DIVIDE AND CONQUER FOR LINEAR EXPECTED TIME *

Jon Louis BENTLEY and Michael Ian SHAMOS
Departments of Computer Science and Mathematics, Carnegie-Mellon University, Pittsburgh, PA 15213, U.S.A.

Received 29 March 1977; revised version received 17 October 1977

Average-case analysis, computational geometry, convex hull, divide-and-conquer, expected time, linear programming, random sets

1. Introduction

Divide-and-conquer is one of the most frequently used methods for the design of fast algorithms. The most common application of the technique involves breaking a problem of size N into two subproblems of size $N/2$, solving these subproblems, then doing work proportional to N to "marry" the partial answers into a solution for the entire problem; this scheme leads to algorithms of $O(N \log N)$ worst-case time complexity. In this paper we investigate a similar divide-and-conquer technique which can be used to construct algorithms with linear average-case time complexity.

The problem of determining the convex hull of a set of points in two and three dimensions has produced a rash of recent papers [4,8,15,16], all containing algorithms with $O(N \log N)$ worst-case performance. That this is optimal follows from the fact that in the worst case all N points may be vertices of the convex hull, and since the vertices of a convex polygon occur in sorted angular order about each interior point, any convex hull algorithm must be able to sort [14,8]. If the boundary of the convex hull contains very few points, however, this lower bound does not apply, and a faster algorithm may be possible. The algorithm of Jarvis [5] runs in time $O(hN)$, where h is the number of actual hull vertices, and thus takes advantage of the fact that h may be small. Unfortunately, if h is not known in advance, the algorithm may take quadratic time. Eddy [2] has developed a hull algorithm analogous to QUICKSORT that has good empirical performance but also has a quadratic worst case. [1] In this paper we use information about the probability distribution of h to obtain an algorithm with $O(N)$ expected running time without sacrificing $O(N \log N)$ worst-case behavior.

This new convex hull algorithm leads to linear expected-time solutions to a host of other geometry problems that are related to hull-finding. Among these are determining the greatest distance between two points of a set, the smallest circle enclosing a set, and constructing linear pattern classifiers. Analogous techniques yield a linear average-case algorithm for linear programming in two variables.

The divide-and-conquer scheme we use to achieve the above results seems to be a general method suitable for the construction of fast average-case algorithms. It achieves fast expected time at the cost of making only relatively weak assumptions about the underlying probability distribution of the inputs. Whereas many fast average-case algorithms display poor worst-case behavior (QUICKSORT, for example; see [13]), those that we give in this paper have optimal worst-case performance. These algorithms are not merely of asymptotic interest – they are faster than previous methods even for very small problem sizes ($N > 40$, for example).

In reading this paper, one must be very careful to keep in mind the distinction between worst-case and

* This research was supported in part by the Office of Naval Research under Contract N00014-76-C-0829.

[1] R.W. Floyd is able to show that Eddy's algorithm runs in linear expected time for certain symmetric distributions (personal communication).

average-case analyses. For example, while any convex hull algorithm must run in time $\Omega(N \log N)$ for some inputs [2], we will give an algorithm with linear expected running time (for some distribution of inputs). Notice that there is no contradiction between a worst-case lower bound of $\Omega(N \log N)$ and an average-case upper bound of $O(N)$.

Basic results from stochastic geometry are described in Section 2; these results form the basis of our probabilistic analysis of the algorithms presented. In Section 3 we give a fast expected-time algorithm for finding convex hulls in the plane and investigate in detail the schema used in the algorithm. Section 4 shows how this method can be applied to other problems and used as a building block for developing additional fast expected-time algorithms. Section 5 contains suggestions for further work along these lines.

2. Results from stochastic geometry

Stochastic geometry deals with the properties of random sets of points, lines and other geometric objects and is an essential tool for analyzing the average case of geometric algorithms. Many phenomena in geometrical probability are counter-intuitive and difficult to explain without the tools of probabilistic measure theory. For example, the statement, "Choose N points at random in the plane", is meaningless without a precise specification of distribution from which the points are to be chosen. Furthermore, not all conceivable distributions satisfy the axioms of probability. Points can be chosen uniformly in the plane only from a set of bounded Lebesgue measure [6], so the intuitively attractive notion of a uniform random selection from the whole plane must be discarded.

The problem of determining $h(N)$, the expected number of vertices of the convex hull of N points, has received a good deal of attention [1,3,9,11]; a summary of this work may be found in [12]. We now quote several results that will be used later in analyzing our algorithms:

[2] We say that $f(N) = \Omega(g(N))$ if $f(N)/g(N)$ is bounded below by some nonzero constant C as $N \to \infty$.

Theorem 1 (Rényi and Sulanke [11]). *If N points are chosen uniformly and independently at random in the plane from a convex r-gon, then as $N \to \infty$,*

$$h(N) = (2r/3)(\gamma + \log_e N) + O(1)\,.$$

(γ = Euler's constant)

Theorem 2 (Raynaud [9]). *If N points are chosen uniformly and independently at random from the interior of a k-dimensional hypersphere, then as $N \to \infty$, $f(N)$, the expected number of faces of the convex hull, is given asymptotically by*

$$f(N) = O(N^{(k-1)/(k+1)})\,.$$

Since with probability one each face of the hull is simplicial and thus is determined by k vertices, Theorem 2 implies that
$h(N) = O(N^{1/3})$, for points chosen uniformly in a circle, and
$h(N) = O(N^{1/2})$, for points chosen uniformly in a sphere.
Thus in any dimension, for points in a hypersphere, the expected number of hull vertices is bounded above by N^p, for some $p < 1$.

Theorem 3 (Raynaud [9]). *If N points are chosen independently from a normal distribution in k dimensions, then as $N \to \infty$ the asymptotic behavior of $h(N)$ is given by*

$$h(N) = O((\log N)^{(k-1)/2})\,.$$

A useful connection can be established between the stochastic properties of convex hulls and the expected number of maximal vectors in a random set. A *maximal* vector is one that is not less than any other in all components. Under very general conditions the expected number of maximal vectors in a set is quite small:

Theorem 4 (Kung, Schkolnick, Thompson [7]). *If N k-dimensional vectors are chosen such that their components are distributed independently, then $A(N, k)$, the expected number of maximal vectors, is bounded by*

$$A(N, k) < (\log_e N)^{k-1} \qquad \text{for } N > 3.$$

Note that a vertex of the convex hull of a finite k-dimensional set is maximal for some assignment of plus and minus signs to all coordinates of its points. This implies that for distributions satisfying the independence assumption of Theorem 4, the expected number of vertices of the convex hull is bounded by

$$E(h) \leq 2^k(\log_e N)^{k-1} = 2(2\log_e N)^{k-1}\,.$$

The multivariate normal of Theorem 3, the multivariate exponential, and the uniform distribution over a hypercube all satisfy the independence-of-components assumption. The qualitative behavior of the hulls of random sets may be understood intuitively as follows: for uniform sampling within any bounded figure F, the hull of a random set will tend to assume the shape of the boundary of F. If F is a polygon, points accumulating in the "corners" will cause the resulting hull to have very few vertices. Because the circle has no corners, the expected number of hull vertices is comparatively high. It is reasonable that only some small fraction of the sample points should survive as hull vertices, but in all of the above theorems the *order* of $h(N)$ is sublinear. Informally we may account for this by noting that the hull is a manifold of strictly lower dimension than the set from which the points are being chosen. If this is not true, we may have $h(N) = O(N)$. For example, if N points are selected uniformly on the boundary of a circle, then $h(N) = N$. As we shall see in the next section, the only assumption about the distribution of points that needs to be made in order to obtain a linear expected-time algorithm is that $h(N) = O(N^p)$, for some $p < 1$.

3. Convex hulls in the plane

The fast convex hull algorithm is easily described as a recursive procedure: If N, the number of given points, is less than some constant C, then the procedure calculates the hull by some straightforward method and returns. If N is large, though, the procedure first divides the N points into two subsets of approximately $N/2$ points each by a method which ensures that the resulting subproblems are random. It then finds the convex hulls of the random subproblems recursively, which will take expected time $2T(N/2)$, since the subproblems are of the same form as the original. The result of each of the recursive calls is a convex polygon whose expected number of vertices is $O(N^p)$, with $p < 1$. The hull of the given set is now just the hull of the union of the hulls found in the subproblems. Shamos [15] has given an algorithm to find the hull of the union of two convex polygons in time proportional to the total number of vertices of both. We may use this algorithm to merge the results of the subproblems in expected time $O(N^p)$. The average running time of this algorithm thus obeys the recurrence

$$T(N) = 2T(N/2) + O(N^p)\,, \tag{1}$$

whose solution, for $p < 1$, is $T(N) = O(N)$. Thus we have shown that the algorithm runs in linear expected time for point sets satisfying the assumptions made in Section 2.

We assumed above two important properties about the division step of the divide-and-conquer algorithm: that it can be accomplished in constant time and that the points in the subproblems obey the same probability distribution as do the original points. A division step with these properties can easily be implemented on a RAM by storing the points in a two by N array of Cartesian coordinates. Each point is initially assigned a random location in the array and a subset of the points is represented as a pair of integers which define the left and right endpoints of a segment of the array. Division into further subsets can be accomplished by taking the arithmetic mean of the endpoints as defining two new segments, etc.; note that the division preserves randomness. In implementing this algorithm recursively, it is crucial to pass only pointers in the subroutine calls. If entire subproblems are passed, equation (1) no longer applies and an $N \log N$ algorithm will result, no matter how few points are on the convex hull.

Let us now note the features of the above algorithm that give it linear expected time. First, the expected size of the output is small. Second, solutions to the random subproblems can be married quickly to form a solution to the total problem. Note that the algorithm also has optimal worst-case performance. Since the largest hull that can be returned by a subproblem is of size N, we always have

$$T(N) \leq 2T(N/2) + O(N)\,, \tag{2}$$

whose solution is $T(N) \leq O(N \log N)$. We can use this algorithm as a paradigm by which to create others with linear expected time and optimal worst-case behavior.

4. Further examples

The first simple extension of the algorithm of Section 3 gives a linear expected-time algorithm for the convex hull of a set of points in three dimensions. Preparata and Hong, in an important recent paper [8], have shown that the hull of the union of two disjoint convex three-dimensional polyhedra can be found in time that in the worst case is only linear in the total number of vertices. Their algorithm makes no essential use of the fact that the polyhedra are disjoint and can be readily modified to include the case in which the intersection is nonempty. If the points are drawn from a distribution satisfying the assumptions of Section 2, then the recurrence relation (1) applies and we again have a linear expected-time algorithm.

Many geometric algorithms are based on finding convex hulls. For example, the diameter of a set (distance between its two farthest points) is always realized by two vertices of the hull. Furthermore, these points can be found in linear time (in two dimensions) once the convex hull is available [14]. We thus immediately have a linear expected-time diameter algorithm. Somewhat more complicated is the problem of determining the smallest circle enclosing a plane set of points. This is a classical problem with an extensive literature. An $O(N \log N)$ worst-case algorithm is given in [16]. It is elementary to show that the two or three points determining this circle are vertices of the convex hull. If we first find the hull with a linear expected-time algorithm, the time required for the remaining step (finding the circle) is not linear in the number of hull vertices. If $E(h)$ is the expected number of hull vertices, we need to know $E(h \log h)$ to complete the analysis. Note that we always have $1 \leqslant h \leqslant N$ and

$$E(h) = \sum_{i=1}^{N} i p_i ,$$

where p_i is the probability that $h = i$. Now, since $\log i \leqslant \log N$,

$$E(h \log h) = \sum_{i=1}^{N} (i \log i)\, p_i$$

$$\leqslant (\log N)\, E(h) .$$

Thus, if $E(h) = O(N^p)$, $p < 1$, then $E(h \log h) = O(N^q)$, for some $q < 1$. We may therefore find the smallest circle enclosing a plane set in linear average time.

In general, determining expectation values of functions of h is a difficult problem and we often must be satisfied with upper bounds. The largest area triangle determined by three points of a set of N points in the plane can be found in time that is quadratic in the number of vertices of the hull [15]. In order to be able to calculate the average-case behavior of the algorithm, we must compute $E(h^2)$. If $E(h) = O(N^p)$, then certainly $E(h^2) \leqslant O(N^{p+1})$. Applying Theorem 2, we may find the largest determined triangle in $O(N^{4/3})$ expected time in two dimensions, and this bound is highly pessimistic.

Theorem 4 leads directly to a linear expected-time algorithm for finding the maxima of N k-dimensional vectors whose coordinates are chosen independently. It is only necessary to remark that the marriage step of the divide-and-conquer algorithm finds the common maxima of two subproblems of size $N/2$, each of which has very few maxima, on the average.

We often observe that the performance of an algorithm is much better than its worst-case lower bounds would lead us to expect; the Simplex algorithm for linear programming is a striking example of this phenomenon. Fast as Simplex is, however, it is known not to be optimal for problems with small numbers of variables, and a divide-and-conquer approach can be used to advantage [17]: the feasible region of a two-variable problem is the intersection of the half-planes determined by the linear constraints. If we denote the ith half-plane by H_i, then we want to form

$$H_1 \cap H_2 \cap \ldots \cap H_N .$$

Since the intersection operator is associative, this may be rearranged as

$$(H_1 \cap \ldots \cap H_{N/2}) \cap (H_{N/2+1} \cap \ldots \cap H_N) .$$

Each term is an intersection of $N/2$ half-planes, and is thus a convex polygonal region of at most $N/2$ vertices. The intersection of two such figures can be found in linear time at worst [14], so equation (2) describes the worst-case behavior of the algorithm.

We may thus find the intersection of N half-planes in $O(N \log N)$ time. If many of the half-planes are redundant, though, the final intersection will have very few vertices, and we may take advantage of this fact to develop a better algorithm. Suppose that K_0 is a bounded convex region of the plane that contains another convex region K_1. If N lines L_i are drawn independently and at random to meet K_0 but not K_1, and we define H_i to be the closed half-plane bounded by L_i that contains K_1, consider $E(v)$, the expected number of vertices of the intersection of all the H_i. Preliminary results were obtained by Rényi and Sulanke [10] and Ziezold [18] has shown by duality that $E(v)$ is of the same asymptotic order as the expected number of points on the hull of a set of N *points* drawn uniformly within K_1. If K_1 shrinks to a point, then $E(v)$ approaches the constant $\pi^2/2$. In any event, under fairly conservative assumptions we will have $E(v) = O(N^p)$, $p < 1$, and a linear average-case algorithm for intersecting N half-planes results. This leads immediately to an $O(N)$ expected-time algorithm for linear programming in two variables and for finding the kernel of a polygon [14].

5. Suggestions for further work

It is natural to try to extend the results of this paper to higher-dimensional problems in geometry and to other problem domains. The limiting factor, however, is not the technique but our inadequate knowledge of the properties of random sets and our inability to develop efficient merge procedures to make divide-and-conquer productive. As an example, no method is now known to find the hull of the union of four-dimensional polyhedra in less than quadratic expected time. (Quadratic time *is* required in the worst case. See [8].) Likewise, the expected value of the square of the number of vertices of the hull of N points chosen uniformly in a four-dimensional hypersphere in not known to be less than $O(N^{8/5})$. We are thus unable to give an algorithm for the four-dimensional convex hull whose expected running time is provably less than $O(N^{8/5})$.

References

[1] H. Carnal, Die konvexe Hülle von n rotationssymmetrische verteilten Punkten, Z. Wahrscheinlichkeits. 15 (1970) 168–176.
[2] W. Eddy, A new convex hull algorithm for planar sets, ACM Transactions on Mathematical Software, 1977, to appear.
[3] B. Efron, The convex hull of a random set of points, Biometrika 52 (1965) 331–344.
[4] R.L. Graham, An efficient algorithm for determining the convex hull of a planar set, Information Processing Lett. 1 (1972) 132–133.
[5] R.A. Jarvis, On the identification of the convex hull of a finite set of points in the plane, Information Processing Lett. 2 (1973) 18–21.
[6] M.G. Kendall and P.A.P. Moran, Geometrical Probability, Griffin (1963) 125 pp.
[7] H.T. Kung, M. Schkolnick and C.D. Thompson, On the average number of maxima in a set of vectors, submitted for publication.
[8] F.P. Preparata and S.J. Hong, Convex hulls of finite sets of points in two and three dimensions, CACM 20 (2) (1977) 87–93.
[9] H. Raynaud, Sur l'enveloppe convexe des nuages des points aléatoires dans R_n, I, J. Appl. Prob. 7 (1970 35–48.
[10] A. Rényi and R. Sulanke, Über die konvexe Hülle von n zufällig gewählten Punkten, I, Z. Wahrscheinlichkeits. 2 (1963) 75–84; II, 3 (1964) 138–147.
[11] A. Rényi and R. Sulanke, Zufällige konvexe Polygone in einem Ringgebeit, Z. Wahrscheinlichkeits. 9 (1968 146–157.
[12] L.A. Santaló, Integral Geometry and Geometric Probability, Encyclopedia of Mathematics and Its Applications, v.1. (Addison-Wesley, Reading, MA, 1976) 404 pp.
[13] R. Sedgewick, Quicksort. Stanford University Dept. of Comp. Sci. Tech. Rpt. STAN-CS-75-492, May, 1975.
[14] M.I. Shamos, Geometric complexity. Proc. Seventh Annual ACM Symposium on Automata and Computability Theory (1975) 224–233.
[15] M.I. Shamos, Problems in Computational Geometry. Unpublished notes. To appear as Computational Geometry (Springer-Verlag, Berlin, 1977).
[16] M.I. Shamos and D. Hoey, Closest-point problems. Proc. Sixteenth Annual IEEE Symposium on Foundations of Computing (October, 1975) 151–162.
[17] M.I. Shamos and D. Hoey, Geometric Intersection Problems. Proc. Seventeenth Annual IEEE Symposium on Foundations of Computer Science (October, 1976) 208–215.
[18] H. Ziezold, Über die Eckenzahl zufälliger konvexer Polygone, Izv. Akad. Nauk. Armjan. SSR. Ser. Mat. 5 (1970) 296–312.