# ON THE AVERAGE NUMBER OF MAXIMA IN A SET OF VECTORS AND APPLICATIONS

J. L. Bentley
H. T. Kung
M. Schkolnick*
C. D. Thompson

July 1977

DEPARTMENT
of
COMPUTER SCIENCE

Carnegie-Mellon University

# ON THE AVERAGE NUMBER OF MAXIMA IN A SET OF VECTORS AND APPLICATIONS

J. L. Bentley
H. T. Kung
M. Schkolnick*
C. D. Thompson

July 1977

Department of Computer Science
Carnegie-Mellon University
Pittsburgh, PA 15213

*IBM Research Laboratory, San Jose, CA 95193

This research was supported in part by the National Science Foundation under Grant MCS 75-222-55 and the Office of Naval Research under Contract N00014-76-C-0370, NR 044-422.

ABSTRACT

A maximal vector of a set is one which is not less than any other vector in all components. We derive a recurrence relation for computing the average number of maximal vectors in a set of n vectors in d-space under the assumption that all $(n!)^d$ relative orderings are equally probable. Solving the recurrence shows that the average number of maxima is $O((\ln n)^{d-1})$. We use this result to construct an algorithm for finding all the maxima that has expected running time linear in n (for sets of vectors drawn under our assumptions). For a given set of random points, the result is also used to derive an upper bound on the expected number of points from the set which are on the boundary of the convex hull of the set.

## 1. INTRODUCTION

The problem of finding all maximal vectors in a set of n d-vectors has recently been studied by Kung, Luccio and Preparata [3] and F. Yao [7]. In this paper we consider the related problem of finding the expected number of maximal elements in a given set. We give a solution to that problem under a very general probability distribution and then apply the answer to the solution of related problems.

A maximal vector is one which is not less than any other vector in all components. More precisely, we say that a vector P dominates the vector Q if P is greater than Q in every component; then a vector is maximal if it is not dominated by any other vector in the set. For example, in $\{(1,2,4), (2,3,1), (3,1,3), (4,4,2)\}$, only (2,3,1) is not maximal. It is helpful to view this problem geometrically when $d = 2$. In that case the vectors can be considered as n points in the plane and a given vector is maximal if and only if there is no point in its first quadrant (above it and to its right).

A probability distribution is implied as we ask for the expected number of maxima. A mathematically tractable yet reasonabel model assumes that for each vector, the magnitude of one component is distributed independently of the magnitude of the other components and, for each component, the magnitudes chosen for each vector are distinct. The second restriction implies that the vectors can be sorted into increasing order on any component, yielding a relative ordering from 1 to n. Thus each set of n d-vectors corresponds to a particular relative ordering for each component, that is, to one of $(n!)^d$ assignments of permutations

of (1,2,3,...,n) to the d components. Examples of multivariate statistical distributions with distinct components distributed independently include the multivariate normal and multivariate uniform drawn from a unit hypercube. (Recall that elements drawn independently from any continuous distribution function are distinct with probability one.)

The solution to the maximal vector problem is often required in the analysis of the runtime of dynamic programming algorithms (see Schkolnick [5] and Schkolnick and Thompson [6]). In dynamic programming the solution to a problem of size n is obtained from the best solutions of problems of size n-1. For many applications a cost vector of length one is sufficient, i.e., there is a single best solution to all subproblems. In cases where more than one best solution must be retained for each subproblem, it may still be possible to design a multidimensional cost function with the property that the best solutions for every subproblem are just the maximal ones. If the cost vectors of candidate solutions are assumed to have the proper distribution, then the maximal vector problem indicates the expected number of best solutions.

In Section 2 we formulate and solve a recurrence that shows that the expected number of maxima among n d-vectors is $O((\ln n)^{d-1})$. We use this result in Section 3 to give an algorithm for finding all the maxima of a set of n d-vectors that has expected running time linear in n. In Section 4 we show for a given set of random points, how this result gives an upper bound on the expected number of points from the set which are on the boundary of the convex hull of the set.

## 2. DETERMINING THE AVERAGE NUMBER OF MAXIMA

In this section we derive the primary result of this paper. We give two derivations of this result. Our first derivation is formal and therefore rather complicated, so we supplement that with a second, informal derivation. The second derivation is not completely precise, but it does give an intuitive idea of the essential workings of the formal derivation. We proceed directly with the formal derivation; the informal begins immediately after the statement of Theorem 2.

Let $A(n,d)$ be the average number of maximal vectors out of n d-vectors. Without loss of generality, assume that the vector components in each dimension are integers from 1 to n. We shall therefore view a set of n d-vectors as a d by n array, whose rows are the vectors and whose columns are permutations of $\{1,2,\ldots,n\}$. Let S be the set of all such arrays. Then S contains $(n!)^d$ arrays. For any array r in S, let $M(r)$ denote the number of maximal vectors in r. By the definition of $A(n,d)$, we have

$$A(n,d) = \frac{\sum_{r \in S} M(r)}{(n!)^d} .$$

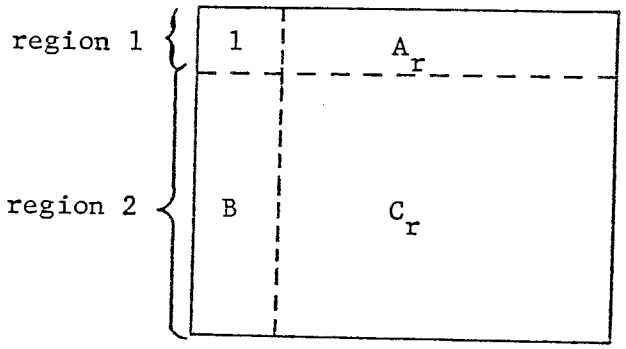
Let T be a subset of S which consists of arrays with their first columns equal to $(1,2,\ldots,n)^T$. Because $M(r)$ is invariant under permutations of the rows in r, it follows that

$$\sum_{r \in T} M(r) = \frac{1}{n!} \sum_{r \in S} M(r).$$

Thus,

$$(2.1) \quad (n!)^{d-1} A(n,d) = \sum_{r \in T} M(r).$$

Any array r in T can be decomposed as




where $B = (2,3,\ldots,n)^T$. (Region 1 contains only one vector, namely, $(1,A_r)$.) For $i = 1,2$, define $M_i(r)$ to be the number of maximal vectors in r which are in region i. Thus,

$$(2.2) \quad M(r) = M_1(r) + M_2(r).$$

(Note that $M_1(r)$ is either zero or one.) Taking sums in both sides, we have by (2.1),

$$(2.3) \quad (n!)^{d-1}A(n,d) = \sum_{r\in T} M_1(r) + \sum_{r\in T} M_2(r).$$

Lemma 1

$$\sum_{r\in T} M_1(r) = \frac{1}{n}\,(n!)^{d-1}A(n,d-1).$$

Proof

Note that $\sum_{r\in T} M_1(r)$ is the number of times, over all r in T, vector $(1,A_r)$ is maximal in r. Note also that vector $(1,A_r)$ is maximal over all n d-vectors in r if and only if vector $A_r$ is maximal over all n (d-1)-vectors in the sub-

array $\begin{bmatrix} A_r \\ C_r \end{bmatrix}$. Then $\sum_{r\in T} M_1(r)$ is simply the number of times, over all r in T, the vector $A_r$ in the firs row is maximal in $\begin{bmatrix} A_r \\ C_r \end{bmatrix}$. Since over all r in T there are $(n!)^{d-1}A(n,d-1)$ maximal (d-1)-vectors and the number of maximal vectors occurring in each row is the same, it follows that

$$\sum_{r\in T} M_1(r) = \frac{1}{n}(n!)^{d-1}A(n,d-1). \qquad \blacksquare$$

Lemma 2

$$\sum_{r\in T} M_2(r) = (n!)^{d-1}A(n-1,d).$$

Proof

Consider an array r in T. Note that a d-vector in region 2 is maximal over all n d-vectors in r if and only if it is maximal over all n-1 d-vectors in region 2. Therefore, for any r in T we shall consider only region 2. $\sum_{r\in T} M_2(r)$ is the number of occurrences of maximal vectors in region 2 over all r in T. Let A be a fixed (d-1)-vector. Over all r in T where r has (1,A) as its first row, there are $((n-1)!)^{d-1}A(n-1,d)$ maximal d-vectors occurring in region 2. Since A may be chosen in $n^{d-1}$ ways, we have

$$\sum_{r\in T} M_2(r) = (n!)^{d-1}A(n-1,d). \qquad \blacksquare$$

The following theorem follows from (2.3) and Lemmas 1 and 2.

Theorem 1

$$(2.4)\quad A(n,d) = A(n-1,d) + \frac{A(n,d-1)}{n},$$

for $n, d \geq 2$.

It is easy to check that

(2.5) $A(1,d) = 1$ for $d \geq 1$,

and

(2.6) $A(n,1) = 1$ for $n \geq 1$.

The recurrence (2.4) with initial conditions (2.5) and (2.6) can be solved by first setting up the generating functions

$$G_n(z) = \sum_{d\geq 0} A(n,d+1)z^d, \quad \text{for } n \geq 1.$$

By (2.4) and (2.6),

$$G_n(z) = G_{n-1}(z) + \frac{z}{n} G_n(z), \text{ or}$$

$$(2.7) \quad G_n(z) = \frac{G_{n-1}(z)}{1 - \frac{z}{n}}$$

for $n \geq 2$. By (2.5), $G_1(z) = 1/(1-z)$. Hence (2.7) implies that

$$G_n(z) = \prod_{1\leq i\leq n} \frac{1}{1 - \frac{z}{i}},$$

which is Eq. (33) of Section 1.2.9 in Knuth [1] with $x_i = 1/i$. Define

$$H^{(r)}(n) = 1 + \frac{1}{2^r} + \frac{1}{3^r} + \ldots + \frac{1}{n^r}.$$

Knuth's analysis shows that the coefficient of $z^d$ in $G_n(z)$ is

$$\sum_{\substack{k_1,k_2,\ldots,k_d\geq 0 \\ k_1+2k_2+\ldots+dk_d=d}} \frac{H^{(1)}(n)^{k_1}}{1^{k_1}k_1!} \frac{H^{(2)}(n)^{k_2}}{2^{k_2}k_2!} \cdots \frac{H^{(d)}(n)^{k_d}}{d^{k_d}k_d!}$$

which is, by definition, $A(n,d+1)$. Clearly,

$$(2.8)\quad A(n,d) \geq \frac{1}{(d-1)!} H^{(1)}(n)^{d-1}.$$

The values of $A(n,d)$ for small d are

$$A(n,1) = 1,$$
$$A(n,2) = H^{(1)}(n), \text{ which is the nth harmonic number } H_n,$$
$$A(n,3) = \frac{1}{2} H^{(1)}(n)^2 + \frac{1}{2} H^{(2)}(n),$$
$$A(n,4) = \frac{1}{6} H^{(1)}(n)^3 + \frac{1}{2} H^{(1)}(n)\, H^{(2)}(n) + \frac{1}{3} H^{(3)}(n).$$

It is not difficult to show that the sum of the coefficients in $A(n,d)$ is always 1. (For example, the sum of the coefficients in $A(n,4)$ is $\frac{1}{6} + \frac{1}{2} + \frac{1}{3} = 1$.) Since $H^{(r)}(n) \leq H^{(1)}(n)^r$ for $n, r \geq 1$, we have

$$(2.9)\quad A(n,d) \leq H^{(1)}(n)^{d-1}.$$

By (2.8) and (2.9) we have

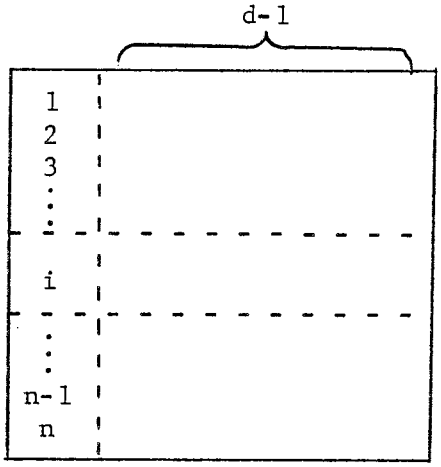
Theorem 2

$$\frac{1}{(d-1)!} H^{(1)}(n)^{d-1} \leq A(n,d) \leq H^{(1)}(n)^{d-1}.$$

Therefore, $A(n,d) = O((\ln n)^{d-1})$ for fixed d.

We now give a more intuitive derivation of the recurrence for $A(n,d)$. As we stated previously, this derivation is not precise, but it should help in getting an intuitive idea of the workings of the previous proof. To compute the expected number of maxima in a set we

will consider the set sorted in order by the first coordinate. (As before, we consider that all numbers have been translated to the integers from 1 to n.) The situation we now have is illustrated in the following figure.




We now ask what is the probability that the i-th vector in the set is a maximum? Since its first coordinate is greater than the first coordinates of the 1-st through the (i-1)-st vectors, it cannot be dominated by any of those. Therefore the i-th vector is a maxima if and only if its remaining d-1 coordinates are maximal in the set of the i-th through the n-th vectors. The probability that the i-th vector is a maximal in this set is, by independence, the expected number of maxima in the set (which is A(n-i+1,d-1)) divided by the total number of vectors in the set (which is n-i+1). Since these probabilities are indepndent for all values of i, to find the expected number of maxima in the set we sum the probabilities of each vector being maximal and we have

$$A(n,d) = \sum_{i=1}^{n} \frac{A(n-i+1,d-1)}{n-i+1}$$

$$= \sum_{j=1}^{n} \frac{A(j,d-1)}{j} \quad .$$

Notice that the last sum is equivalent to the expression for A(n,d) in Theorem 1.

We now give a simpler (and less precise) bound on the growth of A(n,d). It is obvious that A(n,d) must be monotone increasing in n, so if $j \leq n$ then $A(j,d) \leq A(n,d)$. We use this observation in the following derivation.

$$\begin{aligned} A(n,d) &= \sum_{j=1}^{n} \frac{A(j,d-1)}{j} \\ &= \sum_{j=1}^{n} \frac{A(n,d-1)}{j} \\ &= A(n,d-1) \sum_{j=1}^{n} \frac{1}{j} \\ &= A(n,d-1)\ H^{(1)}(n). \end{aligned}$$

Iterating this recurrence on d easily gives the upper bound

$$A(n,d) \leq H^{(1)}(n)^{d-1}.$$

## 3. A FAST EXPECTED TIME MAXIMA ALGORITHM

So far in this paper we have considered the problem of counting the number of maxima in a set of vectors; a related problem is finding the maxima in a set of vectors. This problem has received much attention recently. Kung, Luccio and Preparata [3] give an algorithm for finding the maxima of n vectors in d-space that has worst-case running time of $O(n \ln n)$ for $d = 2$ and $O(n(\ln n)^{d-2})$ for $d \geq 3$. F. Yao [7] shows that the results in 2 and 3-space are optimal by giving a worst-case lower bound of $O(n \ln n)$ (indeed, she gives a bound that is the exact number of comparisons taken by a known algorithm for planar sets).

These results, however, deal only with the worst-case complexity of finding maxima; it is often interesting to consider also the average-case complexity. In this section we will use Theorem 2 and a general divide and conquer schema to give a fast expected time algorithm for finding maxima (this schema is investigated in detail by Bentley and Shamos [1]). The algorithm we develop here will have expected running time linear in n for vector sets drawn under the "Independent and Distinct" assumptions stated in Section 1.

Our maxima algorithm is easily described recursively. Without loss of generality, we assume that n is a power of two. To find the maxima of a set S of n vectors, divide S into two sets A and B, each containing n/2 vectors. Recursively find the maxima of A and B, calling those sets $M_A$ and $M_B$, respectively. It is easy to see that the set of maximum vectors of S is the set of maxima of $M_A \cup M_B$. Therefore we can find all the maxima of S by finding the maxima of $M_A \cup M_B$; to do this we use the algorithm of Kung, Luccio and Preparata [3]. (Recursion in our original algorithm stops when n is less than

some predefined constant.) The division into subproblems can be implemented on a random access computer by storing the vectors in a d by n array of scalar values. Each vector is initially represented as a pair of integers which define the left and right endpoints of a segment in 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 and can be accomplished in constant time.

The expected running time of this algorithm is easy to analyze, given that the expected number of maxima in a set of n d-vectors is $O((\ln n)^{d-1})$. Since division into subproblems can be accomplished in constant time, the recurrence describing the expected running time of our algorithm on n d-vectors is

$$(3.1) \quad T(n,d) = 2T(n/2,d) + F(n,d)$$

where $F(n,d)$ is the expected running time of the marriage step (finding the maxima of $M_A \cup M_B$). Let i be the number of vectors in $M_A \cup M_B$. Then the running time of the marriage step using the algorithm of Kung, Luccio and Preparats is bounded above by $O(i(\ln i)^{d-2})$ for $d \geq 3$. This gives

$$F(n,d) \leq \sum_{i=1}^{n} p(i) \cdot i(\ln i)^{d-2}$$

where $p(i)$ is the probability of there being exactly i maxima in $M_A \cup M_B$. By the fact that the number of maxima in A is independent of that in B, the expected value of i satisfies:

$$E(i) = \sum_{i=1}^{n} p(i) \cdot i$$

$$= 2 \cdot (\text{expected number of maxima in a set of } n/2 \text{ d-vectors})$$

$$= 2 \cdot O((\ln \tfrac{n}{2})^{d-1})$$

$$= O((\ln n)^{d-1}).$$

Therefore, by $\ln i \leq \ln n$, we have

$$F(n,d) \leq (\ln n)^{d-2} \sum_{i=1}^{n} p(i) \cdot i \tag{3.2}$$

$$= O((\ln n)^{2d-3}).$$

Substituting (3.2) into (3.1) gives for the running time of our algorithm the recurrence

$$T(n,d) \leq 2T(n/2,d) + O((\ln n)^{2d-3}).$$

For fixed d, this recurrence is well known to have the solution

$$T(n,d) = O(n).$$

In addition to having a very fast expected running time, our algorithm also has quite a respectable worst-case performance. Note that $F(n,d)$ is always bounded above by $O(n(\ln n)^{d-2})$ for $d \geq 3$, so the worst-case running time of our algorithm is given by

$$T(n,d) = 2T(n/2,d) + O(n(\ln n)^{d-2})$$

which has the solution

$$T(n,d) = O(n\ (\ln n)^{d-1}).$$

Thus in the worst-case our algorithm is only a factor of ln n slower than the best known worst-case algorithm.

We summarize the main result of this section in the following theorem.

Theorem 3.

The maxima of a set of n d-vectors drawn from a distribution satisfying the "Independent and Distinct" property can be found in expected time linear in n.

## 4. RELATION TO CONVEX HULLS

The maximal elements of a set of vectors are a crude representation of the boundary of the set; the boundary can be more precisely defined as the boundary of the convex hull of the set. While working with the convex hull we will view the vectors as points in d-space. The convex hull of the n points is then defined as the smallest convex set containing the n points. One can get an intuitive picture of the convex hull of a planar point set by imagining the n points as n nails in a large board, with about an inch of each nail remaining above the board. The convex hull of this set can be found by taking a large rubber band, stretching it infinitely far out in all directions, then letting it go. It will come to rest about certain of the nails, and the region within the rubber band is the convex hull of the set.

Given a set of n points sampled independently from some underlying probability distribution function in d-space, what is the expected number of points on the resulting convex hull? (Here we use the abbreviation "on the convex hull" to mean "on the boundary of the convex hull".) The answer to this question is of course dependent of n, d, and the underlying distribution. Santalo [4] describes a number of results for different distributions; many of these results and their original references are given in Bentley and Shamos [1]. In this section we will give an upper bound on the number of hull points for distributions satisfying our requirement of independence among the d variables. To arrive at this bound we will first show that every convex hull point is a maximum under at least one of the $2^d$ possible different assignments of + and - signs to the d components, and then use this fact and Theorem 2 to bound the expected number of hull points.

To show that every convex hull point is a maximum under at least one of the assignments of + and - signs, assume that there is some hull point h which is not. This implies that there is at least one point in each of h's $2^d$ orthants; choose one point from each orthant and call this collection P. Because values are distinct, all points in P are properly contained in their orthants. Consider now the convex hull of P; it must properly contain h. (If it contained all the points of P and not h, then it would not be a convex set.) Since h is properly contained in the convex hull of P it must also be properly contained in the convex hull of the original set. This contradicts our assumption and establishes the desired fact.

We have shown that every hull point is a maximum under at least one of the $2^d$ possible assignments of + and - signs to the d variables. Consider now the set of all points that are maximal under at least one of the sign assignments; call this set M. Since the expected size of M is bounded above by $2^d \cdot O((\ln n)^{d-1})$, and M contains all convex hull points, the expected number of convex hull points is certainly bounded above by that expression. Thus we have the following theorem.

Theorem 4.

The expected number of convex hull points in a point set of n points in d dimensions satisfying the "Independent and Distinct" property is bounded above by $O((\ln n)^{d-1})$.

ACKNOWLEDGMENTS

The authors gratefully acknowledge helpful discussions with Professor Michael Shamos.

BIBLIOGRAPHY

[1] Bentley, J. L. and M. I. Shamos, Divide and Conquer for Linear Expected Time, Carnegie-Mellon University Computer Science Department Report, 1977, 8 pp.

[2] Knuth, D. E., The Art of Computer Programming, Vol. 1: Fundamental Algorithms, Addison-Wesley, Reading, MA, 1973.

[3] Kung, H. T., F. Luccio and F. P. Preparata, "On Finding the Maxima of a Set of Vectors," Journal of the ACM, vol. 22, no. 4, October 1975, pp. 469-476.

[4] Santalo, L. A., "Integral Geometry and Geometric Probability," Encyclopedia of Mathematics and Its Applications, Vol. 1, Addison-Wesley, 1976, 404 pp.

[5] Schkolnick, M., "A Clustering Algorithm for Hierarchical Structures," ACM Transactions on Database Systems, vol. 2, no. 1, March 1977, pp. 27-44.

[6] Schkolnick, M. and C. D. Thompson, "A Clustering Strategy for Relational Data Bases," 1977, to appear.

[7] Yao, F. F., On Finding the Maximal Elements in a Set of Plane Vectors, University of Illinois Computer Science Department Report UIUCDCS-R-74-667, July 1974, 6 pp.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

| REPORT DOCUMENTATION PAGE | | READ INSTRUCTIONS BEFORE COMPLETING FORM |
|---|---|---|
| 1. REPORT NUMBER | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE *(and Subtitle)*<br>ON THE AVERAGE NUMBER OF MAXIMA IN A SET OF VECTORS AND APPLICATIONS | | 5. TYPE OF REPORT & PERIOD COVERED<br>Interim<br>6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>J. L. Bentley M. Schkolnick<br>H. T. Kung C. D. Thompson | | 8. CONTRACT OR GRANT NUMBER(s)<br>N00014-76-C-0370,<br>NR 044-422 |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>Carnegie-Mellon University<br>Computer Science Dept.<br>Pittsburgh, PA 15213 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>Office of Naval Research<br>Arlington, VA 22217 | | 12. REPORT DATE<br>July 1977<br>13. NUMBER OF PAGES<br>19 |
| 14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)* | | 15. SECURITY CLASS. *(of this report)*<br>UNCLASSIFIED<br>15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)* A maximal vector of a set is one which is not less than any other vector in all components. We derive a recurrence relation for computing the average number of maximal vectors in a set of n vectors in d-space under the assumption that all $(n!)^d$ relative orderings are equally probable. Solving the recurrence shows that the average number of maxima is $O((\ln n)^{d-1})$. We use this result to construct an algorithm for finding all the maxima that has expected running time linear in n (for sets of vectors drawn under our assumptions). For a given set of random points, the result is also used to derive an upper bound on the expected number of points from the set which are on the boundary on the convex hull of the set.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*