Pergamon

Computers Math. Applic. Vol. 29, No. 7, pp. 23–29, 1995

0898-1221/95 $9.50 + 0.00

0898-1221(95)00015-1

Linear Programming Approaches to the Convex Hull Problem in R^m

P. M. PARDALOS
Department of Industrial and Systems Engineering
University of Florida, Gainesville, FL 32611, U.S.A.
pardalos@math.ufl.edu

Y. LI
Department of Computer Science, The Pennsylvania State University
University Park, PA 16802, U.S.A.

W. W. HAGER
Department of Mathematics, University of Florida
Gainesville, FL 32611, U.S.A.

(Received April 1994; accepted November 1994)

Abstract—In this paper, two linear programming formulations of the convex hull problem are presented. Each formulation is the dual of the other. Linear programming problems that identify a face of the convex hull are also discussed. An efficient algorithm is developed. Computational results obtained on an IBM 3090 are presented.

Keywords—Convex hull, Extreme points, Complexity, Linear programming, Divide-and-conquer.

1. INTRODUCTION

Let $A = \{a_1, \ldots, a_n\}$ be a finite set of distinct points in R^m. The convex hull of a set is the smallest convex set that contains it. For a closed set like A, the convex hull is also the intersection of all closed half-spaces containing it. An element of A is an *extreme point* if it is not a convex combination of other points in A. If E denotes the set of extreme points of A, then any element of A is a convex combination of points in E. Using the set E and some adjacency information, we can identify the boundary of the set A. There are two versions of the *convex hull problem*:

(1) Compute the set of extreme points of A.

(2) Compute the set of extreme points of A and the adjacency information.

Of course, version (2) of the problem is more difficult than version (1). Although we provide linear program formulations of version (2), we primarily focus on version (1). When we use the term convex hull problem in this paper, it is in the sense of (1).

As one of the central problems in computational geometry, the convex hull problem has received much attention in the literature [1–8]. This problem is relevant to many fields, including pattern recognition [9,10], operations research [11], and statistics [2,12].

Recently, much progress has been achieved on the convex hull problem, especially for lower dimensional spaces (see, e.g., [4,13], and the summary contained in [5]). However, in higher dimensional spaces, the problem is difficult computationally, and few results have been reported. In [7], a linear programming type algorithm for the convex hull problem is developed. Numerical experiments for dimensions ranging from 2 to 5 show that their algorithm is quite efficient, and the running time is almost linear in the number of points.

This paper explores linear programming formulations of the convex hull problem further. It is organized in the following way: In Section 2, we present several linear programming formulations of the convex hull problem, and we discuss the different formulations in terms of computational efficiency, and theoretical interest. In Section 3, we extend the results obtained in Section 2 to obtain adjacency information about the convex hull. In Section 4, we present an efficient algorithm, based on the scheme proposed in [7]. Computational results obtained on an IBM 3090 are reported in Section 5.

2. LINEAR PROGRAMMING FORMULATIONS OF THE CONVEX HULL PROBLEM

We use the following notation throughout the paper:

$\mathrm{CO}(S)$ denotes the convex hull of a set S.
I denotes the set $\{1, 2, \ldots, n\}$.
A_J denotes the set $\{a_j \mid j \in J\}$ for any $J \subseteq I$.

Given $a \in A$, the problem of determining whether a is then an extreme point of A is denoted $\mathrm{EXT}(A, a)$. The convex hull of the set A can be found by solving $\mathrm{EXT}(A, a)$ for each $a \in A$. We give two methods for solving $\mathrm{EXT}(A, a)$, based on different definitions of the convex hull. Each method is the dual of the other.

2.1. An Analytic Formulation of the Convex Hull Problem

The analytic formulation of the convex hull problem is related to the definition of an extreme point. Let a_j denote an element of A. Since an extreme point of A is not a convex combination of the other points of A, we consider the following linear program (see also [7]):

$$\text{LP1:} \qquad \min x_j \quad \text{s.t.} \quad \sum_{i \in I} x_i a_i = a_j, \quad \sum_{i \in I} x_i = 1, \qquad x_i \geq 0 \ \forall i \in I.$$

With regard to LP1, we have the following result from [7].

THEOREM 1. *The solution of LP1 is positive if and only if a_j is an extreme point of A.*

PROOF. Observe that for an nonextreme point, 0 is always a solution for LP1 as well as an optimal solution. While for an extreme point, 1 is the only feasible solution, hence an optimal solution of LP1 (also see the proof of Theorem 2.1 of [7]). ∎

2.2. A Geometric Formulation of the Convex Hull Problem

Since the convex hull of A is the intersection of the half spaces containing it, an element of A is an extreme point if and only if there exists a hyperplane H containing a_j with all the other elements of A strictly contained on one side of the hyperplane. Since a hyperplane H can be expressed in terms of its normal x and a translation constant, we have Theorem 2.

THEOREM 2. *A point a_j is an extreme point if and only if*

$$\min_{x \in R^m} \max_{i \in I - \{j\}} x^T (a_i - a_j) < 0. \tag{1}$$

PROOF. An element a_j of A is an extreme point if and only if there exists a hyperplane strictly separating a_j from $A - a_j$. Moreover, if x is a normal vector for a hyperplane strictly separating a_j from $A - a_j$, with the x pointing into the half-space containing a_j, then (1) holds. This completes the proof. ∎

Theorem 2 leads to the following linear program for solving EXT(A, a_j):

$$\text{LP2:} \quad \begin{aligned} &\min \sigma \\ &\text{s.t. } x^T(a_i - a_j) \le \sigma, \qquad \forall i \in I - \{j\}, \\ &\quad \sigma \ge -1, \\ &\quad x \in R^m. \end{aligned}$$

Using some linear programming package we minimize σ. The optimal σ is negative if and only if a_j is an extreme point of A. The constraint $\sigma \ge -1$ ensures the existence of a minimum value for σ. In practice, it is convenient to impose the additional constraint $-1 \le x_k \le 1$, $k = 1, \ldots, m$. With this additional constraint, a_j is an extreme point whenever the optimal σ is negative.

2.3. Relationship between the Formulations

In this subsection, we examine the relationship between the analytic and geometric formulations of the convex hull problem. The dual of the linear program LP1 is the following:

$$\text{DLP1:} \quad \begin{aligned} &\max \quad a_j^T w' + w_{m+1} \\ &\text{s.t.} \quad a_i^T w' + w_{m+1} \le 0, \qquad \forall i \in I - \{j\}, \\ &\qquad a_j^T w' + w_{m+1} \le 1, \\ &\text{where } w = (w_1, \ldots, w_m, w_{m+1}) \in R^{m+1} \quad \text{and} \\ &\qquad w' = (w_1, \ldots, w_m). \end{aligned}$$

Defining $z = -(a_j^T w' + w_{m+1})$, DLP1 can be written

$$\begin{aligned} &\min z \\ &\text{s.t. } y^T(a_i - a_j) \le z, \qquad \forall i \in I - \{j\}, \\ &\quad z \ge -1, \\ &\quad y \in R^m. \end{aligned}$$

Thus, DLP1 is equivalent to LP2. In summary, we have the following theorem.

THEOREM 3. *The analytic formulation LP1 and the geometric formulation LP2 of the convex hull problem are the dual of each other.*

In LP1, the number of constraints is $O(m)$ while the number of variables is $O(n)$. Linear programs like LP1 can be solved efficiently by a variety of linear programming packages. A linear program like LP2 with $O(n)$ inequality constraints and $O(m)$ variables, with n much larger than m, can be solved in time proportional to m, assuming n is fixed, using Megiddo's algorithm (see [14]).

3. ADJACENCY INFORMATION

The formulations of the previous section can be extended to test if a subset of A forms a face of the convex hull of A. In order to do that, it is useful to define the concept of *extreme sets.*

DEFINITION 1. *A subset B of the set A is called an extreme set of A if $CO(B) \cap CO(A - B) = \emptyset$.*

A face F of CO(A) is an extreme set contained in a hyperplane H with $H \cap (A - F) = \emptyset$.

3.1. Linear Programming Formulations for Extreme Sets

Let EXTSET(A, A_J) denote the problem of determining whether a set of points $A_J \subset A$ is an extreme set of A. Generalizing LP1, we obtain a linear program for solving EXTSET(A, A_J)

$$\text{LP3:} \quad \begin{aligned} &\min \sum_{j\in J} x_j \\ &\text{s.t.} \sum_{i\in I} x_i a_i = \sum_{j\in J} y_j a_j, \\ &\sum_{i\in I} x_i = 1, \quad \sum_{j\in J} y_j = 1, \\ &x_i \geq 0, \quad i \in I, \quad y_j \geq 0, \quad j \in J. \end{aligned}$$

The following theorem characterizes the relationship between LP3 and the solution of EXTSET(A, A_J).

THEOREM 4. *If z is the optimal value of the objective function of LP3, then $0 \leq z \leq 1$. Furthermore, z is positive if and only if A_J is an extreme set of A.*

PROOF. It is clear from the constraints that

$$0 \leq \sum_{j\in J} x_j = z \leq \sum_{i\in I} x_i \leq 1.$$

And $z = 0$ if and only if $\text{CO}(A_J) \cap \text{CO}(A - A_J)$ is not empty. The theorem follows then from the definition of extreme sets. ∎

Theorem 4 is a generalization of Theorem 1 in which $J = \{j\}$. Similarly, we have the following generalization of Theorem 2.

THEOREM 5. *A_J is an extreme set if and only if*

$$\min_{x\in R^m} \max_{i\in I-J,\, j\in J} x^T(a_i - a_j) < 0.$$

PROOF. The result follows directly from Theorem 2. ∎

Corresponding to Theorem 4, we have the following generalization of LP2:

$$\text{LP4:} \quad \begin{aligned} &\min \sigma \\ &\text{s.t.} \ x^T(a_i - a_j) \leq \sigma, \qquad \forall i \in I - J \quad \text{and} \quad j \in J, \\ &\sigma \geq -1, \\ &x \in R^m. \end{aligned}$$

The optimal value of LP4 is negative if and only if A_J is an extreme set. Similar to the duality relationship between LP1 and LP2, we have Theorem 6.

THEOREM 6. *LP3 and LP4 are the dual of each other.*

3.2. Identifying a Face

Let FACE(A, A_J) denote the problem of determining whether a given set of points A_J is a face of A. That is, A_J lies in a hyperplane H while the remaining elements of A are strictly contained on one side of H. Let $q = |J|$ and let $a_{j_1}, a_{j_2}, \ldots, a_{j_q}$ be an ordering of the points in A_J. Using LP4, we obtain a linear programming problem to solve FACE(A, A_J).

$$\text{LP5:} \quad \begin{aligned} &\min \sigma \\ &\text{s.t.} \ x^T(a_i - a_j) \leq \sigma, \qquad \forall i \in I - J, \quad j \in J, \\ &\qquad x^T(a_{j_k} - a_{j_{k+1}}) = 0, \qquad \forall k = 1, \ldots, q-1, \\ &\qquad \sigma \geq -1, \\ &\qquad x \in R^m. \end{aligned}$$

A_J is a face of the convex hull of A if and only if the optimal σ is negative.

4. A SIMPLE AND EFFICIENT CONVEX HULL ALGORITHM

Using LP1, [7] develops an efficient algorithm for the convex hull problem. The algorithm first constructs C, a superset of the set of extreme points of A, where the size of C is typically small compared to the size of A. The algorithm is briefly described below:

Phase 1: Initialize C and E to be empty; let $a_1, \ldots, a_n$ be the points of A, arranged in decreasing order relative to their distance to the center of the smallest rectangle circumscribing A.

Phase 2: For $i = 1, \ldots, n$, solve $\text{EXT}(C \cup \{a_i\}, a_i)$, and if a_i is an extreme point of C, then $C = C \cup \{a_i\}$.

Phase 3: For each point $c_i \in C$, solve $\text{EXT}(C, c_i)$, and if c_i is an extreme point of C, then $E = E \cup \{c_i\}$.

We now present a modified version of this algorithm that is often much faster. The basic change we make is to terminate phase 2 whenever we encounter P consecutive nonextreme points that are not extreme points of the current C. We then proceed to a new phase 3 in which we test whether the current point is an extreme point of a specially chosen subset of C.

Algorithm CO

Input: A, the set of n points in m-dimensional space.

Output: E, the set of extreme points of A.

Phase 1: Initialize C and E to be empty sets; find the smallest rectangle R, with sides parallel to the coordinate planes, containing A, let d be the center of R, and let r be a vector whose components are the lengths of the sides of R; let $a_1, \ldots, a_n$ be the points in A, arranged in decreasing order relative to their distance from d, where the distance from a point $x \in A$ to d is defined by

$$|x| = \max\left\{ \frac{x_i - d_i}{r_i} \;\middle|\; i = 1, \ldots, m \right\}.$$

Phase 2: For $i = 1, \ldots, n$, solve $\text{EXT}(C \cup \{a_i\}, a_i)$, and if a_i is an extreme point of $C \cup \{a_i\}$, then set $C = C \cup \{a_i\}$; if the last P points tested are all nonextreme points, then go to phase 3, otherwise continue phase 2.

Phase 3: Let C_K be the first K elements of C; for each point a_j remaining in A, let C_j be the set of points c_i in C such that

$$\|a_j - c_i\| \leq s$$

where the norm $\|.\|$ is the Euclidean norm; solve $\text{EXT}(C_K + C_j + \{a_j\}, a_j)$, and if a_j is an extreme point of $C_K + C_j + \{a_j\}$, then $C = C \cup \{a_j\}$.

Phase 4: For each of the points c_j of C, solve $\text{EXT}(C, c_j)$, and if c_j is an extreme point of C, then $E = E \cup \{c_j\}$.

In our numerical experiments, we used the following values for the constants P, K, r that appear in Algorithm CO:

$$P = 20, \quad K = 5m^{m/2}, \quad \text{and} \quad s = \frac{1}{4}\min\{r_i \mid i = 1, \ldots, m\}.$$

Since the set C generated by phase 3 contains the set of extreme points of A, the set of extreme points of C is also the set of extreme points of A.

5. COMPUTATIONAL RESULTS ON AN IBM 3090

Computational results were obtained on an IBM 3090 using linear programming subroutines from the IMSL package. The points tested are normally distributed in a unit cube. The set R in our numerical experiments was also the unit cube.

The results are presented in the following two tables, where the number of dimensions ranges from 2 to 5, and the number of points ranges from 100 to 6400. In both tables, m is the number of dimensions, n is the number of points, T is the running time in unit of seconds, while E is the number of extreme points.

Table 1. Running time of Algorithm CO.

m	$n = 100$	$n = 200$	$n = 400$	$n = 800$	$n = 1600$	$n = 3200$	$n = 6400$
2	0.08	0.15	0.35	0.63	1.43	3.83	5.60
3	0.15	0.29	0.53	0.98	2.23	4.29	11.46
4	0.64	1.80	3.71	6.75	14.91	30.00	68.68
5	1.09	3.51	9.53	22.32	56.59	113.86	245.17

Table 2. Number of extreme points found using Algorithm CO.

m	$n = 100$	$n = 200$	$n = 400$	$n = 800$	$n = 1600$	$n = 3200$	$n = 6400$
2	11	13	20	14	13	14	14
3	37	44	52	59	69	76	110
4	58	92	125	137	192	207	266
5	74	126	195	260	399	498	669

It is clear from Tables 1 and 2 that the number of extreme points is sublinear in the total number of points, which is consistent with the theory of [2].

6. DISCUSSION AND CONCLUDING REMARKS

The results in the previous section show that Algorithm CO is quite efficient. The technique of sorting is very powerful—the best candidates for extreme points are examined first and the the size of the set C remains small throughout the computation. Note that sorting takes a small fraction of the total running time; for example, if $m = 5$ and $n = 6400$, the sorting time is less than 3 seconds.

Although we have only used formulation LP1 to solve the convex hull problem, formulation LP5 of FACE(A, A_J) can be used to construct adjacency information.

REFERENCES

1. S.G. Akl and G.T. Toussaint, A fast convex hull algorithm, *Information Processing Letters* **7** (5), 219–222 (1978).
2. J.L. Bentley, H.T. Kung, M. Scholnick and C.D. Thompson, On the average number of maxima in a set of vectors and applications, *Journal of the Association for Computing Machinery* **25** (4), 536–543 (1978).
3. J.L. Bentley and M.I. Shamos, Divide and conquer for linear expected time, *Information Processing Letters* **7** (2), 87–91 (1978).
4. J.L. Bently, K.L. Clarkson and D.B. Levine, Fast linear expected-time algorithms for computing maxima and convex hull, In *Proceedings of the First ACM-SIAM Symposium on Discrete Algorithms*, (1990) (to appear).
5. H. Edelsbrunner and W. Shi, An $O(n \log n^2)$ time algorithm for the three-dimensional convex hull problem, Report No. UIUCDCS-R-89-1533, Department of Computer Science, University of Illinois at Urbana-Champaign, (1989).
6. R.L. Graham, An efficient algorithm for determining the convex hull of a finite planar set, *Information Processing Letters* **2** (1), 132–133 (1973).
7. J.B. Rosen, G.L. Xue and A.T. Phillips, Efficient computation of convex hull in R^d, Technical Report 89-51, Computer Science Department, University of Minnesota, Minneapolis, MN, (1989).
8. J.B. Rosen, G.L. Xue and A.T. Phillips, Efficient computation of convex hull in R^d, In *Advances in Optimization and Parallel Computing*, (Edited by P.M. Pardalos), pp. 267–292, North-Holland, Amsterdam, (1992).
9. M.M. Mcqueen and G.T. Toussaint, On the ultimate convex hull algorithm in practice, *Pattern Recognition Letters* **3** (1), 29–34 (1985).
10. G.T. Toussaint, A historical note on convex hull finding algorithms, *Pattern Recognition Letters* **3** (1), 21–28 (1985).
11. H. Freeman and R. Shapira, Determining the minimum-area encasing rectangle for an arbitrary closed curve, *Communications of the Association for Computing Machinery* **18** (7), 409–413 (1975).
12. S. Chatterjee and S. Chatterjee, Finding extreme points in multivariate space: Algorithms and data analysis, Manuscript, Department of Computer Science, Northeastern University and Department of Computer Science, New York University, NY, (1989).
13. D.G. Kirkpatrick and R. Seidel, The ultimate planar convex hull algorithm?, *SIAM J. Comput.* **15**, 287–299 (1986).
14. N. Megiddo, Linear programming in linear time when the dimension is fixed, *J. ACM* **31**, 114–127 (1984).
15. D.R. Chand and S.S. Kapur, An algorithm for convex polytopes, *Journal of the Association of Computing Machinery* **17** (1), 78–86 (1970).
16. O.L. Chernykh, Construction of the convex hull of a finite set of points when the computations are approximate, *U.S.S.R. Comput. Maths. Math. Phys.* **28** (5), 71–77 (1988).
17. V. Chvatal, *Linear Programming*, W.H. Freeman and Company, (1983).
18. A.M. Day, The implementation of an algorithm to find the convex hull of a set of three-dimensional points, *ACM Trans. on Graphics* **9** (1), 105–132 (1990).
19. L. Devroye, Moment inequalities for random variables in computational geometry, *Computing* **30**, 111–119 (1983).
20. R.A. Dwyer, On the convex hull of random points in a polytope, *Journal of Applied Probability* **25**, 688–699 (1988).
21. D.T. Lee and F.P. Preparata, Computational geometry—A survey, *IEEE Trans. Computers* **C-33** (12), 1072–1101 (1984); Erratum, **C-34** (6), 584 (1985).
22. F.P. Preparata and M.I. Shamos, *Computational Geometry*, Springer-Verlag, New York, (1985).
23. F.P. Preparata and S.J. Hong, Convex hulls of finite sets of points in two and three dimensions, *Communications of the ACM* **20** (2), 87–93 (1977).