An $\Omega(n^2 \log n)$ Lower Bound to the Shortest Paths Problem††

Andrew C. Yao
Computer Science Department, Stanford University

David M. Avis
Department of Operations Research, Stanford University

Ronald L. Rivest
Laboratory for Computer Science, M.I.T.

Summary.

Let P be a polyhedron with f_s s-dimensional faces. We show that $\Omega(\log f_s)$ linear comparisons† are needed to determine if a point lies in P. This is used to establish an $\Omega(n^2 \log n)$ lower bound to the all-pairs shortest path problem between n points.

1. Introduction.

Let G be an undirected complete graph on n vertices $\{v_1,v_2,\ldots v_n\}$, with a non-negative weight w_{ij} $(i < j)$ assigned to each edge (v_i,v_j). The n×n shortest distance matrix for G is $D = (d_{ij})$ where $d_{ii} = 0$ and d_{ij} $(i \neq j)$ is the minimum weighted path length between v_i and v_j. Several ingenious algorithms have been invented to solve the all-pairs shortest path problem, in which D is to be computed. The classical methods of Dijkstra [2] and Floyd [4] both require cn^3 running time in the worst case, and more recently Fredman [5] gave an algorithm with a worst-case bound $O(n^3(\log \log n)^{1/3}/(\log n)^{1/3})$, which is $o(n^3)$. It is likely that substantially better algorithms (say, $O(n^{2.2})$) do not exist, but no lower bound better than cn^2 is known [8] for general algorithms with branching instructions. (For a straight-line computation with two operations "+" and "min", Kerr [7] showed that cn^3 steps are needed.)

In this paper we prove that $\Omega(n^2 \log n)$ comparisons between linear functions of edge weights are needed in the decision tree model. In fact, $\Omega(n^2 \log n)$ comparisons are required to verify that $D = (d_{ij})$ is the shortest distance matrix for a graph G with $\{w_{ij}\}$. In the process we shall show that $\Omega(\log f_s)$ linear comparisons are necessary to determine if a point is in a polyhedron with f_s s-dimensional faces (see Section 2 for definitions). This general theorem is of interest in itself since (1) it relates the complexity of polyhedral decision problems (e.g. Rabin [9]) to some classical aspect of polyhedrons studied by mathematicians (the number of vertices, faces, etc.), and (2) it is potentially possible to derive from it

† $\Omega(g(x))$ means $\geq cg(x)$ for some positive constant c.

Research Supported by NSF Grants #'s: MCS 72-06336 A04, MCS72-03752 A03 and MCS75-14294

non-linear lower bounds for other computational problems, e.g. constructing minimum-cost spanning trees (although Tarjan's result [15] suggests that a non-linear lower bound on minimum-cost spanning trees may be difficult to obtain).

2. Definitions and Notations

(1) Complexity of finding shortest paths. Consider the all-pairs shortest path problem for a graph with n vertices and weights $\{w_{ij}\}$. We are interested in the linear decision tree model. An algorithm is a ternary tree with each internal node representing a test of the form "$\Sigma\lambda_{ij}w_{ij}:c$", and each leaf containing a set of linear functions $\{f_{ij}, 1 \le i, j \le n\}$ on $n(n-1)/2$ variables. For any input, the algorithm proceeds by moving down the tree, testing and then branching according to the test result, until a leaf is reached. At that point, the shortest distance matrix $D = (d_{ij})$ is given by $d_{ij} = f_{ij}(\vec{w})$. The cost of an algorithm is the height of the tree, and the complexity L_n is defined to be the minimum cost for any algorithm.

(2) Polyhedral Decision Problems.

A set P in R^N is a polyhedron if $P = \{\vec{x} | \vec{x}\epsilon R^N, \ell_i(\vec{x}) \le 0,\ i = 1,2,\ldots,m\}$, where m is an integer, $\vec{x} = (x_1,x_2,\ldots x_N)$, and $\ell_i(\vec{x}) = \sum_{1\le j\le n} c_{ij}x_j$ for some real numbers c_{ij}. We remark that we are restricting attention to homogeneous polyhedra, i.e. cones. The polyhedral decision problem $B(P)$ is to determine whether $\vec{x}\epsilon P$ for an input $\vec{x}$. Here we are also interested in the linear decision tree model (each internal node representing a test $\Sigma\lambda_i x_i:c$), with a "yes" or "no" decision at every leaf. The complexity of $B(P)$ is the minimum height of any decision tree, and is denoted by $C(P)$.

(3) Faces of a Polyhedron. Let $P = \{\vec{x} | \ell_i(\vec{x}) \le 0,\ i = 1,2,\ldots,m\}$ be a polyhedron in R^N. To each subset H (maybe $\emptyset$) of $\{1,2,\ldots,m\}$, we define a set $F_H(P) \subseteq R^N$ by

$F_H(P) = \{\vec{x} | \ell_i(\vec{x}) < 0$ for each $i\epsilon H$,

$\ell_i(x) = 0$ for each $i \notin H\}$. We say that $F_H(P)$ is a face of dimension s if the smallest subspace of R^N containing $F_H(P)$ has dimension s. The empty face has dimension -1 by convention. Let $\mathcal{F}_s(P)$ be the set of faces of dimension s of P. Note that no two elements of $\mathcal{F}_s(P)$ overlap. The set of faces $\mathcal{F}_s(P)$ is independent of the choice of $\ell_i(x)$. That is, if $P = \{\vec{x} | \ell_i'(\vec{x}) \le 0,\ i = 1,2,\ldots,m'\}$, the set $\mathcal{F}_s(P)$ constructed using $\{\ell_i'(\vec{x})\}$ is the same as the one constructed using $\{\ell_i(\vec{x})\}$. For an intrinsic definition of faces, see for example [6,10]. A face of dimension 1 is called an edge, as it is part of a line (agreeing with intuition).

(4) Open Polyhedron. A non-empty set Q in R^N is called an open polyhedron if $Q = \{\vec{x} | \ell_i(\vec{x}) < 0,\ i = 1,2,\ldots,m\}$. The concepts of faces and set of faces are defined identically as for polyhedra. More precisely, let $P = \{\vec{x} | \ell_i(\vec{x}) \le 0,\ i = 1,2,\ldots,m\}$, 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 "$\vec{\lambda}\cdot\vec{x}:c$". It is easy to see that about log v tests are necessary if T has v vertices. The following thereom is a generalization:

Theorem 1. Let $P = \{\vec{x} \mid \ell_i(\vec{x}) \leq 0 \text{ for } i = 1,2,\ldots m\}$ be a polyhedron in R^N. Then for each s,

$$2^{C(P)} \cdot \binom{C(P)}{N-s} \geq |F_s(P)| .$$

Corollary. $C(P) \geq 1/2 \log |F_s(P)|$.

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 $\binom{C(P)}{N-s} \leq 2^{C(P)}$.

We first show that we can assume that in an optimal algorithm each query "$\Sigma\lambda_i x_i : c$" has $c = 0$. Let T be a decision tree for B(P). A node v is said to be <u>inhomogeneous</u> if the associated query "$\Sigma\lambda_i x_i : c$" has $c \neq 0$. Without loss of generality, we shall assume $c > 0$ since we can always ask an equivalent query $\Sigma(-\lambda_i)x_i : (-c)$ otherwise. We shall remove inhomogeneous nodes from T by performing the following operation for each inhomogeneous node v: eliminate v, the " >", and "=" branches of the subtree rooted at v; connect the "<" branch directly to the father of v. The resulting tree T´ clearly has a height no greater than the original tree T , and has no inhomogeneous nodes. It remains to show that T´ is a decision tree algorithm for B(P) . Let $a = \min\{c \mid \Sigma\lambda_i x_i : c$ is associated with some inhomogeneous node in $T\}$, and let $b = \max\{|\lambda_i|\}$ be similarly defined. Then, for each $\vec{x} \epsilon D = \{\vec{x} \mid |x_i| < a/Nb \ \forall i\}$, the decision tree T always branches to the "<" path at each inhomogeneous node. Hence, the tree T´ also works correctly for $x \epsilon D$.
But this implies that T´ also works for all $\vec{x}$, as all the comparisons in T´ are homogeneous and the problem B(P) is homogeneous. We have thus proved that we can assume all queries are of the form "$q(\vec{x}):0$" where $q(\vec{x}) = \Sigma\lambda_i x_i$.

We will assume in what follows that P is of dimension N, i.e. that $\{P\} = F_N(P)$.
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 subspace of R^N containing all of P; thus $\dim(S) = N'$. Now every test $\Sigma\lambda_i x_i : c$ in R^N either corresponds to a linear test $\Sigma\lambda'_i x'_i : c$ in S (where $\vec{x}'$ is, for $\vec{x} \epsilon S$, $\vec{x}$ expressed in a basis for S having the same origin as R^N), or else (if $\{\vec{x} \epsilon R^N \mid \Sigma\lambda_i x_i = c\} \cap S = \emptyset$) the test $\Sigma\lambda_i x_i : c$ is useful only for determining if $\vec{x} \epsilon S$, and not for telling if $\vec{x} \epsilon P$ under the assumption that $\vec{x} \epsilon S$. Therefore the complexity of determining if an $\vec{x} \epsilon R^N$ is in P is at least as great as the complexity of determining if an $\vec{x} \epsilon S$ is in P. Since $\dim(S) = \dim(P)$ we are finished with our demonstration. In any case we should also like to remark that for our application of Theorem 1 to the complexity of the shortest paths problem, this assumption holds.

We shall employ an "Oracle" to help our proof. The following lemma is essential to the construction of the oracle:

Lemma 1: Let $Q = \{\vec{x} | p_i(\vec{x}) < 0, i = 1,2,\ldots,t\}$ be an open polyhedron, $q(\vec{x}) = \sum_{i=1}^{N} \lambda_i x_i$ a linear form, $Q_1 = Q \cap \{\vec{x} | q(\vec{x}) < 0\}$, and $Q_2 = Q \cap \{\vec{x} | q(\vec{x}) > 0\}$. Then for each s, there exists a $j \in \{1,2\}$ such that Q_j is non-empty, and $|F_s(Q_j)| \geq 1/2 |F_s(Q)|$.

Proof of Lemma 1.

If $Q_2 = \emptyset$, then $Q \subseteq \{\vec{x} | q(\vec{x}) \leq 0\}$. Since Q is an open set, we must have $Q \subseteq \{\vec{x} | q(\vec{x}) < 0\}$. Therefore, $Q_1 = Q$, and $j = 1$ satisfies the requirements. Similarly, for the case $Q_1 = \emptyset$ we can choose $j = 2$. It remains to prove the lemma when both Q_1 and Q_2 are non-empty. We shall accomplish this by constructing a 1-1 mapping ψ from $F_s(Q)$ into $F_s(Q_1) \cup F_s(Q_2)$. This then implies that $|F_s(Q)| \leq |F_s(Q_1)| + |F_s(Q_2)|$. We can then choose a j such that $|F_s(Q_j)| \geq 1/2|F_s(Q)|$.

Now we construct ψ. Let $F_H(Q) \in F_s(Q)$. Define

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

Case 1) $A_1 \cup A_2 = \emptyset$: In this case $F_H(Q) \subseteq \{\vec{x} | q(\vec{x}) = 0\}$. Let us write $Q_1 = \{\vec{x} | p_i(\vec{x}) < 0, i = 1,2,\ldots,t+1\}$, with $p_{t+1}(\vec{x}) = q(\vec{x})$. Clearly $F_H(Q_1) = F_H(Q) \cap \{q(\vec{x}) = 0\} = F_H(Q)$

Define $\psi(F_H(Q)) = F_H(Q_1)$.

Case 2) $A_1 \cup A_2 \neq \emptyset$: Assume that $A_1 \neq \emptyset$; the case $A_2 \neq \emptyset$ can be similarly treated. Write as before $Q_1 = \{\vec{x} | p_i(\vec{x}) < 0, i = 1,2,\ldots,t+1\}$ with $p_{t+1}(\vec{x}) = q(\vec{x})$. Define $H' = H \cup \{t+1\}$. Clearly, $F_{H'}(Q_1) = F_H(Q) \cap \{\vec{x} | q(\vec{x}) < 0\}$ is non-empty 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 $F_s(Q)$ are disjoint, it follows that all the $\psi(F_H(Q))$ are disjoint, hence distinct. This completes the proof of Lemma 1.

The Oracle:

The Oracle shall specify a way to answer questions with the help of a sequence of open polyhedra $V_0, V_1, \ldots$. Initially, $V_0 = Q$ where $Q = \{\vec{x} | \ell_i(\vec{x}) < 0, i = 1,2,\ldots,m\}$. At the time of the jth query $q_j(\vec{x}):0$, the oracle has constructed $V_0, V_1, \ldots, V_{j-1}$. The oracle decides the answer for the query in the following way:

let $Q_1 = V_{j-1} \cap \{\vec{x} | q_j(\vec{x}) < 0\}$, $Q_2 = V_{j-1} \cap \{\vec{x} | q_j(\vec{x}) > 0\}$; by Lemma 1, there is an i such that Q_i is non-empty, and $|F_s(Q_i)| \geq 1/2 | F_s(V_j-1)|$;

The oracle's answer is then: $q_j < 0$ if $i = 1$, and
$q_j > 0$ if $i = 2$.

The oracle then defines V_j to be Q_i.

Analysis of the Oracle.

Let $q_j(\vec{x}):0$ $(j = 1,2,\ldots,t)$ be the entire sequence of queries asked by the algorithm under the above oracle, and let $\varepsilon_j q_j(\vec{x}) < 0$ be the results of the queries $(\varepsilon_j = \pm 1)$. Then,

$$V_t = \{\vec{x} \mid \ell_i(\vec{x}) < 0, \quad i = 1,2,\ldots,m, \quad \varepsilon_j q_j(\vec{x}) < 0, \; j = 1,2,\ldots,t\} \neq \emptyset \qquad (1)$$

and

$$|F_s(V_t)| \geq 1/2\, |F_s(V_{t-1})| \geq 1/2^2\, |F_s(V_{t-2})| \geq \ldots \geq 1/2^t\, |F_s(V_0)| \quad .$$

$$|F_s(V_t)| \geq 1/2^t\, |F_s(Q)| \quad . \qquad (2)$$

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

$$\{\vec{x} \mid \varepsilon_j q_j(\vec{x}) < 0, \; j = 1,2,\ldots, t\} \subseteq P \; .$$

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

$$\{\vec{x} \mid \varepsilon_j q_j(\vec{x}) < 0, \; j = 1,2,\ldots t\} \subseteq Q = \{\vec{x} \mid \ell_i(\vec{x}) < 0, \; i = 1,2,\ldots,m\}$$

Therefore, (1) can be written as

$$V_t = \{\vec{x} \mid \varepsilon_j q_j(\vec{x}) < 0, \; j = 1,2,\ldots,t\} \; . \qquad (3)$$

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,

$$\binom{t}{N-s} \geq |F_s(V_t)| \qquad (4)$$

(2) and (4) lead to $2^t \cdot \binom{t}{N-s} \geq |F_s(V_t)|$. (5)

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

General discussions of the maximal number of faces of dimension s that a polyhedron can have are given in [6] and [12]. We now turn our attention to the polyhedron associated with the all-points shortest-paths problem.

4. The Shortest Paths Problem.

In this section we make use of results derived in the previous section to obtain an $\Omega(n^2 \log n)$ lower bound for the shortest paths problem. Theorem 1 can not be directly applied to the shortest paths problem, as the latter is not a polyhedral decision problem. The shortest paths problem is, however, closely related to the following polyhedral decision problem, which is a special case of the verification problem for finding shortest paths .

Verifying the Triangle Inequalities:

Let $P^{(n)}$ be the polyhedron in $R^{n(n-1)/2}$ defined as follows: A vector $w \in R^{n(n-1)/2}$ is written as

$$w = (w_{12}, w_{13}, \ldots, w_{1n}, w_{23}, \ldots w_{2n}, \ldots w_{n-1,n});$$

$$P^{(n)} = \{\vec{w} \mid w_{ik} > 0, \quad \ell_{ijk}(\vec{w}) > 0 \;\text{ for }\; i < k, \; i \neq j \neq k\}$$

where $\ell_{ijk}(\vec{w}) = w_{ik} - w_{ij} - w_{jk}$. The problem $B(P^{(n)})$ is to determine if $\vec{w} \in P^{(n)}$, i.e. whether all the w_{ij}'s are positive and all the triangle inequalites are satisfied by (w_{ij}).

The following lemma relates the complexity for shortest paths L_n to the complexity of $B(P^{(n)})$:

Lemma 2: $L_n \geq C(P^{(n)}) - n(n-1)/2$

Proof: Let T be an optimal decision tree algorithm for computing the shortest distance matrix (d_{ij}) from the input matrix (w_{ij}). The height of T is L_n, by definition of L_n. We can obtain a decision tree T´ for the problem $B(P^{(n)})$ by modifiying T as follows. Replace each leaf of T by a sequence of $n(n-1)/2$ distinct tests of the form "Is $d_{ij} = w_{ij}$?" Since at each leaf of T we have $d_{ij} = f_{ij}(\vec{w})$, T´ is a linear decision tree. We construct T´ so that $\vec{w}$ is accepted iff all of the newly added tests have "yes" answers. The correctness of T´ is ensured by the fact [8, page 89] that a matrix is a shortest distance matrix iff it satisfies all the triangle inequalities and by the fact that if all the triangle inequalities are satisfied without equality the matrix is positive. Hence $L_n + n(n-1)/2 \geq C(P^{(n)})$.

To obtain an explicit bound on L_n, we need a recent combinatorial result of Avis, which states that $P^{(n)}$ has a very large set of edges.

Lemma 3: There exists a positive constant c so that $|F_1(P^{(n)})| \geq 2^{n^2(\log n - c \log\log n)/4}$, for all n .

Proof: This counting argument is given in [1], and is omitted here due to its length.

Theorem 2: $L_n \geq n(\log n - c \log\log n)/4$,

for some constant $c' > 0$.

Proof: By Theorem 1 and Lemma 3 we know that

$$C(P^{(n)}) + \log \binom{C(P^{(n)})}{N-1}$$
$$\geq n^2(\log n - c \log\log n)/4 ,$$

where $N = \binom{n}{2}$. Since $\binom{a}{b} \leq a^b/b!$, if x satisfies

$$x + (N-1)\log x - \log((N-1)!)$$
$$= n^2(\log n - c \log\log n)/4 \qquad (6)$$

it must also satisfy $x \leq C(P^{(n)})$. Now (6) implies

$$x + (n^2/2 - n/2 - 1)\log x$$
$$= (5n^2\log n)/4 - (cn^2\log\log n)/4 + O(n^2) \qquad (7)$$

since $\log((N-1)!) = (N-1/2)\log(N-1) - (N-1)\log e + O(1)$ (Stirling's approximation), and $\log(N-1) = \log(n^2) - O(1)$. The solution to (7) is

$$x = (n^2\log n)/4 - (cn^2\log\log n)/4 + O(n^2) \qquad (8)$$

Using (8) and Lemma 2 we can conclude that

$$L_n \geq C(P^{(n)}) - \binom{n}{2}$$
$$\geq x - \binom{n}{2}$$
$$\geq (n^2\log n)/4 - (cn^2\log\log n)/4 - O(n^2)$$
$$\geq (n^2\log)/4 - (c'n^2\log\log n)/4$$

for some $c' > c$.

5. Remarks.

(1) We have shown that $L_n \geq (n^2\log n)/4 - (c \log\log n)/4$, and a $\Omega(n^2\log n)$ bound is the best we can obtain under this approach as $\log|F_s(P)| \leq cn^2 \log n$ for all s.

The best upper bound know (Fredman [5]) is $L_n \leq cn^{2.5}$. Hence a large gap still exists even in this decision tree model.

(2) The linear decision tree model has received considerable attention in the recent

literature ([3],[5],[7],[11],[14],[15]). This model only counts the number of branchings, and thus tends to underestimate the total running time (for example, it is conceivable that no shortest-paths algorithm can achieve $cn^{2.5}$ in total running time). Nevertheless, the linear decision tree model enables us to study non-trivial lower bounds, and Theorem 1 has added yet another useful device in this model.

References

[1] D.M. Avis, "Some Polyhedral Cones Related to Metric Spaces," Ph.D. Thesis, Department of Operations Research, Stanford University, (April 1977).

[2] E.W. Dijkstra, "A note on two problems in connection with graphs," Numer. Math., 1 (1959), 269-271.

[3] D.P. Dobkin, R.J. Lipton, and S.P. Reiss, "Excursions Into Geometry," Yale University Research Report # 71.

[4] R.W. Floyd, "Algorithm 97: Shortest path," C.ACM 5 (1962), p. 345.

[5] M.L. Fredman, "New bounds on the complexity of the shortest path problem," Siam J. on Computing 5 (1976), 87-89.

[6] B. Grünbaum, Convex Polytopes, (Interscience, New York, 1967).

[7] L.R. Kerr, "The effect of algebraic structure on the computational complexity of matrix multiplications," Ph.D Thesis, Cornell University, (1970).

[8] E. Lawler, Combinatorial Optimization: Networks and Matroids, (Holt, Rinehart and Winston, New York 1976).

[9] M.O. Rabin, "Proving simultaneous positivity of linear forms," JCSS 6 (1972), 639-650.

[10] R. Rockfellar, Convex Analysis, Princeton University Press, 1970

[11] P.M. Spira and A. Pan, "On finding and updating spanning trees and shortest paths," SIAM J. on Computing 4 (1975), 375-380.

[12] R. Stanley, "The Upper Bound Conjecture and Cohen-Macaulay Rings," Studies in Applied Math. (Vol LIV, No. 2), (MIT, June 1975), 135-142.

[13] R. Tarjan, "Applications of Path Compression on Balanced Trees," Stanford Computer Science Department Report STAN-CS-75-512, (August 1975).

[14] A.C. Yao, "On the complexity of comparison problems using linear functions," Conference Record, IEEE 16th Annual Symposium on Switching and Automata Theory, 1975, 85-89.

[15] A.C. Yao, "On Computing the Minima of Quadratic Forms," The 7th Annual ACM Symposium on the Theory of Complexity, 1975, 23-26.