SIAM J. COMPUT.
Vol. 9, No. 2, May 1980

0097-5397/80/0902-0009 $01.00/0

ON THE POLYHEDRAL DECISION PROBLEM*

ANDREW C. YAO† AND RONALD L. RIVEST‡

Abstract. Computational problems sometimes can be cast in the following form: Given a point **x** in R^n, determine if **x** lies in some fixed polyhedron. In this paper we give a general lower bound to the complexity of such problems, showing that $\frac{1}{2}\log_2 f_s$ linear comparisons are needed in the worst case, for any polyhedron with f_s s-dimensional faces. For polyhedra with abundant faces, this leads to lower bounds nonlinear in n, the number of variables.

Key words. adversary strategy, complexity, dimension, edge, face, linear decision tree, lower bound, polyhedron

1. Introduction. Computational problems sometimes can be cast in the following form. Given n numbers $x_1, x_2, \cdots, x_n$, determine if they satisfy some fixed set of linear inequalities, i.e., if the point $\mathbf{x}=(x_1, x_2, \cdots, x_n)$ lies in some "polyhedron". For example, the problem of verifying a maximum element can be stated as "Given $x_1, x_2, \cdots, x_n$, determine if $x_1 \geqq x_i$ for all i." As another example, a version of the minimum spanning tree verification problem is the following: Given a weight function w on the set of edges in a graph G, determine if $w(T_0) \leqq w(T)$ for all spanning trees T of G (T_0 is a fixed spanning tree, and $w(T)$ is the sum of edge weights in T). The aim of this paper is to establish a general lower bound on this type of problems, in terms of some intrinsic characteristics of the polyhedron in question. In contrast to a previous result of this type (Rabin [5]), the present bound can give values larger than the number of variables.

2. Definitions and notations. Let R^n be the space of real n-tuples. A set P in R^n is a *polyhedron* if $P=\{\mathbf{x}|\mathbf{x}\in R^n,\ l_i(\mathbf{x})\leqq 0,\ i=1,2,\cdots,m\}$, where m is an integer, $\mathbf{x}=(x_1, x_2, \cdots, x_n)$, and $l_i(\mathbf{x})=\sum_{1\leqq j\leqq n} c_{ij}x_j - a_i$ for some real numbers c_{ij}, a_i. The *polyhedral decision problem* $B(P)$ is to determine whether $\mathbf{x}\in P$ for any input $\mathbf{x}$. We are interested in the *linear decision tree model* [1], [5], [10]. An algorithm is a ternary tree with each internal node representing a test of the form "$\sum \lambda_i x_i - c : 0$", and each leaf containing a "yes" or "no" answer. For any input, the algorithm proceeds by moving down the tree, testing and branching according to the test results ($<$, $=$, or $>$), until a leaf is reached. At that point, the answer to the question "Is $\mathbf{x}\in P$?" is supplied by the leaf. The *cost* of an algorithm is the height of the tree, i.e., the maximum number of tests made for any input. The *complexity* of $B(P)$ is the minimum cost of any algorithm, and is denoted by $C(P)$.

Faces of a polyhedron. Let $P=\{\mathbf{x}|l_i(\mathbf{x})\leqq 0,\ i=1,2,\cdots,m\}$ be a polyhedron in R^n. To each subset H (maybe $\emptyset$) of $\{1,2,\cdots,m\}$, we define a set $F_H(P)\subseteq R^n$ by $F_H(P)=\{\mathbf{x}|l_i(\mathbf{x})<0$ for each $i\in H$; $l_i(x)=0$ for each $i\notin H\}$. We say that $F_H(P)$ is a *face* of dimension s if the smallest affine subspace of R^n containing $F_H(P)$ has dimension s. (An *affine subspace* is the solution to a set of inhomogeneous equations. See, for example, [6] for more discussions.) The empty face has dimension -1 by convention. Let $\mathscr{F}_s(P)$ be the set of faces of dimension s of P. Note that no two elements of $\mathscr{F}_s(P)$

* Received by the editors July 28, 1978, and in revised form May 20, 1979. This research was supported in part by National Science Foundation under Grants MCS-72-03752 A03, MCS-77-05313, and MCS-75-14294.

† Computer Science Department, Stanford University, Stanford, California 94305.

‡ Laboratory for Computer Science, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139.

overlap. The set of faces $\mathcal{F}_s(P)$ is independent of the choice of $l_i(x)$. That is, if $P=\{\mathbf{x}|l_i'(\mathbf{x})\leqq 0,\ i=1,2,\cdots,m'\}$, the set $\mathcal{F}_s(P)$ constructed using $\{l_i'(\mathbf{x})\}$ is the same as the one constructed using $\{l_i(\mathbf{x})\}$. For an intrinsic definition of faces, see for example [3], [8]. A face of dimension 1 is called an edge, as it is part of a line (agreeing with intuition).

Open polyhedra. A nonempty set Q in R^n is called an *open polyhedron* if $Q=\{x|l_i(x)<0,\ i=1,2,\cdots,m\}$. The concepts of faces and set of faces are defined identically as for polyhedra. More precisely, let $P=\{\mathbf{x}|l_i(\mathbf{x})\leqq 0,\ i=1,2,\cdots,m\}$, then $F_H(Q)=F_H(P)$, $\mathcal{F}_s(Q)=\mathcal{F}_s(P)$.

3. Lower bounds for polyhedral decision problems. Let T be a polygon on the plane. Suppose we are asked to decide if a given point x is inside T by making a series of tests of the form "$\boldsymbol{\lambda}\cdot\mathbf{x}-c$: 0". It is easy to see that about $\log v$ tests are necessary if T has v vertices. Our main result is the following generalization.

THEOREM 1. *Let $P=\{\mathbf{x}|l_i(\mathbf{x})\leqq 0$ for $i=1,2,\cdots,m\}$ be a polyhedron in R^n. Then for each s,*

$$2^{C(P)}\cdot\binom{C(P)}{n-s}\geqq|\mathcal{F}_s(P)|.$$

COROLLARY.

$$C(P)\geqq\tfrac{1}{2}\log_2|\mathcal{F}_s(P)|.$$

Theorem 1 relates the complexity of $B(P)$ to certain "static" combinatorial properties of the polyhedron P. Informally, if a polyhedron P has many edges (or faces), then the theorem says it is difficult to decide whether a point lies in P. The rest of this section is devoted to proving Theorem 1. Note that the corollary follows from Theorem 1 since $\binom{C(P)}{n-s}\leqq 2^{C(P)}$.

We will assume in what follows that P is of dimension n. The following informal argument demonstrates that this can be done without loss of generality. Suppose that $\dim(P)=n'<n$. Let $S\subseteq R^n$ be the smallest affine subspace of R^n containing all of P; thus $\dim(S)=n'$. Now every test $\sum\lambda_ix_i-c$: 0 in R^n either corresponds to a linear test $\sum\lambda_i'x_i'-c'$: 0 in S (where $\mathbf{x}'$ is, for $\mathbf{x}\in S$, $\mathbf{x}$ expressed in a basis for S), or else (if $\{\mathbf{x}\in R^n|\sum\lambda_ix_i=c\}\supseteq S$) the test $\sum\lambda_ix_i-c$: 0 is useful only for determining if $\mathbf{x}\in S$, and not for telling if $\mathbf{x}\in P$ under the assumption that $\mathbf{x}\in S$. Therefore the complexity of determining if an $\mathbf{x}\in R^n$ is in P is at least as great as the complexity of determining if an $\mathbf{x}\in S$ is in P. Since $\dim(S)=\dim(P)$ we are finished with our demonstration.

To prove Theorem 1 we shall adopt the "adversary approach" commonly used in deriving lower bounds for decision trees. We shall design an adversary strategy $\mathcal{A}$ which, for any algorithm, will specify the outcomes for successive queries based on the results of previous queries. The following lemma is essential to the construction of $\mathcal{A}$.

LEMMA 1. *Let $Q=\{\mathbf{x}|p_i(\mathbf{x})<0,\ i=1,2,\cdots,t\}$ be a nonempty open polyhedron, $q(\mathbf{x})=\sum_{i=1}^n\lambda_ix_i-c$ a linear form, $Q_1=Q\cap\{\mathbf{x}|q(\mathbf{x})<0\}$ and $Q_2=Q\cap\{\mathbf{x}|q(\mathbf{x})>0\}$. Then for each s, there exists a $j\in\{1,2\}$ such that Q_j is nonempty, and $|\mathcal{F}_s(Q_j)|\geqq\frac{1}{2}|\mathcal{F}_s(Q)|$.*

Proof of Lemma 1. If $Q_2=\varnothing$, then $Q\subseteq\{\mathbf{x}|q(\mathbf{x})\leqq 0\}$. Since Q is an open set, we must have $Q\subseteq\{\mathbf{x}|q(\mathbf{x})<0\}$. Therefore, $Q_1=Q$, and $j=1$ satisfies the requirements. Similarly, for the case $Q_1=\varnothing$ we can choose $j=2$. It remains to prove the lemma when both Q_1 and Q_2 are nonempty. We shall accomplish this by constructing a 1–1 mapping ψ from $\mathcal{F}_s(Q)$ into $\mathcal{F}_s(Q_1)\cup\mathcal{F}_s(Q_2)$. This then implies that $|\mathcal{F}_s(Q)|\leqq|\mathcal{F}_s(Q_1)|+|\mathcal{F}_s(Q_2)|$. We can then choose a j such that $|\mathcal{F}_s(Q_j)|\geqq\frac{1}{2}|\mathcal{F}_s(Q)|$.

Now we construct ψ. Let $F_H(Q) \in \mathcal{F}_s(Q)$. Define

$$A_1 = F_H(Q) \cap \{\mathbf{x} | q(\mathbf{x}) < 0\},$$
$$A_2 = F_H(Q) \cap \{\mathbf{x} | q(\mathbf{x}) > 0\},$$
$$A_3 = F_H(Q) \cap \{\mathbf{x} | q(\mathbf{x}) = 0\}.$$

Case (1). $A_1 \cup A_2 = \varnothing$. In this case $F_H(Q) \subseteq \{\mathbf{x} | q(\mathbf{x}) = 0\}$. Let us write $Q_1 = \{\mathbf{x} | p_i(\mathbf{x}) < 0, \quad i = 1, 2, \cdots, t+1\}$, with $p_{t+1}(\mathbf{x}) = q(\mathbf{x})$. Clearly $F_H(Q_1) = F_H(Q) \cap \{q(\mathbf{x}) = 0\} = F_H(Q)$. Define $\psi(F_H(Q)) = F_H(Q_1)$.

Case (2). $A_1 \cup A_2 \neq \varnothing$. Assume that $A_1 \neq \varnothing$; the case $A_2 \neq \varnothing$ can be similarly treated. Write as before, $Q_1 = \{\mathbf{x} | p_i(\mathbf{x}) < 0, \; i = 1, 2, \cdots, t+1\}$ with $p_{t+1}(\mathbf{x}) = q(\mathbf{x})$. Define $H' = H \cup \{t+1\}$. Clearly $F_{H'}(Q_1) = F_H(Q) \cap \{x | q(x) < 0\}$ is nonempty and is an s-dimensional face of Q_1.

Define $\psi(F_H(Q)) = F_{H'}(Q_1)$.

It remains to show that the ψ constructed is an 1–1 mapping. It is easily seen that $\psi(F_H(Q)) \subseteq F_H(Q)$. Since all the $F_H(Q)$ in $\mathcal{F}_s(Q)$ are disjoint, it follows that all the $\psi(F_H(Q))$ are disjoint, hence distinct. This completes the proof of Lemma 1. □

It would be interesting to know if the same value of j can be used for every value of s in Lemma 1.

The adversary strategy $\mathcal{A}$. The adversary $\mathcal{A}$ will specify a way to answer questions with the help of a sequence of open polyhedra $V_0, V_1, V_2, \cdots$. Initially, $V_0 = Q$ where $Q = \{\mathbf{x} | l_i(\mathbf{x}) < 0, \; i = 1, 2, \cdots, m\}$. That Q is an open polyhedron (i.e., $Q \neq \varnothing$) is a consequence of the assumption that P has dimension n (see e.g. [8, Lemma (2.3.10)]). When the jth query "$q_j(x)$: 0" is asked, $\mathcal{A}$ has constructed $V_0, V_1, \cdots, V_{j-1}$. The adversary $\mathcal{A}$ will decide the outcome and construct V_j in the following way: Let $Q_1 = V_{j-1} \cap \{\mathbf{x} | q_j(\mathbf{x}) < 0\}$, and $Q_2 = V_{j-1} \cap \{\mathbf{x} | q_j(\mathbf{x}) > 0\}$; by Lemma 1, there is an $i \in \{1, 2\}$ such that $Q_i \neq \varnothing$, and $|\mathcal{F}_s(Q_i)| \geqq \frac{1}{2} |\mathcal{F}_s(V_{j-1})|$; the adversary's answer to the jth query is then "$q_j < 0$" if $i = 1$, and "$q_j > 0$" if $i = 2$; V_j is defined to be Q_i.

Analysis of the adversary strategy. Let $q_j(x)$: 0 ($j = 1, 2, \cdots, t$) be the entire sequence of queries asked by the algorithm faced with outcomes determined by $\mathcal{A}$. Let $\varepsilon_j q_j(\mathbf{x}) < 0$ be the results of the queries ($\varepsilon_j = \pm 1$). Then,

$$V_t = \{\mathbf{x} | l_i(\mathbf{x}) < 0, i = 1, 2, \cdots, m, \; \varepsilon_j q_j(\mathbf{x}) < 0, j = 1, 2, \cdots, t\} \neq \varnothing \tag{1}$$

and

$$|\mathcal{F}_s(V_t)| \geqq \frac{1}{2} |\mathcal{F}_s(V_{t-1})| \geqq \frac{1}{2^2} |\mathcal{F}_s(V_{t-2})| \geqq \cdots \geqq \frac{1}{2^t} |\mathcal{F}_s(V_0)|, \text{ i.e.,}$$

$$|\mathcal{F}_s(V_t)| \geqq \frac{1}{2^t} |\mathcal{F}_s(Q)|. \tag{2}$$

For each $\mathbf{x} \in V_t$, the same leaf in the tree T is reached and the algorithm must say "yes, $\mathbf{x} \in P$". Since the algorithm only knows that $\mathbf{x} \in \{\mathbf{x} | \varepsilon_j q_j(\mathbf{x}) < 0, j = 1, 2, \cdots, t\}$, we have

$$\{\mathbf{x} | \varepsilon_j q_j(\mathbf{x}) < 0, j = 1, 2, \cdots, t\} \subseteq P.$$

As Q is the "largest" open set contained in P, we have

$$\{\mathbf{x} | \varepsilon_j q_j(\mathbf{x}) < 0, j = 1, 2, \cdots, t\} \subseteq Q$$
$$= \{\mathbf{x} | l_i(\mathbf{x}) < 0, i = 1, 2, \cdots, m\}.$$

Therefore, (1) can be written as

$$V_t = \{\mathbf{x} | \varepsilon_j q_j(\mathbf{x}) < 0, j = 1, 2, \cdots, t\}. \tag{3}$$

As there are only t linear functions in (3), there can be at most $\binom{t}{n-s}$ s-dimensional faces of V_t. Therefore,

$$\binom{t}{n-s} \geqq |\mathcal{F}_s(V_t)|. \tag{4}$$

Equations (2) and (4) lead to

$$2^t \cdot \binom{t}{n-s} \geqq |\mathcal{F}_s(V_t)|. \tag{5}$$

As the left-hand side of (5) is an increasing function of t, and $C(P) \geqq t$, we have proved Theorem 1. □

4. Remarks. General discussions on the maximum number of faces that a polyhedron can have are given in [3] and [7]. As there can be $\approx \binom{m}{n-1}$ edges for certain polyhedra defined by m inequalities, the corollary to Theorem 1 establishes a lower bound of order $n \log m$ for, say $m > n^2$, to the corresponding polyhedral decision problem.

It would be interesting to find a "natural" problem in concrete computational complexity for which the bound of Theorem 1 yields a nontrivial (i.e., nonlinear) lower bound. In this regard we mention that, originally, it was hoped that the present approach would lead to an $\Omega(n^2 \log n)$ lower bound to the complexity of the all-pair shortest paths problem. That bound would follow if the *triangular polyhedron* $P^{(n)}$ in $R^{\binom{n}{2}}$, defined as $\{\mathbf{x} | \mathbf{x} = (x_{ij} | 1 \leqq i < j \leqq n); x_{ik} \geqq 0, x_{ij} + x_{jk} \geqq x_{ik}$ for all $1 \leqq i < k \leqq n$ and $1 \leqq j \leqq n\}$ (we define $x_{ii} = 0$ and $x_{ij} = x_{ji}$, if $i > j$), has at least $\exp(cn^2 \log n)$ edges[1]. However, it has recently been shown by Graham, Yao, and Yao [2] that $P^{(n)}$ has less than $\exp(cn^2)$ edges, with the implication that only a cn^2 lower bound can be obtained in this approach.

One candidate for the application of Theorem 1 is the problem of constructing optimal alphabetic trees [4], for which the best algorithm known has an $O(n \log n)$ running time. For a start, what is the number of edges in the polyhedron corresponding to deciding if a complete balanced tree is an optimal alphabetic tree? Another candidate is the verification problem for minimum spanning trees mentioned in the Introduction. It seems difficult, however, to obtain a nonlinear bound in this case, since the number of edges involved is no more than $\exp(cn \log^* n)$ (because the problem can be solved in $O(n \log^* n)$ by Tarjan's result [9]).

[1] See [11] for a proof of this statement. We remark that it was incorrectly stated in [11] that $P^{(n)}$ has provably $\exp(cn^2 \log n)$ edges.

REFERENCES

[1] D. P. DOBKIN, R. J. LIPTON AND S. P. REISS, *Excursions into geometry*, Computer Science Dept. Tech. Rep. 71, Yale University, New Haven, CT, 1976.

[2] R. L. GRAHAM, A. C. YAO AND F. F. YAO, *Information bounds are weak in the shortest distance problem*, Computer Science Dept. Rep. STAN-CS-78-670, Stanford University, Stanford, CA, 1978, J. Assoc. Comput. Mach., to appear.

[3] B. GRÜNBAUM, *Convex Polytopes*, Interscience, New York, 1967.

[4] D. E. KNUTH, *The Art of Computer Programming, Vol.* 3, *Sorting and Searching*, Addison-Wesley, Reading, MA , 1973.

[5] M. O. RABIN, *Proving simultaneous positivity of linear forms*, J. Comput. Systems Sci., 6 (1972), pp. 639–650.

[6] O. SCHREIER AND E. SPERNER, *Modern Algebra and Matrix Theory*, translated by M. David and M. Hausner, Chelsea, New York, 1959.

[7] R. STANLEY, *The upper bound conjecture and Cohn–Macaulay rings*, Studies in Applied Math., 54 (1975), pp. 135–142.

[8] J. STOER AND C. WITZALL, *Convexity and Optimization in Finite Dimension I*, Springer Verlag, New York, 1970.

[9] R. E. TARJAN, *Applications of path compressions on balanced trees*, Computer Science Dept. Rep. STAN-CS-75-512, Stanford University, Stanford, CA, 1975.

[10] A. C. YAO, *On the complexity of comparison problems using linear functions*, Proc. 16th IEEE Ann. Symp. on Switching and Automata Theory, Berkeley, CA, 1975, pp. 85–89.

[11] A. C. YAO, D. M. AVIS AND R. L. RIVEST, *An* $\Omega(n^2 \log n)$ *lower bound to the shortest paths problem*, Proc. 9th ACM Annual Symposium on Theory of Computing, Boulder, CO, 1977, pp. 11–17.