá, Helgason, and Venugopal (1998); Erickson (1999), or Ottmann, Schuierer, and Soundaralakshimi (2001). This problem is not to be confused with that of identifying the facets and faces of convex polyhedral cones or polytopes (convex cones or convex polytopes) that result from systems of equalities and/or inequalities, and which are closely related (via duality or polarity) to the problem of interest in this article. In other words, we are finding the edges of polytopes is abundant, but this area is out of the scope of this work. For more information on the relationships between hulls and convex polyhedra see Wets and Witzgall (1967); Bremner, Fukuda, and Marzetta (1998); and Fukuda (2004, 2005).

This article introduces three new approaches that find all 1-faces (the edges) of the convex hull of a set of points. A k-face of a hull is a "face of dimension k" (Wets and Witzgall (1967)).

2. Definitions and assumptions.

Let $\mathcal{P} = \{p^1, p^2, \dots, p^n\}$ be a set of *n* points in *m*-dimensional space. The convex hull of \mathcal{P} , con(\mathcal{P}), is the set of all convex combinations of the points in \mathcal{P} : con(\mathcal{P}) = $\{z \in \Re^m | z = \sum_{i=1}^n p^i \lambda_i; 0 \leq \lambda_i \in \Re; \text{ and } \sum_{i=1}^n \lambda_i = 1\}$ (Rockafellar (1970)). For the purpose of this work, points and vectors are synonyms. We assume there are no duplicated points in \mathcal{P} and each point is an extreme point of con(\mathcal{P}). It is possible to identify the extreme points of a convex hull with the traditional approach or with more efficient algorithms available in the literature; *e.g.*, Dulá and López (2006).

The symbols β , ϵ , and γ are scalars, while π , x, y, u, and u^{ij} are vectors where π is in \Re^m and x, y, $u = (1, 1, ..., 1)^T$, and u^{ij} are in \Re^n (u^{ij} is a vector with ones, like u, except that the *i*-th and *j*-th coordinates are zeros), with $1 \le i \le n$, $1 \le j \le n$, and $i \ne j$. The *i*-th coordinate of point π is denoted π_i .

P is the $m \times n$ matrix with columns p^1, p^2, \ldots, p^n ; *i.e.*, $P = [p^1 p^2 \ldots p^n]$. The two extreme points at the end of an edge are "*adjacent*." We employ Matrix *A* with dimension $n \times n$ in our algorithms to indicate whether p^i and p^j are adjacent. If i < j, the element a_{ij} in the *i*-th row and *j*-th column of *A* equals 1 if p^i and p^j are adjacent, and equals 0 otherwise.

 $\mathcal{H}(\pi,\beta)$, is a hyperplane with defining vector (normal) π and level value β . A supporting hyperplane of $\operatorname{con}(\mathcal{P})$ is a hyperplane $\mathcal{H}(\pi,\beta)$ such that $\pi^T p^i \leq \beta$ for all $p^i \in \mathcal{P}$ and $\pi^T p^j = \beta$ for at least one point $p^j \in \mathcal{P}$. This means that $\mathcal{H}(\pi,\beta)$ touches $\operatorname{con}(\mathcal{P})$ at least at one point and keeps the entire hull in one of the closed halfspaces that it defines.

Points $p^i \in \mathcal{P}$ and $p^j \in \mathcal{P}$ are adjacent if and only if there is a supporting hyperplane of $\operatorname{con}(\mathcal{P})$, $\mathcal{H}(\tilde{\pi}, \tilde{\beta})$, such that $\tilde{\pi}^T p^i = \tilde{\pi}^T p^j = \tilde{\beta}$, and $\tilde{\pi}^T p^k < \tilde{\beta}$ for every point $p^k \in \mathcal{P}$ different from p^i and p^j . This hyperplane contains p^i and p^j but does not contain any other point from \mathcal{P} . Figure 1 illustrates these concepts.

3. Background.

Most of the research on faces/facets of hulls focuses on identifying their extreme points; e.g., Rosen, Xue, and Phillips (1992); Dulá and Helgason (1996); or Dulá, Helgason, and Venugopal (1998). Work on identifying higher dimensional faces is mostly about finding the facets (top-dimensional faces), or on finding *all* the higher dimensional faces; see Seidel



Figure 1: A convex hull and example of an edge and a supporting hyperplane.

(1997). An exception is an article by Wets and Witzgall (1967), who provide two algorithms for positive hulls that can be used to find the edges of convex hulls. Both procedures require transforming the convex hull into a positive hull by adding a new dimension and assigning to each point the same constant, different from zero, as the new coordinate (see Figure 2). Then, the positive hull of the points (vectors) in the expanded dimension is employed because there is a one-to-one relation between the faces of the two hulls: any k-face of the positive hull corresponds to one and only one (k-1)-face of the convex hull and vice versa. Extreme rays of the positive hull correspond to extreme points of the convex hull and 2-faces of the positive hull correspond to edges of the convex hull.

The algorithms by Wets and Witzgall (1967) are useful to find k-faces of positive hulls in general, $1 \leq k \leq m-1$, not only 2-faces. Their first algorithm is based on the idea that, without loss of generality, the points $\hat{p}^1, \ldots, \hat{p}^k$ in $\hat{\mathcal{P}} = \{\hat{p}^1, \ldots, \hat{p}^n\}$ (hats indicate points in the expanded dimension \Re^{m+1}) subdetermine a face of the positive hull of $\hat{\mathcal{P}}$ if and only if the linear hull of $\{\hat{p}^1, \ldots, \hat{p}^k\}$ is the lineality space of the positive hull of $\hat{\mathcal{P}} \cup \{-\hat{p}^1, \ldots, -\hat{p}^k\}$. The second algorithm characterizes the faces of a convex hull in terms of sign patterns of matrices representing the hull. Wets and Witzgall (1967) explain that the latter is expected to be computationally more efficient since it does not have to start from scratch for each decision, but they also warn of the risk of cycling in the presence of degeneracies if degeneracy



Figure 2: Transforming a convex hull in \Re^2 into a positive hull in \Re^3 .

provisions and zero tolerances are not chosen and handled correctly.

4. Three new algorithms for finding the edges of a convex hull.

4.1. The "Repelling-Support" idea. Recall that the definition of adjacency states that two points in \mathcal{P} are adjacent (or determine an edge) if and only if there is a supporting hyperplane of $\operatorname{con}(\mathcal{P})$ that contains both points and keeps (strictly) all other points of \mathcal{P} to one of its sides. This is depicted in Figure 1. The idea is to use a linear program (LP), which we label LP1(ij), to try to find such a supporting hyperplane as follows.

$$\begin{array}{rcl} \max & \epsilon \\ S.T. & \pi^T p^k & -\beta & = 0; & \text{if } k = i \text{ or if } k = j, \\ \pi^T p^k & -\beta & +\epsilon \leq 0; & \text{if } k \neq i, j, \\ & \epsilon \leq 1, \\ \pi, & \beta, & \epsilon, & \text{free in sign.} \end{array}$$

The corresponding dual, named D1(ij), is:

$$\begin{array}{rcl} \min & & \gamma \\ S.T. & Py & = 0, \\ & -u^T y & = 0, \\ & u^{ij^T} y & + \gamma & = 1, \end{array}$$

 y_i, y_j , free in sign; $y_k \ge 0$ for all $k \ne i, j$; $\gamma \ge 0$.



Figure 3: Projection of a set of points onto a selected hyperplane.

4.1.1. The "Repelling-Support" algorithm. Let $LP1(i, j), i \neq j$, be the "Repelling-Support" LP formulation and D1(i, j) be its dual LP, and let a star, "*", indicate optimality.

"Repelling-Support" pseudo-code for adjacency

Input: m, n, P; Output: A. Initialization: A = 0. For i = 1, to n - 1, For j = i + 1, to n, Solve D1(i, j), $a_{ij} = \gamma^*$, Next j, Next i,

STOP. The matrix A is such that $a_{ij} = 1, i < j$, if and only if p^i and p^j are adjacent.

Solving an LP with "few" rows and "many" columns is faster than solving one with "many" rows and "few" columns. Since n is usually significantly greater than m, we solve D1(i, j) instead of LP1(i, j).

4.2. The "projection" idea. This approach projects the points in \mathcal{P} onto hyperplanes properly selected in order to identify the edges of $\operatorname{con}(\mathcal{P})$. An example of such projection appears in Figure 3, where all projections occur following the direction $\pi^{12} = p^1 - p^2$.

Let $\pi^{ij}, i \neq j$, be the vector $p^i - p^j$ and recall $\mathcal{H}(\pi^{ij}, \beta)$ is the hyperplane defined by π^{ij} with level value β . For any point $p \in \Re^m$, let \overline{p} be the projection of point p on hyperplane $\mathcal{H}(\pi^{ij}, 0) : \overline{p} = p - \lambda_p \pi^{ij}$, where

$$\lambda_p = \frac{\pi^{ij^T} p}{\pi^{ij^T} \pi^{ij}} \in \Re.$$

We will use the notational shortcuts

$$\overline{p}^i := \overline{p^i} \quad \text{and} \quad \overline{\mathcal{P}} := \{\overline{p} : p \in \mathcal{P}\}.$$

Since $\overline{p}^i = \overline{p}^j$, it follows that \overline{p}^i is extreme in the projected hull if and only if \overline{p}^j is also extreme. To determine if \overline{p}^i is an extreme point of $\operatorname{con}(\overline{\mathcal{P}})$, it suffices to solve a phase I LP (Wets and Witzgall (1967)). We call our formulation LP2(i, j) and it is as follows.

$$\begin{array}{rcl} \max & 0\\ S.T. & \overline{P}x & = \overline{p^i},\\ & x_1 + \ldots + x_n & = 1,\\ & x_i = x_j = 0; \ x \ge 0, \end{array}$$

where \overline{P} is the matrix whose columns are the points in $\overline{\mathcal{P}}$.

4.2.1. The "Projection" algorithm. Let LP2(i, j) be the LP that determines whether \overline{p}^i is extreme when projections occur onto $\mathcal{H}(\pi^{ij}, 0)$.

"Projection" pseudo-code for adjacency

Input: m, n, P; Output: A. Initialization: A = 0. For i = 1, to n - 1, For j = i + 1, to n, $\overline{\mathcal{P}} = \emptyset; \ \overline{P} = 0,$ $\pi^{ij} = p^i - p^j,$ For k = 1, to n, $\overline{p}^{k} = p^{k} - \left(\frac{\pi^{ij^{T}}p^{k}}{\pi^{ij^{T}}\pi^{ij}}\right)\pi^{ij},$ $\overline{\mathcal{P}} = \overline{\mathcal{P}} \cup \{\overline{p}^{k}\},$ Next k, $\overline{P} \leftarrow \overline{\mathcal{P}},$ Solve LP2(i, j), If LP2(i, j) is not feasible, then $a_{ij} = 1$, Else, continue End if, Next j,

Next i

STOP. The matrix A is such that $a_{ij} = 1, i < j$, if and only if p^i and p^j are adjacent.



Figure 4: Origin translation, resulting positive hull, and intersection of hyperplane with hull.

4.3. The "Conical or Truncation" idea. Two ideas to find the edges of a convex hull emerge from translating the origin to a point in \mathcal{P} . Figure 4(a) illustrates how the extreme vectors of the positive hull after translating the origin to $p^{\tilde{i}}$ correspond to edges of the convex hull. Also, if a hyperplane cuts off $p^{\tilde{i}}$ from all the remaining points, Figure 4(b), the intersection of the hyperplane with the convex hull is another convex hull whose extreme points correspond to edges of the whole convex hull. An approach based on the first idea (the "conical" approach) is both easier to implement and computationally more efficient.

For the purpose of our algorithm it suffices to iteratively check the feasibility of the following LP (phase I LP), which we call LP3(ij).

$$\begin{array}{rcl} \max & 0\\ S.T. & P^i x & = b,\\ & x_i = x_j = 0; & x \ge 0, \end{array}$$

where $b = p^j - p^i$, $i \neq j$, $1 \leq i \leq n$, $1 \leq j \leq n$, and P^i is the matrix which columns are the vectors in $\mathcal{P}^i = \{p^1 - p^i, p^2 - p^i, \dots, p^n - p^i\}$ (the origin is translated to p^i).

4.3.1. The "Conical" algorithm.

"Conical" pseudo-code for adjacency

Input: m, n, P; Output: A. Initialization: A = 0. For i = 1, to n - 1, Initialization: $\mathcal{P}^i = \emptyset$; $P^i = 0$, For k = 1, to n, $\mathcal{P}^i = \mathcal{P}^i \cup \{p^k - p^i\}$, Next k,
$$\begin{split} P^i \leftarrow \mathcal{P}^i, \\ \text{For } j &= i+1, \text{ to } n, \\ b &= p^j - p^i, \\ \text{Solve } LP3(i,j), \\ \text{If } LP3(i,j) \text{ is not feasible, then } a_{ij} = 1, \\ \text{Else, continue,} \\ \text{End if,} \\ \text{Next } j, \end{split}$$

Next i,

STOP. The matrix A is such that $a_{ij} = 1, i < j$, if and only p^i and p^j are adjacent.

<u>Note on speeding up the algorithms.</u> It is important to be aware of techniques that accelerate algorithms, like preprocessors, LP warm-starts, Restricted Basis Entry (RBE) (Ali (1993)), the algorithm by Dulá and López (2006) in the case of the Conical algorithm, and more. For illustration purposes we accelerated the Conical approach with algorithm "PolyFrame", by Dulá and López (2006). We implemented PolyFrame directly, but the algorithms can be further accelerated with preprocessors and other techniques. The impact is significant CPU times reductions. In the remainder of this work "Naive-Conical" and "Frame-Conical" refer to the naive (or "pure") and the PolyFrame implementations, respectively.

5. Computational results and Conclusions.

Figures 5 and 6 illustrate the behavior of CPU times of the algorithms, including the Frame-Conical approach, depending on cardinality and dimension changes, respectively.

To summarize, this article describes three algorithms that identify the edges (or 1dimension faces) of finitely generated convex hulls. We provide LP formulations that detect important characteristics of the corresponding convex hulls. We test the algorithms computationally to verify their effectiveness and to compare them to an algorithm proposed by Wets and Witzgall (1967) that solves the same problem. The latter is, to the best of the authors' knowledge, the best currently available algorithm for this purpose, but we, as Wets and Witzgall (1967), also experienced problems with this approach since we did not implement special handling of degeneracy or zero tolerances. The missing data in our figures are due to the impossibility of recording the corresponding times.



Figure 5: Effect of cardinality on CPU times.

Two of the algorithms (the Projection and the Conical methods) emerge as the best performers, especially for large problems. In the case of small problems (mainly in low dimensions) the approach by Wets and Witzgall is the fastest, perhaps for the benefit of not having to start from scratch each iteration. We also illustrate how to boost the performance of the Conical approach with an algorithm by Dulá and López (2006). All three algorithms, as well as the improved Conical approach, are subject to further improvements by using preprocessors and accelerators.



Figure 6: Effect of dimension on CPU times.

References

- I. ALI, Streamlined computation for data envelopment analysis, Europ. J. of Oper. Res., 64 (1993), pp. 61-67.
- D. BREMNER, K. FUKUDA, AND A. MARZETTA, *Primal-dual methods for vertex and facet* enumeration, Discrete Comput. Geom., 20 (1998), pp. 333-357.
- J.H. DULÁ AND R.V. HELGASON, A new procedure for identifying the frame of the convex hull of a finite collection of points in multidimensional space, Europ. J. of Oper. Res., 92 (1996), pp. 352-367.
- J.H. DULÁ, R.V. HELGASON, AND N. VENUGOPAL, An algorithm for identifying the frame of a pointed finite conical hull, INFORMS J. Comput., 10 (1998), pp 323-330.
- J.H. DULÁ AND F.J. LÓPEZ, Algorithms for the frame of a finitely generated unbounded polyhedron, INFORMS J. Comput., 18-1 (2006), pp 97-110.
- J. ERICKSON, New lower bounds for convex hull problems in odd dimensions, SIAM J. Comput., 28-4 (1999), pp. 1198-1214.
- K. FUKUDA, Frequently Asked Questions in Polyhedral Computation, http://www.ifor.math.ethz.ch/~fukuda/polyfaq/polyfaq.html, version June 18, 2004.

- K. FUKUDA, Introduction, cdd/cdd+ Reference Manual, ftp://ftp.ifor.math.ethz.ch/pub/fukuda/cdd/cddman/node2.html, version August 18, 2005.
- T. OTTMANN, S. SCHUIERER, AND S. SOUNDARALAKSHIMI, Enumerating extreme points in higher dimension, Nordic J. on Comput., 8 (2001), pp 179-192.
- R.T. ROCKAFELLAR, *Convex Analysis*, Princeton University Press, Princeton, New Jersey, 1970.
- J.B. ROSEN, G.L. XUE, AND A.T. PHILLIPS, Efficient computation of extreme points of convex hulls in R^d, in Advances in Optimization and Parallel Computing, P.M. Pardalos, ed., North Holland, Amsterdam, The Netherlands, 1992, pp. 267-292.
- R. SEIDEL, Convex hull computations, in *Handbook of Discrete and Computational Geometry*, J.E. Goodman and J. O'Rourke, eds., CRC Press, Boca Raton, Florida, 1997, pp. 361-375.
- R.J.-B. WETS AND C. WITZGALL, Algorithms for frames and lineality spaces of cones, J. of Res. National Bureau of Standards – B. Math. and Mathematical Physics 71B (1967), pp. 1-7.