OPERATIONS RESEARCH CENTER

working paper

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

An Algorithm for Reliability Analysis

of Planar Graphs

by

Daniel Bienstock

OR 141-85 September 1985

The research reported in this paper was partially carried out at MIT and was funded in part by grant NSF-ECS-8316224.

Abstract

We give an algorithm for the computation of K-terminal reliability in planar graphs, whose worst-case complexity is strictly exponential in the square root of the total number of nodes.

An Algorithm for Reliability Analysis of Planar Graphs

Daniel Bienstock, GSIA, Carnegie-Mellon University

1. Introduction.

The K-terminal reliability problem can be stated as follows: given an undirected graph $G=(V,E)$ whose arcs are erased independently with known probabilities, and a subset K of V, compute the probability that K remains connected. This problem is known to be #P-hard [1], and, not surprisingly, the best algorithm has complexity that is strictly exponential in $|V|$ [4].

The case of planar G has received attention recently. Even though this restriction of the general problem is still #P-hard, even when $|K|=2$ [10], the complexity of the case $K=V$ still unknown. In this paper we present an algorithm for computing K-terminal reliability of planar graphs, whose complexity is at most strictly exponential in the *square root* of $|V|$, a large improvement over the general case. Our algorithm uses elements of two closely related reliability algorithms, those of Rosenthal [11] and Fratta and Montanari [6], as well as Miller's version of the planar separator theorem [7] and pertinent properties of planar graphs.

This paper is organized as follows: in Section 2 we describe the Rosenthal, Fratta and Montanari algorithms; Sections 3 and 4 present relevant characteristics of planar graphs and Section 5 contains our algorithm.

2. The algorithms of Rosenthal and Fratta and Montanari

The algorithms of Rosenthal (1977) and Fratta and Montanari (1976) share a basic idea which we call **RFM**. In this section we describe RFM as it appears in Rosenthal's algorithm, although we will mention the Fratta and Montanari version later.

First we need some notation (this notation is different from Rosenthal's).

Let X be an arbitrary set. A *labeled partition* of X is an ordinary partition of X where some of the blocks are labeled with an asterisk. For instance, (134*,25,6*) is a labeled partition of {1,2,3,4,5,6}.

Let G be a probabilistic graph, that is a graph whose arcs and nodes are probabilistically erased. A *state* of G will be a specification of which arcs and nodes are operative and which are failed. In the context of this paper, only arcs will fail. An *event* will be a collection of states.

Let H be a probabilistic graph containing, among its nodes, a subset K and a subset S (which may intersect). A state **s** of H will be called a (K,S)-bond state if **s** implies that every node of K remains connected to at least one node of S. The effect on S of a (K,S)-bond state may be represented by a labeled partition of S as follows: let x be a labeled partition of S. Then we interpret x as implying that

(i) Every block of x remains connected, but disconnected from the rest of S.

(ii) Every node of K remains connected to exactly one labeled block of x, and every labeled block of x remains connected to at least one node of K.

The collection of all (K,S)-bond states corresponding to a given labeled partition will be called a (K,S)-bond event (and will be represented by that labeled partition). If x is a labeled partition of S, then P(x) will denote the probability of the (K,S)-bond event x. The vector of entries P(x) (one for each labeled partition x) will be called the vector of (K,S)-bond probabilities of H. We will drop K and S and H from our notation whenever the context will make it unambiguous.

Armed with the above definitions, we can describe RFM. Let H be a graph whose arcs are independently erased. Suppose H contains two distinguished subsets of nodes S and K. We want to compute the (K,S)-bond probabilities of H.

To reduce this problem to a smaller one, consider a node cut C of H that splits H into subgraphs H(1) and H(2), both of which are defined to contain C (any arcs that link nodes of C are assigned arbitrarily to H(1) or H(2)).

For i=1,2, let K(i) be the subset of K contained in H(i); and similarly define S(i). Then (recursively) compute, for i=1,2, the (K(i), S(i) $\cup$ C)-bond probabilities of H(i) (if one of the K(i)'s is empty then all of the corresponding bond events will be unlabeled partitions). These parameters can now be used to compute the (K,S)-bond probabilities of H as follows.

Let x, y be two bond events, respectively of H(1) and H(2). If both x and y occur simultaneously, either

(1) Some nodes of K remain isolated from S, or

(2) A certain (K,S)-bond event of H occurs. We denote this event by x*y.

In general, for each (K,S)-bond event z of H, we can write

$$P(z)=\sum P(x)P(y), \qquad (1)$$

where the sum is taken over all bond events x and y such that z=x*y.

Consequently, in order to compute the (K,S)-bond probabilities of H, we enumerate all pairs of bond events x, y of H(1) and H(2) respectively, that have positive probability, and use identity (1). We will refer to this enumeration as the *splicing* of H(1) and H(2).

The * operator can be computed in time linear in w, the number of nodes of S$\cup$C [2]. Therefore, if for i=1,2, there are N(i) positive bond probabilities of H(i), then the splicing operation will take time O(wN(1)N(2)).

This concludes our description of RFM. For a more thorough discussion the reader is referred to [2]. Here we only point out that RFM is, in general, inefficient when applied to graphs that are very dense or contain very dense subgraphs.

3. Some properties of planar graphs.

We say that a graph is *planar* if it can be drawn on the plane without its edges crossing. Such a drawing is called a *layout* of the graph. Given a layout of a planar graph, the edges of the graph partition the plane into several regions, called the *faces* of the graph. With one exception, all the faces are bounded. The bounded faces are called inner faces and are bounded by cycles of the graph (which are called inner *facial cycles*) whenever the graph is 2-connected. In that case, the unbounded face, also called the outer face, is also delimited by a cycle, which is called the outer facial cycle.

An *embedding* of a planar graph is a description of the graph via adjacency lists, where for each node we list its neighbors in (say) clockwise order (note: an embedding may correspond to more than one layout. However, all such layouts will be equivalent in the sense that they will have the same facial cycles, and we can go from one layout to another by flipping the graph "inside out" about facial cycles).

A planar graph is *maximal* if we cannot add any new edges without destroying planarity. Given an embedding of an arbitrary planar graph, we can transform the graph into a maximal planar graph in linear time, by performing depth first search in clockwise order, starting from any node, and inserting new edges into the adjacency lists. In a maximal planar graph, each facial cycle has exactly three edges.

Consider a cycle of a planar graph. Given an embedding of the graph, the removal of the cycle partitions the graph into two regions, the *strict interior* and the *strict exterior* of the cycle (the choice of which of the two regions to be called interior is arbitrary, unless a specific layout is being used). The interior of the cycle is the strict interior, together with the cycle itself. The exterior of the cycle is similarly defined.

There are several fairly recent algorithmic results which are of particular importance. The first one concerns the planarity testing problem: given a graph, is it planar? This question can be decided in time linear in the total number of nodes [7]. A related problem is that of finding an embedding of a planar graph. This task can also be performed in linear time [3].

The most relevant result (to us) concerns planar separator theorems, which we state in abridged form. Let G be an n-node planar graph. Then, by deleting $O(\sqrt{n})$ nodes of G, we can partition G into two subgraphs each of which has roughly $n/2$ nodes (Lipton and Tarjan, 1979 [8]). Of special importance is Miller's (1984) version of the planar separator theorem [9], which is once more stated in abridged form: let G be an n-node graph, each of whose facial cycles has length at most z. Then there exists a simple cycle with at most $4\sqrt{(2zn)}$ nodes whose strict interior contains at most $2/3\ n$ nodes and at least $1/3\ n$ nodes (and consequently, we can say the same about the strict exterior of the cycle). Such a cycle is called a *cycle separator* and can be found in linear time. Thus, if G is maximal, in linear time we can find a cycle separator with at most $c\sqrt{n}$ nodes, where $c=4\sqrt{6}$.

4. Planar graphs and bond probabilities

Consider a planar graph bounded by an outer facial cycle with nodes numbered 1,2,...,m as they appear clockwise in that cycle. Suppose we erase some of the arcs of the graph. This erasure will induce a partition of the nodes of the outer facial cycle according to their connected components. How many partitions of $\{1,2,\ldots,m\}$ can be achieved in this manner? A crude upper bound is the total number of partitions of an m-element set, a quantity that grows nearly as fast as m! [5].

However, this is a very weak upper bound. In order to see why, let us consider an example. Suppose m=6. Then the partition (13,245,6) cannot be achieved by erasing arcs, since the first and second blocks "cross" each other. On the other hand, the partition (12,36,45) may be achieved.

In order to compute a tighter upper bound on the number of achievable partitions, let us take a strictly combinatorial approach. Suppose we take a circle containing a set $M=\{1,2,\ldots,m\}$ of selected points, numbered to reflect clockwise ordering. We will say that a partition x of M is *non-crossing* if, as we travel clockwise around the circle, no two blocks of x ever alternate (i.e., no two blocks of x ever "cross"). We will indicate the number of non-crossing partitions of $\{1,2,\ldots,m\}$ by b(m). In the remaining part of this section, we will compute b(m).

Consider point m. Given a non-crossing partition, let k be the highest numbered point in the same block as m (k=0 if m is in a block by itself). Then points 1,2,...,k are non-crossing partitioned, and the same holds for points k+1,...,m-1. Setting b(0)=1, we conclude that

$$b(m)=\sum_{k=0}^{m-1} b(k)b(m-k-1) \quad \text{for } m>0. \tag{2}$$

Now set $g(z)=\sum b(m)z^m$. Equation (2) will yield

$$g(z)=1+z\,g^2(z).$$

This equation, together with the boundary condition b(0)=1, give

$$g(z) = (1-(1-4z)^{1/2})/2z,$$

which implies that (see [5] for related problems)

$$b(m) = 2^{2m - O(\log m)}.$$

Consequently, the number of non-crossing partitions is (asymptotically) a negligible fraction of the total number of partitions.

Now let us consider a probabilistic planar graph H containing a special subset of nodes K, and suppose H is bounded by a (simple) outer facial cycle S of m nodes. How many (K,S)-bond probabilities can be strictly positive? We just saw the number of possible partitions of S that can be achieved by deleting arcs of H is at most strictly exponential in m. Moreover, if a partition x of S has N blocks, then there are at most $2^N \le 2^m$ ways of labeling the blocks of x. We conclude that the number of positive bond probabilities is at most 2^{3m}.

5. An algorithm for K-terminal reliability computation in planar graphs.

There are several algorithms for K-terminal reliability analysis in planar graphs that achieve a complexity strictly exponential in the square root of the total number of nodes. We have chosen the specific algorithm given below because it requires the simplest exposition.

We will first consider a more general problem. Let G be an n-node planar graph containing a special subset of nodes K. We will assume that we are given an embedding of G. We want to compute the (K,S)-bond probabilities of G, where S is the node set of a (simple) facial cycle of G, without loss of generality the outer facial cycle of G. Let $|S|=m$, and set $T(n;m)$=worst-case running time of algorithm PL given below.

Theorem 1

$$T(n;m) \le 2^{a\sqrt{n}+6m}, \qquad (3)$$

for a certain constant a.

Proof: The proof will be by induction on n, since $m \le n$. Let c be the constant that appears in the cycle separator theorem. Now, relationship (3) holds if both n and m are bounded above by constants, provided we choose a large enough constant a. Consequently, let us assume that $n \ge n'$, where

$$2n'/3 + c\sqrt{n'} + 1 + 3\log n' < 3n'/4,$$

and also that (3) holds for arguments smaller than n'. Let $X(N,M)=2^{a\sqrt{N}+6M}$. Below we will prove that

$$\begin{aligned} T(n;m) \le \max\{ & O(n^2)\, 2^{6m+6c\sqrt{n}} + O(n)\, X(2n/3+c\sqrt{n}\,;\, m+c\sqrt{n}), \\ & O(n)\, 2^{3m+6c\sqrt{n}} + 2X(2n/3+c\sqrt{n}+1\,;\, m+c\sqrt{n}+1), \\ & 2^{3c\sqrt{n}} X(2n/3\,;\, m) + X(2n/3\,;\, c\sqrt{n})\} + O(n). \qquad (4) \end{aligned}$$

We will conclude from (4) that

$$\begin{aligned} T(n;m) &< 2^{6m+c\sqrt{n}+3\log n} + 2^{a\sqrt{(3n/4)}+6m+6c\sqrt{n}+6} + O(n) < \\ &< 2^{[a\sqrt{(3/4)}+6c]\sqrt{n}+6m+7} < 2^{a\sqrt{n}+6m} \end{aligned}$$

if a is chosen large enough. Pending the proof of (4), the proof of Theorem 1 is concluded. ♦

Next we describe algorithm PL and prove (4). Let us denote by a *block* a planar graph with an outer facial cycle that is simple. Thus the input to PL is an embedding of an n-node block G containing a special subset of nodes K. G is bounded by an outer facial cycle S with m nodes.

Algorithm PL

(1) If G is not maximal, make it so by adding new edges which are perfectly unreliable. This task can be carried out in linear time (as outlined in Section 3). Moreover, all the nodes of G will remain in the interior of S.

(2) Find a cycle separator C of G. Let EXT be the subgraph of G contained in the interior of S but also in the exterior of C. Similarly, let INT be the subgraph of G contained in the interior of both S and G. There are three possible cases:

(i) $|S \cap C| > 1$.

We can write

$$EXT = G_1 \cup G_2 \cup \ldots \cup G_j \cup P_1 \cup P_2 \cup \ldots \cup P_h ,$$

where each G_i is a <u>block</u> whose outer facial cycle is made up of portions of S and C, and each P_k is a maximal <u>path</u> contained in both S and C (along which the clockwise direction on C coincides with the clockwise direction on S). Similarly, we can write

$$INT = G_{j+1} \cup G_{j+2} \cup \ldots \cup G_w \cup P_{h+1} \cup P_{h+2} \cup \ldots \cup P_z .$$

If an arc of C appears both in a block of EXT and in a block of INT, we replace that arc in one of the blocks by a perfectly unreliable arc.

Figure 1 - Case (i).

For $1 \le i \le w$, let S(i) and K(i) be, respectively, the outer facial cycle of, and the subset of K contained in G_i. It is not difficult to see that we can generate the blocks G_i and the respective sets K(i) and S(i) in linear time (and also, $w=O(n)$). Next, we (recursively) compute the (K(i),S(i))-bond probabilities of G_i, for $1 \le i \le w$. Clearly, this task will take time, at most,

$$O(n)X(2n/3 + c\sqrt{n} ; m + c\sqrt{n}),$$

since each block contains at most 2n/3 nodes not counting those of C.

Having computed these bond probabilities, we now <u>splice</u> (terminology of Section 2) the blocks G_i one by one until recovering the graph G. That is, we start with graph $H_1 = G_1$ and at step i we splice H_i with a new G_j to obtain H_{i+1}, until after $w-1$ steps we have H_w=G. It is not difficult to see that we can order the blocks G_j so that at every step H_i is a block. Since at each step, the blocks spliced contain at most all of C and S, the complexity of each splicing is at most

$$O(n)\, 2^{6m+6c\sqrt{n}},$$

where we use the result of Section 4 concerning the number of positive bond probabilities of a block, and the result in Section 3 describing the complexity of a splice operation.

We conclude that the overall complexity of case (i) is at most

$$O(n^2)\, 2^{6m+6c\sqrt{n}} + O(n)\, X(\, 2n/3 + c\sqrt{n}\, ;\, m + c\sqrt{n}) \;=\; T_1(n;m).$$

(ii) $|S \cap C| = 1$.

Let v be the node common to S and C. In this case the interior of C (with the exception of v) lies in the strict interior of S (see Figure 2(a)). As we travel clockwise around S, let u, w be the nodes immediately preceding and following v, respectively. Similarly, as we travel clockwise around C, let x, z immediately precede and follow v. Then we split v into two nodes v' and v" which are connected by a perfectly reliable arc, with v" immediately following v'. The old neighbors of v become neighbors of v' or v" as follows:

(a) For each arc (v,s) with s located clockwise between w and z (inclusive) there is an arc (v",s) with the same reliability as (v,s).

(b) For each arc (v,t) with t strictly following z there is an arc (v',t) with the same reliability as (v,t).

Now we interpret cycles S and C as going through both v' and v", in that order (see Figure 2(b)).

Figure 2 - Case (ii).

It is clear that after the splitting of v both EXT and INT are blocks (if EXT and INT share an arc of C, replace that arc in, say, EXT, by a perfectly unreliable arc). Hence, we have reduced case (ii) to an instance of case (i) with only two blocks, at the cost of adding a single additional node to the problem. Thus, the time required to find the bond probabilities of EXT and INT is at most

$$2X(2n/3 + c\sqrt{n} + 1; m + c\sqrt{n} + 1).$$

Next, we splice EXT and INT. Since EXT has at most $2^{3m + 3c\sqrt{n} + 3}$ positive bond probabilities and INT has at most $2^{3m + 3}$ positive bond probabilities, the splice operation will take time at most $O(n)2^{6m + 6c\sqrt{n}}$ (where we include in O() the work involved in consolidating v' and v'', which is at most proportional to the total number of positive bond probabilities of G). Thus, the total complexity of case (ii) is at most

$$O(n)\, 2^{3m+6c\sqrt{n}} + 2X(2n/3 + c\sqrt{n} + 1; m + c\sqrt{n} + 1) \equiv T_2(n;m).$$

The last case is (iii), $|S \cap C| = 0$.

In this case C and its interior lie in the strict interior of S. Notice that we cannot analyze EXT directly since C and S are two disjoint facial cycles of this graph.

Nevertheless, we <u>can</u> analyze INT recursively with PL, that is compute the (K',C)-bond probabilities of INT, where K' is the subset of K contained in the interior of C. These probabilities can be computed in time $X(2n/3 + c\sqrt{n}\,;\, c\sqrt{n})$. Having carried out this task, let y be a labeled partition of C (i.e., a (K',C)-bond event) with positive probability. Suppose we contition on y occurring in INT. Then all the nodes in each block of y remain connected (via INT) and therefore, so far as EXT is concerned, we may contract each such block into a single node (this observation is implicit in both the Rosenthal and Fratta and Montanari versions of RFM). Let EXT(y) be the obtained graph. The key observation is that, since partition y must be non-crossing, the graph EXT(y) is in fact planar (and we can obtain an embedding for EXT(y) from the one for EXT in linear time). Notice that in graph EXT(y) the cycle C has been "contracted", and now we can analyze EXT(y) recursively.

Figure 3 - Case (iii).

Thus, let K''(y) be the subset of K in the strict exterior of C, together with one new node corresponding to each labeled block of y. We recursively compute the (K''(y),S)-bond probabilities of EXT(y). This operation will at most take time $X(2n/3\,;\,m)$.

Further, each (K''(y),S)-bond event of EXT(y) is a (K,S)-bond event of G, since it corresponds to a labeled partition of S. We simply keep track of the conditioning by multiplying each bond probability of EXT(y) by the bond probability of y in INT.

The number of graphs EXT(·) that arise is at most the number of non-crossing partitions of |C| points. Thus, the overall workload for case (iii) is at most

$$X(2n/3; c\sqrt{n}) + 2^{3c\sqrt{n}}X(2n/3\,;\,m) \equiv T_3(n\,;\,m).$$

We have now concluded the case analysis for step (2) of algorithm PL. We conclude that

$$T(n\,;\,m) \le \max\{\,T_1(n\,;\,m)\,,\ T_2(n\,;\,m)\,,T_3(n\,;\,m)\,\} + O(n),$$

i.e., equation (4). ♦

Now let G be a probabilistic n-node planar graph and K a subset of the nodes of G. We wish to compute the K-terminal reliability of G. Without loss of generality G is maximal; let S be a facial cycle containing at least one node of K. Once more, without loss of generality S is the outer facial cycle.

Using algorithm PL, we compute the (K,S)-bond probabilities of G. The K-terminal reliability of G will be the sum of bond probabilities corresponding to labeled partitions with exactly one labeled block (there are at most five such partitions since S has length three and one of its nodes is in K). Thus the time required to compute the K-terminal reliability of G is at most

$$O(2^{a\sqrt{n}}).$$

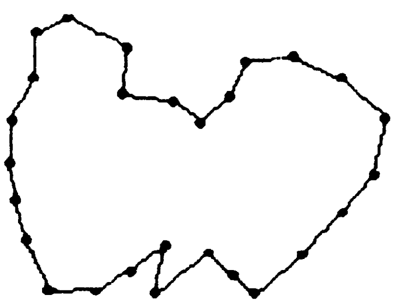
6. Conclusions

Is our algorithm efficient? It is not difficult to see that any algorithm based on RFM will have complexity at least strictly exponential in $\sqrt{n}$, in the worst case: mesh graphs are among those for which all non-crossing partitions can be achieved by erasing arcs, and cuts of size $\sqrt{n}$ must be used when analyzing such graphs.

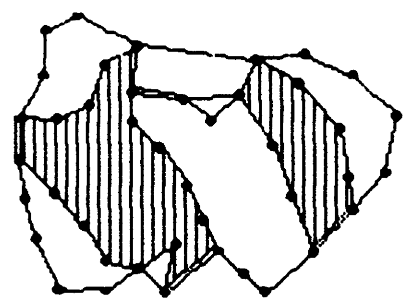
We want to stress the fact that the size of K does not play a role in the complexity of our algorithm. Since the general case is #P-hard, this fact may add some credibility to the conjecture that the case $|K|=n$ is also #P-hard.

* * * * * * * * * * *

Figure 1

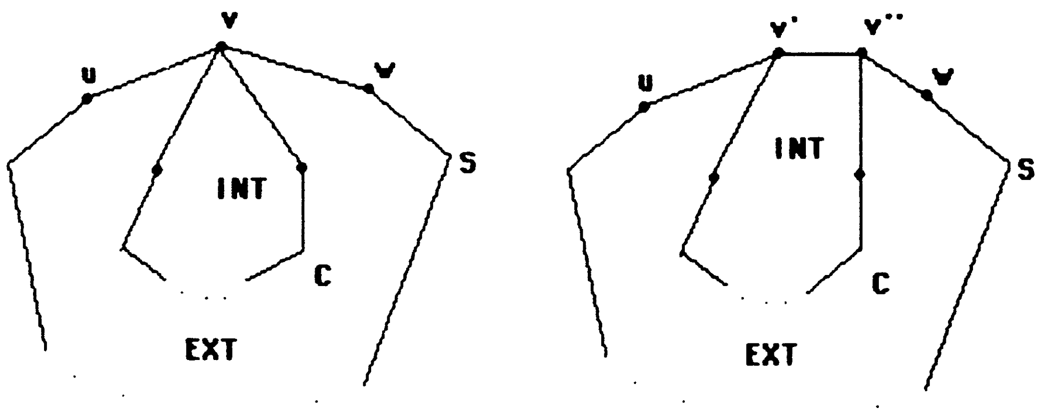


(a) Original graph with S shown.



(b) Both S and C shown. INT is the shaded region.

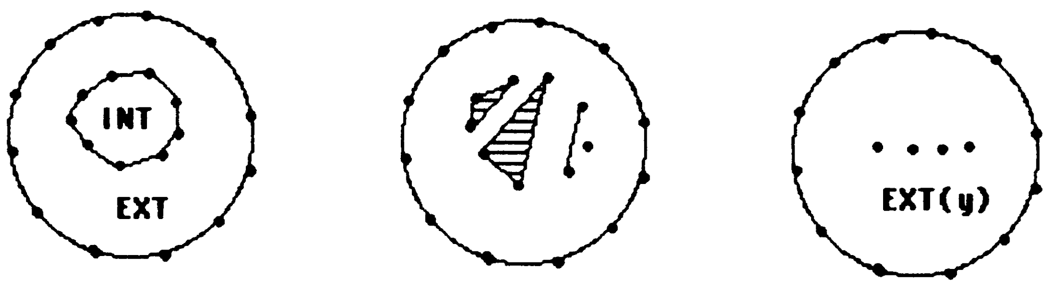
Figure 2



(a) Original graph and cut.

(b) After expansion of v.

Figure 3



(a) Original graph with S and C shown. Here |C|=9.

(b) A four-block partition y occurs in INT.

(c) INT is shrunk into 4 nodes in EXT(y).

References

[1] M.O. Ball, The Complexity of Network Reliability Computations, Networks 10 (1980) 153-165.

[2] D. Bienstock, Some Lattice-Theoretic Tools for Network Reliability Analysis, Working paper OR-138-85, MIT Operations Research Center (1985).

[3] K.S. Booth and G.S. Lueker, Testing the Consecutive Ones Property, Interval Graphs, and Graph Planarity using PQ-Tree Algorithms, J. Comput. Syst. Sci. 13 (1976) 335-379.

[4] J.A. Buzacott, A Recursive Algorithm for Finding Reliability Measures Related to the Connection of Nodes in a Graph, Networks 10 (1980) 311-327.

[5] L. Comtet, Advanced Combinatorics, D. Reidel Publishing Co., Doldrecht, Holland (1974).

[6] L. Fratta and U. Montanari, Decomposition Techniques for Evaluating Network Reliability, IBM report RC5923 (1976).

[7] J.E. Hopcroft and R.E. Tarjan, Efficient Planarity Testing, J. Assoc. Comput. Mach. 21 (1974) 549-568.

[8] R.J. Lipton and R.E. Tarjan, A Separator Theorem for Planar Graphs, SIAM J. Appl. Math. 36 (1979) 177-189.

[9] G. Miller, Finding Small Simple Cycle Separators for 2-Connected Planar Graphs, Proc. of 16th ACM Symp. on Theory of Computing, (1984) 376-382.

[10] J.S. Provan, The Complexity of Reliability Computation in Planar and Acyclic Graphs, in preparation.

[11] A. Rosenthal, Computing the Reliability of Complex Networks, SIAM J. Appl. Math. 32 (1977) 384-393.