A NOTE ON RANDOM GRAPHS, SHORTEST EDGES AND DISTRIBUTED COMPUTING

Michael M. Krieger *

March 26, 1983

Commemorating the 70th birthday of Paul Erdös

ABSTRACT

The average behavior of graph algorithms can often be analyzed by studying the expected structure of random graphs over a suitable sample space. Since the analysis considers "events" distributed throughout the graph, it can provide the basis for designing and analyzing algorithms which are similarly distributed. As such, results about random graphs have special relevence to modelling computation in distributed networks.

In this paper we prove some probabilistic results about shortest edges which are related to shortest paths and spanning trees. With this as a vehicle, we illustrate the connection between random graphs and distributed computing.

*Computer Science Department
Universidade Federal de Minas Gerais
Engineering Systems Department
University of California, Los Angeles

This work was supported in part by the National Science Foundation Grants MCS 78-07468 and MCS 78-18924, the Advanced Research Projects Agency of the Department of Defense under Contract MDA 903-82-C-0064, and the Fulbright Commission (Brazil). For exceptional typing in difficult circumstances, thanks are due to Caruzo de Medeiros Araujo Filho.

INTRODUCTION

This paper is organized as follows: in section A we define the concept of random graphs and sketch the history of research about them; section B considers the application of results about of random graphs to graph algorithms; sections C and D develope specific results concerning the distribution of shortest edges in a random graph and their relation to minimal spanning trees; and section E relates these results to algorithms and distributed computing and poses problems for future research.

Throughout our discussion we will consider underected graphs with labeled vertices and without loops or multiple edges. Notation will follow standard practice, any exceptions being explicitly defined.

A. Random Graphs

We begin by introducing the concept of random graphs and their structure.

Let S be a collection of graphs and C some graph property of interest. If we can show that a randomly selected member G has property C with probability P, then we know something about the structure of a random graph (in S). In particular,if S is infinite and its members can be subdivided into classes S_i of graphs having i vertices, then we may be able to reveal the expected structure of sufficiently large graphs in S by proving results of the form

$$\lim_{n\to\infty} \text{Prob} \{ G \in S_n \text{ has property } C \} = P.$$

A classical example is: C is the property of connectivity and S_n consists of graphs with n vertives and n/2(logn + c) edges. In this case it is known that $P = \exp(-e)^{-c}$.

At this point we note in passing that this suggests a general approach to graph algorithms with improved average behavior: design algorithms to behave optimally for the most probable structure of the input and to work reasonably for the worst case. An early example of this approach is outlined in Krieger (13).

Clearly, then, to exploit the expected structure we must identify both the properties of interest and the appropriate classes of graphs over which to average.

The following three examples will illustrate some important models of random graphs and at the same time highlight developments in this area.

1. Classical Model

As an integrated body of results, the theory of random graphs begins with the seminal papers of Erdös and Renyi (4, 5 and 6). They considered the following model: let $\mathcal{S}_{n,p}$ be the class of graphs with n vertices and p = p(n) edges; then the random graph G_p is said to (not) have property $\mathcal{C}$ if

$$\lim_{n\to\infty} \text{Prob } \{ G \text{ e } \mathcal{S}_{n,p} \text{ has property } \mathcal{C} \} = 1 \ (= 0,\text{resp}) \qquad (1).$$

We will denote this limiting probability by P(p, $\mathcal{C}$).

Generally Erdös and Renyi were concerned with properties of connectivity and the existence of certain subgraphs (e.g., trees and cycles of specified size), for which they proved many results considerably more refined than (1).

In particular, they sought "threshhold" functions for each property: p(x) is a threshhold for property $\mathcal{C}$ if whenever $\lim_{x\to\infty} (f(x)/p(x) = \infty$ and $\lim_{x\to\infty} g(x)/p(x) = 0$, then $P(f,\mathcal{C}) = 1$ and $P(g,\mathcal{C}) = 0$.

Surprisingly, it was found that numerous graph properties indeed exhibit sharp threshholds.

It is convenient to think of these results as saying that almost all sufficiently large graphs with edge density greater than $p(n)/\binom{n}{2}$ have property $\mathcal{C}$. Intuitively the class of graphs consists of those with average vertex degress 2p(n)/n. Thus, if we think of the existence of each property as depending on the edge density, we can observe the "evolution of a random graph" by letting p(n) become an ever faster growing function.

For many years after the initial works, relatively little appeared about the structure of random graphs. In the last few years, however, a host of new papers have appeared. This renewed activity almost surely derives from stimulus in the mid 70's of algorithmic applications of random graphs (see section B, below) and several new structural results, principally that of Posa (16). Leuker (15) includes a reasonably comprehensive bibliography.

2. Geometric Models

In networks and other applications of graphs, the relative position of nodes is relevent and so the distance (or cost) of an edge between two nodes plays a role. It may be the case, as in packet radio, that two nodes are connected if they are within a certain distance.

More specifically we define a sample space as follows: let R be some region (e.g. the unit square or the Euclidean plane) on which points (nodes) are distributed by a Poisson process of fixed density; two nodes v and w are connected by an edge whenever $d(v,w) < r$, a given radius of connection; we are interested in the expected structure of graphs formed this way. Since r parameterizes the edge density of the graphs, we can again ask about the probability of some property as a function of r for sufficiently large graphs.

In the most general setting, such questions fall within the difficult area of stochostic geometry. Specific questions involving connectivity and Euclidean distance were long ago considered as models for epidemics and other biological problems (see, for example, Gilbert (8)). Results paralleling the classical models have been elusive; a few are presented in De Witt and Krieger (2); De Witt (1) surveys the history in some detail and makes a significant start toward a Euclidean theory.

3. Fixed degree model

Let n labeled vertices and some number s be given. We form a (directed) graph by letting each vertex choose s neighbors at random. The space of all $\binom{n-1}{s}^n$ possible graphs thus formed is $D(n,s)$. A space $UD(n,s)$ of undirected graphs may be formed by ignoring direction and assigning to $G \in UD(n,s)$ the sum of the probabilities of graphs $G' \in D(n,s)$ which yielded G. Shamir and Upfal have recently proposed this model, considering asymtotically the relation between and the existence of 1- factors (17).

In many real network situations, this model may be more appropriate than that of Erdös and Renyi since it prevents the occurance of isolated vertices. Note that $UD(n,s)$ is <u>not</u> a subspace of some $S_{n,p}$ since members of $UD(n,s)$ do not all have the some number of edges. For example, members of $UD(n,1)$ may have as few as n/2 or as many as n edges.

B. Algorithms and Random Graphs

As alluded to in the the previous section, by exploiting the most probable structure of graphs, algorithms can be expected to achieve much improved average performance. For this reason, one might have expected the results Erdös and Renyi (5) to have been seized upon by algorithmic analysts. In fact they were ignored for almost a decade and a half. This situation has a two-fold explanation.

First, algorithmic analysis - especially in probabilistic terms - only came into its own in the early seventies under the influence of Knuth(11) and the Cook/Karp results (9). Furthermore, the earlier approach to average analysis principally arose in the context of sorting where the analysis focused on familiar combinatorial properties of permutations and other linear objects.

Secondly, the work of Erdös and Renyi was little known beyond the circle of pure combinatorialists, most of whom had little interest in computing until recently. Moreover, the computer scientest who stumbled upon those early papers would almost surely have found them forboding,and turned away.

Perhaps the earliest computer reference to the possibility of using the ER work was that in Frank and Frisch (7). Actual algorithmic use of random graph results first appeared in Krieger(12) and Karp(10), the latter surveying a variety of problems. The application in(12) provides a convenient illustration. As noted earlier, a graph with average degree $\log n + c$ is connected with probability $\exp(-e^{-c})$. This implies that a random graph with $n \cdot \log n$ edges is almost surely connected. From this, it follows that the shortest $n \cdot \log n$ edges of a graph contain an MST. Thus, by modifying the standard MST algorithms to focus on the smallest $n \cdot \log n$ edges, fast expected time algorithms were achieved.

In a similar vein, improvements have been made for a variety of problems, including shortest paths, colorings, and Hamiltonian path (see references in(15)).

For shortest path and MST problems, short (least cost) edges play a critical algorithmic role. To that end, we analyze their structure in the following sections.

C. The Local Structure of Ordered Graphs

Let G be a complete graph with randomly assigned edge weights and distinguishable vertices 1,2,..., n . Since we are only concerned with relative order of the edges, we may take these weights to be the integers 1,2,...,t where we define $t \equiv n(n-1)/2$ for convenience. The weight or cost of an edge $e = (v,w)$ will be devoted by $c(e)$ or $c(v,w)$. While our ultimate concern is with properties of undirected graphs, it is convenient to treat edges as directed as follows.

Def: An edge (v,w) of G is of type (i:j) if $i \leq j$ and the edge is the i^{th} smallest from vertex v and j^{th} smallest from w. Such an edge is said to be on rank i with respect to v and rank j with respect to w. If $i < j$, then $e = (v,w)$ is directed from v (the head) to w (the tail) and vice versa if $i > j$; when $i = j$, e is bidirectional. By NV(v) we denote the vertex nearest to v, i.e., that vertex w minimizing $c(v,w)$. NE(v) denotes the edge N(v,NV(w)). Note that NE(v) is of type (1:j). The notations NV and NE obviously can be extended when v is replaced by a set of points to denote the vertex and edge closest to some member of the set. S_k consists of the set of all (k:j) edges, $1 \leq j \leq n-1$.

Let F_k be the set of edges of type (k:j) where k is fixed and $k \leq j \leq n-1$. An edge on F_k is called a k-edge. D_k denotes the set of (k:k) edges.

We would like to compute $P_{i,j}$, the probability that (v,w) is of type (i:j) as well as the expected size of F_k over all assignments of 1,2,...,t to the edges of G. Thus we denote by σ_i the value of $\sum_{j=1}^{n-1} P_{i,j}$ and by τ_i that of $\sum_{j=i}^{n-1} P_{i,j}$ over all weight assignments.

For clarity in writing proofs, we define $m \equiv n-2$. $(x)_r$ denotes the falling factorial $x(x-1)\ldots(x-r+1)$. We make use of the following identities

$$\text{(D1)} \qquad \sum (r-1)_x \, (t-r)_{y-x} = \frac{(t)_y}{y+1} \binom{y}{x}^{-1}$$

This follows readily from the binomial product identity (25) in Knuth (11).

$$\text{(D2)} \qquad \sum_{k=0}^{m} \binom{m+k}{k} \cdot 2^{-k} = 2^m$$

Finally, we have

$$\text{(D3)} \qquad \binom{m}{x}\binom{2m}{x+k}^{-1} = (x+k)_k \frac{m\,!}{(2m)_{m+k}} \binom{2m-x-k}{m-k}$$

Lemma 1:

$$P_{i,j} = \frac{2-\delta_{ij}}{2n-3}\binom{n-2}{i-1}\binom{n-2}{j-1}\binom{2n-4}{i+j-2}^{-1}, \qquad \delta_{ij} = \begin{cases} 1 & \text{if } i=j \\ 0 & \text{if } i\neq j \end{cases}$$

Proof: Let e = (v,w) be an (i:j) edge in G and assume e has length r.

Then there are

$$\binom{m}{i-1}\binom{m}{j-1}(r-1)_{i+j-2}(t-r)_{2m-(i+j-2)} \qquad (1)$$

assignment of weights to the edges from v and w which make e an (i:j) edge. The first two terms represent the choices of edges less than r, the third counts the ways to assign 1,2,...,r-1 to these edges, and the last counts the ways to assign weights greater than r to the remaining edges.

Using identity (D1) we sum (1) over r:

$$\sum_{r=1}^{t} (r-1)_{i+j-2}\,(t-r)_{2m-(i+j-2)} = \frac{(t)_{2m+1}}{2m+1} \Big/ \binom{2m}{i+j-2}$$

In the case of i = j we must multiply by 2 to account for the possibility that e is a j - edge for v and i-edge for w. Dividing by $(t)_{2m+1}$ for the 2m+1 edges considered here, yields the result.

Lemma 2:

$$\sigma_k = \frac{2}{n-1} - P_{k,k}$$

Proof: Each vertex has a k-edge. Thus there are n end points of k edges. Edges of type (k:k) account for $2tP_{k,k}$ of them so there are $n - tP_{k,k}$ edges in S_k. The probability of such an edge is thus

$$|S_k|/t = n/t - P_{k,k} = 2/(n-1) - P_{k,k}\ .$$

Lemma 3:
$$\tau_k \approx \frac{1}{n}\left[1 + \binom{2k-2}{k-1}/2^{2k-2}\right]$$

Proof:
$$\tau_k = \sigma_k - \sum_{j=1}^{k-1} P_{j,k} = \frac{2}{n-1} + P_{k,k} - \frac{2}{2n-3}\sum_{j=1}^{k} \binom{m}{k-1}\binom{m}{j-1} / \binom{2m}{k-1+j-1} \qquad (2)$$

Now
$$\binom{m}{k-1}\binom{m}{j-1} / \binom{2m}{k-1+j-1} \approx \binom{k-1+j-1}{j-1} / 2^{k-1+j-1} = \frac{1}{2^{k-1}} \binom{k-1+j-1}{j-1} / 2^{j-1}$$

so using this and identity (D2) we find that the sum in (2) approaches

$$\frac{1}{2^{k-1}} \sum_{\ell=0}^{k-1} \binom{k-1+\ell}{\ell} / 2^{\ell} = \frac{1}{2^{k-1}} \cdot 2^{k-1} = 1$$

whence
$$\tau_k = \frac{2}{n-1} - \frac{2}{2n-3} + P_{k,k} \approx \frac{1}{n} + P_{k,k}$$

Similarly approximating $P_{k,k}$ yields the lemma.

Applying Stirling's approximation to $\binom{2k-2}{k-1}$ gives

Cor: For sufficiently iarge k and n >> k

$$\tau_k \approx \frac{1}{n}\left[1 + \frac{1}{\sqrt{\pi\ (k-1)}}\right]$$

By way of illustration we note a few of these values

$$P_{1,1} = P_{1,2} = \frac{1}{2n-3}\ ;\ P_{1,3} = \frac{n-3}{(2n-3)(2n-5)} \doteq \frac{1}{4n}$$

$$P_{2,2} = \frac{n-2}{(2n-3)(2n-5)};\ P_{2,3} = \frac{3}{2}\,\frac{n-2}{(2n-3)(2n-5)} = \frac{3}{2} P_{2,2}$$

$$t\tau_1 = \frac{3n^2-5n}{2(2n-3)} \approx \frac{3n}{4}$$

$$t\tau_2 = n - tP_{2,2} - tP_{1,2} \approx \frac{5n}{8}\ .$$

D. Minimal Spanning Trees

We first note the standard result about the structure of minimum spanning trees:

Lemma 4: An edge e is in the MST M of G if and only if there is a partition L+R of the vertices of G for which C(e) is least among edges {v,w} satisfying $v \in L$, $w \in R$.

Remark 1: Clearly $NE(v) \in M$, since we only need take $L = \{v\}$, $R = V - \{v\}$, so M contains F_1. Thus of the n-1 edges in M, by Lemma 2 we know that approximately 75% are 1-edges. This constitutes a combinatorial demonstration of a result first proved analytically by DeWitt (1).

The ease of finding these edges suggests the possibility of improved MST algoritms. Indeed the results in this paper grew out of attempts to exploi

and generalize this observation.

If we consider F_1 as a subgraph, we have

Lemma 5:

F_1 is a forest with $|D_1|$ components none of which is an isolated poin

Proof: As a subset of M, F_1 can contain no cycles, i.e., F_1 is a forest. Let v_o be some vertex in F_1 and $v_o, v_1, v_2, \ldots, v_r$ be the path formed in F_1 by putting $v_i = NV(v_{i-1})$ until this procedure terminates. This only can happen when some v_r satisfies $NV(v_r) = v_{r-1}$ i.e., the path terminates in a (1:1) edge. Hence each component has a (1:1) edge.

If (v,w) and (x,y) are (1:1) edges, let $v = v_o, v_1, \ldots, v_p = x$ be the unique path joining v and x (we can assume without loss of generality that $w \neq v_1$ and $y \neq v_{p-1}$.) But then we must have $v_p = NV(v_{p-1})$, v_{p-1} $NV(v_{p-2})$ $\ldots$ $v_2 = NV(v_1)$. This contradicts $v_o = NV(v_1)$.

Corollary: Each component of F_1 consists of a (1:1) edge with a (possibly empty) tree rooted at each end. (see figure 1)

Note that the edges of the two trees are directed opposite to those in some definitions of rooted trees.

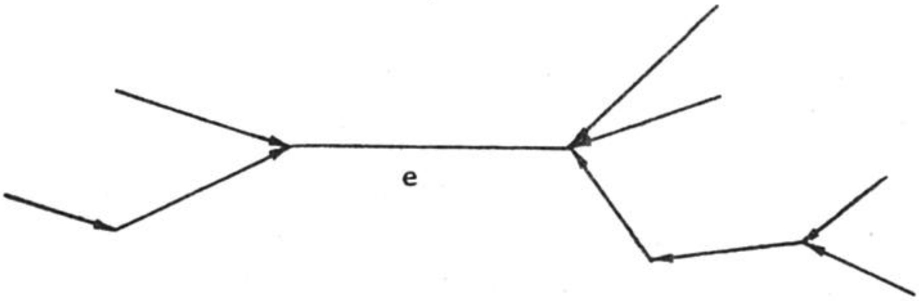

Figure 1: Typical Component of F_1; e is of type (1:1)

We know that the MST M contains F_1 and that the expected number of edges in F_1 is .75 n. This is a good start at building M and leads us to ask if much more of M can be easily found in F_2, F_3, etc? By way of answering this we have the following

Theorem: $\mathrm{Ex}\ \{|\, M \cap F_2 \,|\} > .123n$

To prove this we need two lemmas.

Lemma 6: The probability that an edge e is of type (1:1) and NE(e) is of type (2:k) is $0(2k/3^{k+1}m)$.

Proof: The number of ways the edge f = NE(e) can be of type (2:k) and have c(f) = r is given by

$$2(n-2)\binom{n-3}{k-1}(t-r)_{3(n-2)-1-k}(r-1)_k \qquad (*)$$

In expression (*), 2(n-2) counts the number of ways to choose the endpoints of f and $\binom{n-3}{k-1}$ counts the choices of endpoints for the edges from z which are less than r (see Figure 2). Of the 3n-6 edges from vertices x,y,and z, only these k-1 and e have lengths less than r and these can be assigned in $(r-1)_k$ ways. The remaining edges other than f can be given weights greater than r in $(t-r)_{3n-6-1-k}$ ways.

Summing over r using D_1 yields

$$2(n-2)\cdot\binom{n-3}{k-1}(t)_{3m-6}\Big/(3n-6)\cdot\binom{3n-7}{k} = (t)_{3n-6}\,2k\binom{n-2}{k}\Big/(3n-6)\cdot\binom{3n-7}{k}$$

Dividing by $(t)_{3n-6}$--the total number of ways the lengths may be assigned--and approximating the binomial coefficients completes the proof.

Lemma 7: Let e and e' be edges of type (1:1); let f = NE(e) and f = NE(e'). Then, Prob{f is of type (2:2) and f = f'} = 1/32n.

Proof: Let f = (v,w), e = (x,v), e' = (w,y) and c(e) = r, c(f) = q, and c(e') = s. Because of our assumptions on e,f, and e', all edges in the coboundary of {x,y,v,w} have a cost greater than q. There are 4n - 16 such edges. Also the edges (x,y), (x,w), (y,v) have cost greater than q so we have to assign 4n - 13 costs from the t - q available. Similarly, we must assign the values f and s from q - 1. Finally, noting that x and y could have been chosen in (n-2) (n-3) ways, we have the expression

$$\sum_{q} (n-2)(n-3)(q-1)_2 \ (t-q)_{4n} - 16 + 3$$

for the required probability which we can evaluate as

$$(n-2)(n-3).(t)_{4n-10} \ / \ (4n-10)\binom{4n-11}{2} = \frac{2(n-2)(n-3)}{(4n-10)_3 \ (t)_{4n-10}}$$

This approaches 1/32n as n becomes large.

We now can complete the proof of the Theorem, first noting that $p_{1,1} = \frac{1}{(2n-3)}$ implies that there are about n/4 edges of type (1:1). Each such edge has a nearest neighbor edge in the MST and of type (2:k). Of these, we know n/9 are of type (1:2) by Lemma 6. Using Lemma 7 we note that n/64 of the edges of type (2:2+) meet a type (1:1) edge at either end. These observations suffice to show that the number of members of F_2 in the MST is n/4 - n/9 - n/64 ≈ .123n, proving the Theorem.

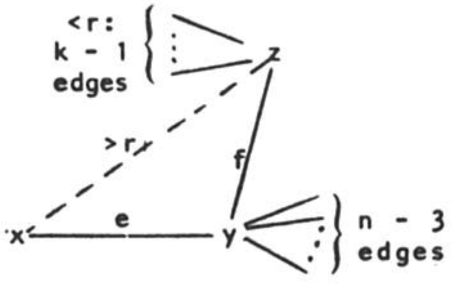

Figure 2: for Lemma 6

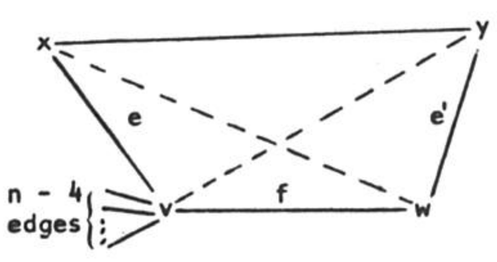

Figure 3: for Lemma 7

E. Shortest edges and algorithms

The forgoing results have implications for general as well as distributed algorithms.

We have seen that essentially 7/8 of the MST can be found among the shortest 2 edges, i.e., in S_2. This suggests that the entire MST will be found in S_κ for a very small k. From an algorithmic point of view, these edges are easy to find. On the one hand, the shortest "few" from each vertex can be flagged when the data is input at virtually no extra computing cost. If the data are constantly being updated, then maintaining a heap of edge distances from each vertex provides an efficient way to access the short edges.

The actual integration of these edges into a complete, efficient MST algoritnm is dependent on the choice of data structure for the graph representation. As such it merits a separate discussion as will be forthcoming in (14).

In distributed environments, nodes will generally 'know' the distance (cost, delay) to their nearest neighbors. Information about more distant nodes may be unavailable, costly, or stale. If this local information can be exploited for required computations, communications overhead can be decreased. Low-cost spanning trees are a commonly used structure supporting network activity and the previous sections suggest that such a tree can be built by each node making connection to its k nearest neighbors.

Of course, our analysis is not complete; namely, we need to know a value of k sufficient to assure connection. Putting k = logn is an attractive guess and raises:

Question 1: On a random weighted graph, what is the relation between E, the smallest $n \cdot \log n$ edges, and $\dot{S}_{\log n}$.

In practice the absolute least cost tree is rarely required; hence we may only want to be sure that k is sufficient to guarantee connectivity at reasonable cost. Thus we ask:

Question 2: For what k is S_k connected? This is answered in (17) for the model of §A.3 above: $\beta > 2$ implies $\lim_{n \to \infty}\{\text{Prob}(G \in UD(n,\beta) \text{ is disconnected}\} = O(n^{-c/2})$.

Question 3: Suppose that m is the cost of the MST and s_k that of the cheapest tree in S_k; what is the expected value of $s_k - m$?

Similar questions and analysis apply to shortest paths, which are perhaps even more fundamental to networks. For example, De Witt and Krieger(3) showed that generally shortest paths have relatively few edges. In the present context, we would like to have answers to

Question 4: If E is the set of edges in the shortest path from v to w, what is the probability P_k that $E \subseteq S_k$?

Clearly answers to these and similar questions will provide tools for more efficient computation.

References

1. H.K. DeWitt, "The Theory of Random Graphs with Applications to the Probabilistic Analysis of Optimization Algorithms", UCLA, 1977 (dissertation)

2. H.K. DeWitt and M.M. Krieger, "Expected Structure of Euclidean Graphs" in Algorithms and Complexity: New Directions and Recent Results, ed. J.F. Traub Academic Press, New York. 1976, p.451.

3. ———, "Expected Structure of Shortest Paths and Algorithm Behavior", in Proc. 10 th S-E Conf. Combinatorics, Graph Theory, and Computing, pp 367-380.

4. P. Erdös and A. Renyi. "On Random Graphs, I", Publicationes Mathematicae Debrecen 6 (1959), pp. 290-297; reprinted in Paul Erdös: The Art of Counting J. Spencer, ed., MIT Press, Cambridge, Mass., 1973.

5. ———, "On the Evolution of Random Graphs", Magvar Tud. Akad. Mat. Kut. Int. Közl. 5 (1960), pp. 17-61; reprinted as above.

6. ———, "On the Strength of Connectedness in Random Graphs", Acta Math. Acad. Sci. Hung. 12 (1961) pp. 261-267.

7. H. Frank and I. Frisch, Communications, Transmission, and Transportation Networks, Addison-Wesley, (1971) p. 390.

8. E.N.Gilbert, "Random Plane Networks", J.SIAM, 9 (1961) pp. 533-543.

9. R.M.Karp,"Reducibility amonq combinatorial problems,"in Complexity of Computer Computations, R.E.Miller and J.W.Thatcher, eds., Plenum Press, New York, 1972, pp. 85-104.

10. ———, "The Probabilistic Analysis of Some Combinatorial search Problem: as in 2 above.

11. D.E.Knuth, The Art of Computer Programming, Vol. 1: Fundamental Algorithms, Addison-Wesley, Reading, Mass., 1973, p. 94 et seg.; see also analysis in Vol. 2 and 3.

12. M.M.Krieger, "Notes on Algorithmic Analysis", UCLA Computer Science Department, 1974-75.

13. ———, "Random Graphs and Minimal Spanning Tree Algorithms", in Algorithms and Complexity: New Directions and Recent Results, ed. J.F.Traub, Academic press, New York, 1976, p.471.

14. ———, "Graphs, Data Structures, and Spanning Trees: A Tutorial for Fast Implementations" (in prep.)

15. G.S.Leyker, "Optimization Problems on Graphs with Independent Random Edge Weights", SIAM J. Comp., pp. 338-351.

16. L.Posa,"Hamiltonian circuits in random graphs",Discrete Math., 14 (1976) pp. 359-364.

17. E.Shamir and E.Upfal, "One-Factor in Random Graphs Based on Vertex Choice", Discrete Math.