System Design and Assessment Notes

Note 16

11 January 1974

AN EFFICIENT ALGORITHM FOR THE SYMBOLIC SOLUTION OF NETWORK RELIABILITY

by

WILLIAM P. DOTSON, JR.
Captain, USAF
AFWL, ELC

ABSTRACT

An efficient algorithm for the analysis of unreliable networks is presented. The efficiency of the algorithm is estimated to be four to eleven times as great as that of the most efficient algorithm previously reported. The algorithm herein simultaneously solves for the symbolic expression of both terminal reliability and unreliability. These expressions may subsequently be numerically evaluated for any desired set of node probabilities.

The dual to the algorithm is proposed as a means for achieving tighter upper bounds on the solutions for networks in which it is not practical to run the algorithm to termination.

Network reliability algorithms have wide applications in many areas. The most pertinent are situations in which reliability data on total networks (systems) is not directly available; but reliability data on the nodes (subsystems) is available.

INDEX

TABLES

AN EFFICIENT ALGORITHM FOR THE SYMBOLIC SOLUTION OF NETWORK RELIABILITY

I. INTRODUCTION

We will represent a network by an oriented graph with weighted nodes and unweighted arcs. The arcs represent such things as a pair of wires or a line of sight radio transmission path assumed to be completely reliable. The weighted nodes represent such things as an electrical system with some associated failure rate, a disturbance on a transmission path or a transient upset in a piece of equipment with some associated rate of occurrence. The weight assigned to each node is its probability of existence and is subscripted with the node label. We make the usual assumption that node failures are uncorrelated. This kind of probabilistic graph can be made quite general by use of the symbols shown in Table I.

The potential applications for algorithms which analyze the reliability of networks are manifold both for commercial and military systems. Consider for example the analysis of a long-distance HF radio network in which ionospheric disturbances are possible; or the radio stations themselves are subject to failure. Or consider a microwave link network in which the propagation loss on some paths is unacceptably high due to a storm or a temperature inversion. Still another example would be a switched network in which the switches are subject to failure. One final example would be the use of such algorithms to determine the existence probability of a node by representing the internal subsystems of that node as a probabilistic graph; i.e., the subsystems of of a missile, an aircraft, or a microwave communications facility.

II. APPROACH

The approach used in the proposed algorithm is best illuminated by means of an example. We shall use the example shown in Figure 1 which is similar to the illustrative example of reference [1] by the symbol conventions of our Table I. The objective now is to find the symbolic expression for the terminal reliability, i.e., the probability that connectivity exists from Node 1 to Node 7.

We use a special form of a Veitch diagram [2] to find the required reliability expression. Veitch diagrams permit a labelling of every elementary event pertaining to the network. Now if we define

$$K : \text{Number of weighted nodes in the network} \tag{1}$$

then the number of elementary events in the Veitch diagram is:

$$E = 2^K \tag{2}$$

The special form of the Veitch diagram used here is simply a row vector with E elements. The diagram for the example of Figure 1 is shown in Figure 2. The labelling of the diagram is accomplished in terms of the subscripts of each weighted node in the network. The labelling procedure can be accomplished correctly in many ways, however, there is one procedure which is particularly convenient and which forms the basis for the proposed algorithm. We shall return to this point after completing the example in a more traditional manner.

To perform the labelling correctly, it is only required that each node subscript cover exactly E/2 elements in the diagram and that no two subscripts cover the same space. The probability of occurrence of any elementary event in the diagram can now be formed by proper interpretation of the labels covering that element. For example, the left most element in Figure 2 implies that the probability of occurrence of the event (Nodes 2, 3, 4, 5, and 6 exist) is:

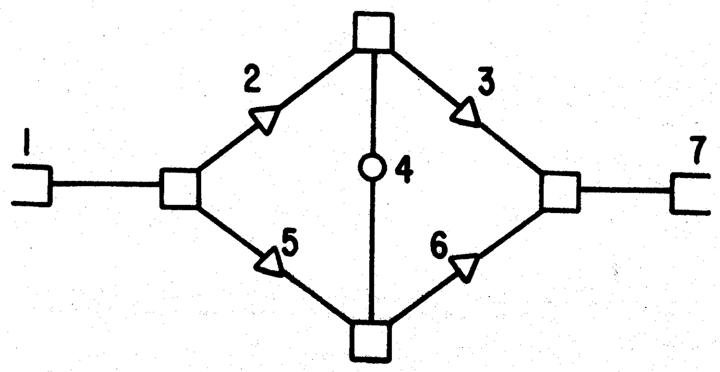

Fig. 1. An Example of a Probabilistic Graph.

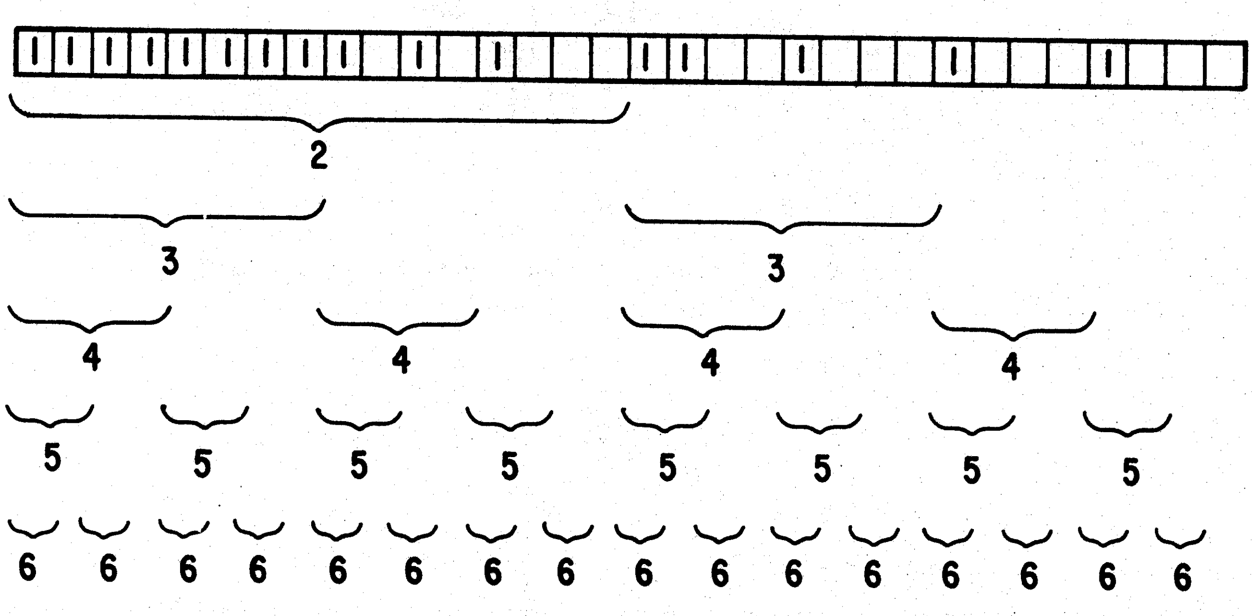

Fig. 2. Veitch Diagram for the Network of Fig. 1.

Table I, Symbols

———	:	UNWEIGHTED ARC
□	:	UNWEIGHTED NODE
○	:	WEIGHTED UNDIRECTED NODE
▷	:	WEIGHTED DIRECTED NODE
⊐	:	SOURCE OR TERMINAL NODE, UNWEIGHTED

NOTE: A WEIGHTED ARC CAN BE DRAWN AS:

—○—	:	UNDIRECTED
—▷—	:	DIRECTED

$$\{2\ 3\ 4\ 5\ 6\} \Rightarrow p_2p_3p_4p_5p_6$$

Similarly, the right most element in Figure 2 implies that the probability of occurrence of the event (Nodes 2, 3, 4, 5, and 6 do not exist) is:

$$\{\bar{2}\ \bar{3}\ \bar{4}\ \bar{5}\ \bar{6}\} \Rightarrow q_2q_3q_4q_5q_6$$

The probability of non-elementary events in the diagram can likewise be found from the labelling. For example, the probability of the event (Node 2 exists, corresponding to the first 16 elementary events) is simply:

$$\{2\} \Rightarrow p_2$$

Similarly, the probability of the event (Nodes 2 and 3 exist, corresponding to the first 8 elementary events) is just:

$$\{2\ 3\} \Rightarrow p_2p_3$$

It remains to find the terminal reliability of the network from Node 1 to Node 7. This is accomplished by summing the probabilities of all events in the Veitch diagram which are favorable to connectivity. The favorable events are denoted by placing a "1" in the appropriate elementary event location of the Veitch diagram.

We now return to the comment made earlier concerning our particular labelling procedure. Since how one begins labelling the Veitch diagram is arbitrary as long as the rules for correct labelling are observed, we postulate the existence of an algorithm which will:

a) discover the label for a single favorable event (hopefully non-elementary) on the Veitch diagram.

b) said label will cover the largest possible number of elementary events.

Such an algorithm is nothing more than one which discovers the shortest path through a network. For the purpose of future reference, we shall call this algorithm PATH.

Applying PATH to the example of Figure 1, the non-elementary event {2 3}is found to be a short path and, therefore, favorable to connectivity. Hence the probability of connectivity from Node 1 to Node 7 is:

$$P_{1-7} \geq p_2 p_3 \tag{3}$$

Now we imagine the construction of a Veitch diagram, with E elementary events, wherein the labelling is begun with the event discovered by PATH, namely {2 3}. The construction is shown in Figure 3.

We now observe that the remainder of the Veitch diagram is described by the events:

$$\{ \overline{2} \} , \{ 2\ \overline{3} \}$$

Now if we apply PATH to the network of Figure 1 subject to the event $\{ \overline{2} \}$ the event { 5 6 } is discovered to be favorable to connectivity. Hence, equation 3 becomes:

$$P_{1-7} \geq p_2 p_3 + q_2 p_5 p_6 \tag{4}$$

The event { 5 6 } discovered in the network of Figure 1 subject to the event $\{ \overline{2} \}$ implies a second intermediate Veitch diagram as shown in Figure 4. The remainder of this diagram is described by the events:

$$\{ \overline{2}\ \overline{5} \} , \{ \overline{2}\ 5\ \overline{6} \}$$

Repeating the process for the network of Figure 1 subject to the event $\{ 2\ \overline{3} \}$ we find the favorable event { 5 6 }. As before this implies that equation 4 becomes:

$$P_{1-7} \geq p_2 p_3 + q_2 p_5 p_6 + p_2 q_3 p_5 p_6 \tag{5}$$

The remainder of this third intermediate Veitch diagram is described by the events:

$$\{ 2\ \overline{3}\ \overline{5} \} , \{ 2\ \overline{3}\ 5\ \overline{6} \}$$

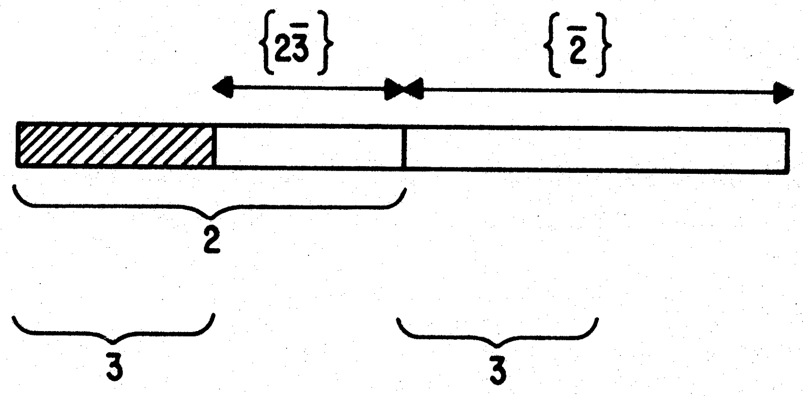

Fig. 3. An Intermediate Veitch Diagram for the Network of Fig. 1.

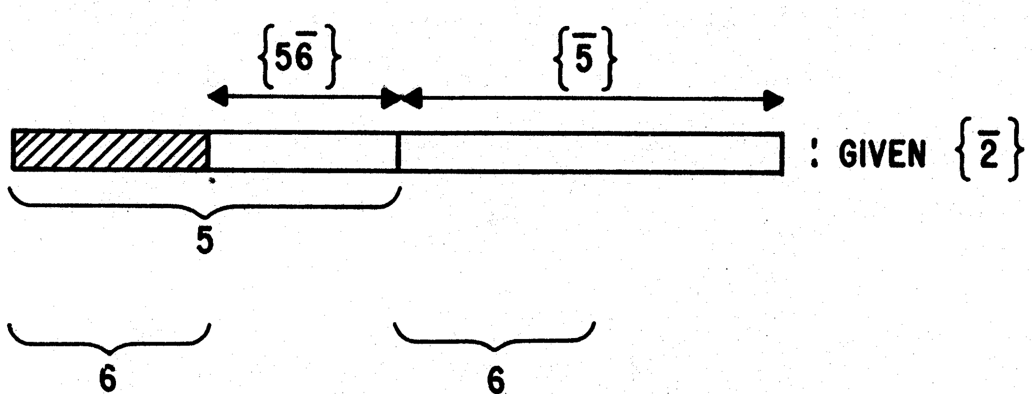

Fig. 4. A second Intermediate Veitch Diagram for the Network of Fig. 1.

At this point we have (equation 5) a lower bound for P_{1-7} as well as four "areas" of the complete Veitch diagram yet to be explored. These "areas" are:

$$\{\ \overline{2}\ \overline{5}\ \}\ ,\ \{\ \overline{2}\ 5\ \overline{6}\ \}\ ,\ \{\ 2\ \overline{3}\ \overline{5}\ \}\ ,\ \{\ 2\ \overline{3}\ 5\ \overline{6}\ \}$$

Continuing as before, the first of these, $\{\ \overline{2}\ \overline{5}\ \}$, when explored by PATH results in the discovery that no path exists, hence, we have

$$Q_{1-7} \geq q_2 q_3 \tag{6}$$

The second "area", $\{\ \overline{2}\ 5\ \overline{6}\ \}$, results in the event $\{\ 3\ 4\ 5\ \}$. Hence,

$$P_{1-7} \geq p_2 p_3 + q_2 p_5 p_6 + p_2 q_3 p_5 p_6 + q_2 p_3 p_4 p_5 q_6 \tag{7}$$

and the still remaining "areas" to be explored in this intermediate Veitch diagram are:

$$\{\ \overline{2}\ \overline{3}\ 5\ \overline{6}\ \}\ ,\ \{\ \overline{2}\ 3\ \overline{4}\ 5\ \overline{6}\ \}$$

Exploring $\{\ 2\ \overline{3}\ \overline{5}\ \}$ results in the event $\{\ 2\ 4\ 6\ \}$.

Hence:

$$P_{1-7} \geq p_2 p_3 + q_2 p_5 p_6 + p_2 q_3 p_5 p_6 + q_2 p_3 p_4 p_5 p_6 + p_2 q_3 p_4 q_5 p_6 \tag{8}$$

and the remaining "areas" to be explored in this intermediate Veitch diagram are:

$$\{\ 2\ \overline{3}\ \overline{4}\ \overline{5}\ \}\ ,\ \{\ 2\ \overline{3}\ 4\ \overline{5}\ \overline{6}\ \}$$

Exploring the area $\{\ 2\ \overline{3}\ 5\ \overline{6}\ \}$ results in the discovery that no path exists.

Therefore,

$$Q_{1-7} \geq q_2 q_5 + p_2 q_3 p_5 q_6 \tag{9}$$

Summarizing the remaining areas to be explored, they are:

$$\{\ \overline{2}\ \overline{3}\ 5\ \overline{6}\ \}\ ,\ \{\ \overline{2}\ 3\ \overline{4}\ 5\ \overline{6}\ \}\ ,\ \{2\ \overline{3}\ \overline{4}\ \overline{5}\ \}\ ,\ \{\ 2\ \overline{3}\ 4\ \overline{5}\ \overline{6}\ \}$$

Exploring each of the above areas results in the discovery that no path exists in each of them.

Therefore:

$$Q_{1-7} \geq q_2 q_5 + p_2 q_3 p_5 p_6 + q_2 q_3 p_5 q_6 + q_2 p_3 q_4 p_5 q_6 + p_2 q_3 q_4 q_5 + p_2 q_3 p_4 q_5 q_6 \tag{10}$$

Since all labels have been explored without generating any new labels, it follows that the entire Veitch diagram has been completely explored. Therefore, equations 8 and 10 are both exact equalities.

Two observations must be made:

a) Once a favorable event was found in a network (or sub-network) the remaining "areas" on the intermediate Veitch diagram for that network (or sub-network) was described in such a manner as to guarantee disjointness from the favorable event and from each other. It follows that any subsequently discovered favorable event in a remaining area is necessarily disjoint from any prior favorable event since the area in which it was found is caused to be a part of the event description.

b) The algorithm may be terminated at any point in its operation and still yield realistic upper and lower bounds for the terminal reliability since both P and Q are found simultaneously.

What remains is to formalize the procedure exposed in the preceding example in terms of a workable algorithm. In so doing we note that the PATH algorithm was applied to many "sub-networks" in an attempt to find a favorable event in each. Each sub-network was generated from a preceding favorable event which occurred in a sub-network disjoint from all other sub-networks. At this point we need to introduce some notational devices.

Let:

$\underset{\sim}{D}_i$: i^{th} disjoint sub-network; (11)

an N vector containing a description of the state of each interesting node (weighted nodes, as well as source and terminal nodes) in the network. The description of each node will be coded as follows:

0_n : The state of the n^{th} node is irrelevant (12)

$+1_n$: The n^{th} node exists (13)

-1_n : The n^{th} node does not exist (14)

For example, the first disjoint sub-network is $\underset{\sim}{D}_1$, and N vector described by:

$$\underset{\sim}{D}_1 = \{ 0_1 \quad 0_2 \quad 0_3 \quad \cdots \quad 0_N \} \tag{15}$$

Now let:

$\underset{\sim}{L}$: The short path description (if one exists) in $\underset{\sim}{D}_i$. This will be an N vector containing a description of the state of each interesting node in the network in order for a path to exist. The description of each node is coded as follows: (16)

0_n : The state of the n^{th} node is irrelevant (17)

1_n : The n^{th} node exists (18)

$\underset{\sim}{M}$: The N vector describing the nodes appearing in $\underset{\sim}{L}$ with a "1" specification and in $\underset{\sim}{D}_i$ with a "0" specification. The node description is coded as: (19)

1_n : n^{th} node satisfies (19) (20)

0_n : n^{th} node does not satisfy (19) (21)

Now suppose that no path exists in $\underset{\sim}{D}_i$.

Then let:

$\underset{\sim}{C} = \underset{\sim}{D}_i$: The description of a disjoint sub-network in which no path exists. An N vector containing a description of each interesting node in the network coded as in (12), (13), and (14). (22)

On the other hand, suppose that a path does exist in $\underset{\sim}{D}_i$. Furthermore, we have $\underset{\sim}{M}$.

Let:

m_n be the n^{th} element of $\underset{\sim}{M}$

then let:

$$S = \sum_{n=1}^{N} m_n \qquad (23)$$

Now, if:

S = 0, then let:

$\underset{\sim}{F} = \underset{\sim}{D}_i$: the description of a disjoint sub-network which explicitly contains a path. An N vector containing a description of each interesting node in the network coded as in (12), (13), and (14). (24)

Or if:

S > 0, then let:

$$\underset{\sim}{F} = \underset{\sim}{D}_i + \underset{\sim}{M} \qquad (25)$$

The + symbol when used with an equality means the following:

Let $\underset{\sim}{Z}$ represent an ordered set of elements, z_j, then if:

$$\underset{\sim}{Z} = \underset{\sim}{X} + \underset{\sim}{Y} \qquad (26)$$

define:

$$z_j = x_j + y_j \qquad (27)$$

It would be well to observe at this point that $\underset{\sim}{C}$ and $\underset{\sim}{F}$ correspond to labels (unfavorable event and favorable event, respectively) on the complete Veitch diagram. Now in the case of (25), we observe that the label $\underset{\sim}{M}$ does not cover the entire intermediate Veitch diagram, i.e., the Veitch diagram of the network given the occurrence of $\underset{\sim}{D}_i$. We further observe that $\underset{\sim}{M}$ always specifies nodes as in (19) and (20). Hence that region of the intermediate Veitch diagram (given $\underset{\sim}{D}_i$) not covered by the label $\underset{\sim}{M}$ is covered by the label:

$$\overline{\underset{\sim}{M}} \Rightarrow \underset{\sim}{R}_1 + \underset{\sim}{R}_2 + \cdots + \underset{\sim}{R}_S \tag{28}$$

For example, suppose that the N vector $\underset{\sim}{M}$ is:

$$\underset{\sim}{M} = [0_1 \quad 0_2 \quad 1_3 \quad 0_4 \quad 1_5 \quad 1_6 \quad 0_7], \text{ then}$$

$$\overline{\underset{\sim}{M}} \Rightarrow \underset{\sim}{R}_1 + \underset{\sim}{R}_2 + \underset{\sim}{R}_3 \quad \text{where}$$

$$\underset{\sim}{R}_1 = [0_1 \quad 0_2 \quad -1_3 \quad 0_4 \quad 0_5 \quad 0_6 \quad 0_7]$$

$$\underset{\sim}{R}_2 = [0_1 \quad 0_2 \quad 1_3 \quad 0_4 \quad -1_5 \quad 0_6 \quad 0_7]$$

$$\underset{\sim}{R}_3 = [0_1 \quad 0_2 \quad 1_3 \quad 0_4 \quad 1_5 \quad -1_6 \quad 0_7]$$

If we write the N vector $\underset{\sim}{M}$ in terms of only those nodes which have a "1" code in their location as an event, for our previous example:

$$\underset{\sim}{M} = \{\, 3\ 5\ 6 \,\}, \text{ and}$$

$$\overline{\underset{\sim}{M}} \Rightarrow \{\, \overline{3} \,\} + \{\, 3\ \overline{5} \,\} + \{\, 3\ 5\ \overline{6} \,\}$$

In general, if $\underset{\sim}{M} = \{\, 1\ 2\ 3\ 4 \cdots \}$, $\overline{\underset{\sim}{M}} \Rightarrow \{\, \overline{1} \,\} + \{\, 1\ \overline{2} \}$

$$+ \{\, 1\ 2\ \overline{3} \,\} + \{\, 1\ 2\ 3\ \overline{4} \,\} + \cdots \tag{29}$$

Using the preceding then, that region of the intermediate Veitch diagram not covered by the label $\underset{\sim}{M}$ is covered by:

$$\underset{\sim}{T}_{k+1} = \underset{\sim}{D}_i + \underset{\sim}{R}_1 \tag{30}$$

$$\underset{\sim}{T}_{k+2} = \underset{\sim}{D}_i + \underset{\sim}{R}_2 \tag{31}$$

$$\underset{\sim}{T}_{k+S} = \underset{\sim}{D}_i + \underset{\sim}{R}_S \tag{32}$$

where k is the number of previously found $\underset{\sim}{T}$ vectors in the same major cycle of the algorithm to be presented.

III. THE ALGORITHM

The notation presented in Section II now allows a compact presentation of the algorithm shown in Figure 5. One new sub-routine appears in the figure. NETMAT is an algorithm which operates on the original connectivity matrix of the graph, according to $\underset{\sim}{W}$, to generate a connectivity matrix in which the row(s) and column(s), corresponding to nodes in $\underset{\sim}{W}$ which have a -1 code, are zeroed out. This is equivalent to removing those nodes from the graph. The resulting degraded connectivity matrix is passed to PATH to determine the shortest path through the resulting graph if one exists.

The algorithm, as presented in the Figure 5, will terminate only when the terminal reliability (and unreliability) expression is complete. There are several options for terminating the algorithm with an approximate solution. One such option is to add a set of instructions such that termination occurs after a certain number of $\underset{\sim}{F}$ vectors have been accumulated. Another option would be to terminate after a certain number of $\underset{\sim}{C}$ vectors have been accumulated. Still another option is to skip to the β loop, in Figure 5, from the α loop if a specified upper bound on the number of $\underset{\sim}{T}$ vectors is about to be exceeded. Of course, any combination of the options can be exercised quite easily.

In any event, the algorithm is designed so that it provides both a terminal reliability and terminal unreliability expression in symbolic form. The sum of the implied probabilities of all $\underset{\sim}{F}$ vectors can subsequently be numerically evaluated for the terminal reliability as can the sum of the implied probabilities of all $\underset{\sim}{C}$ vectors for the terminal unreliability. Thus, if the algorithm is not allowed to terminate with a complete solution, it provides both upper and lower bounds which are realistic.

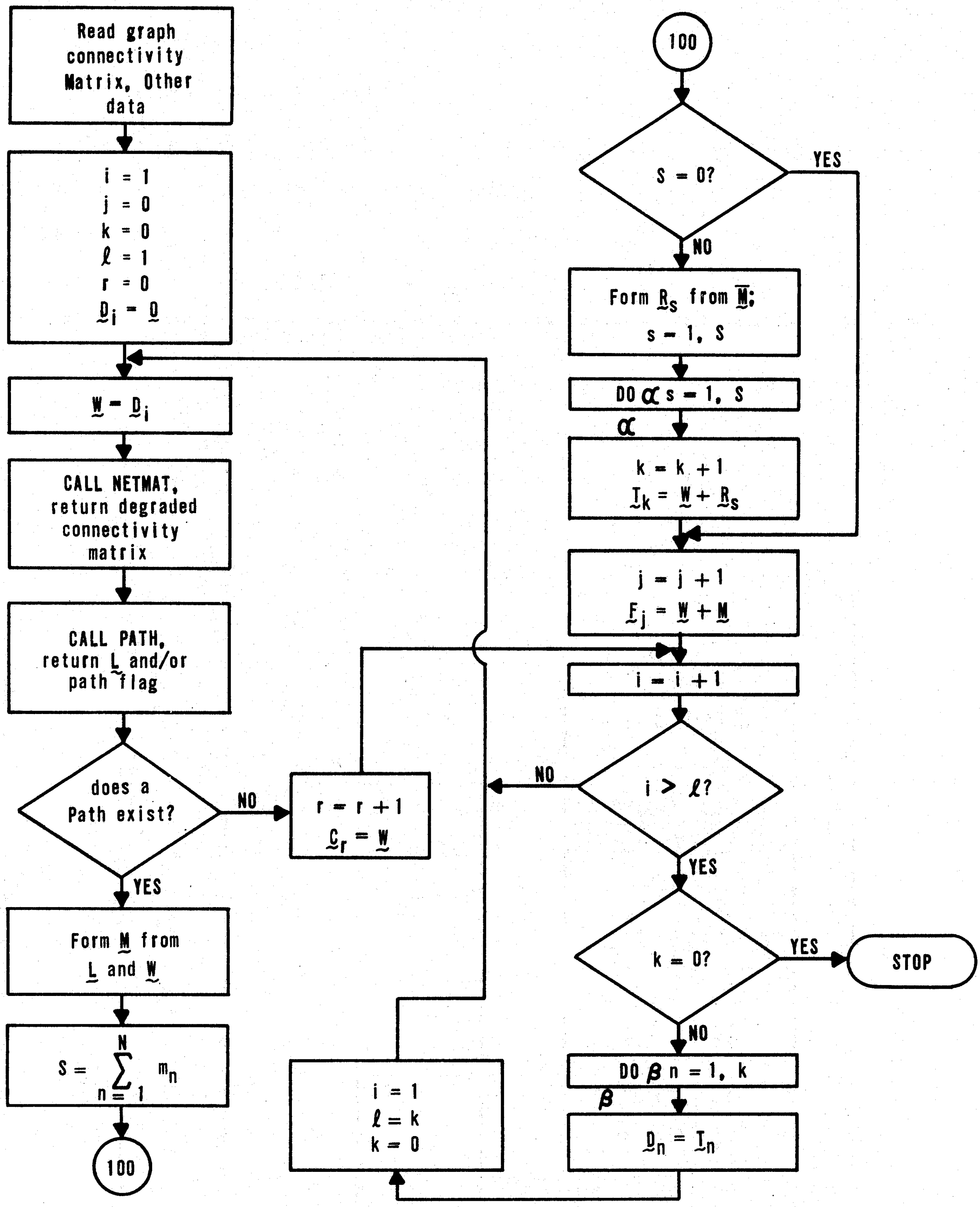

Fig. 5. Flow Chart for the Algorithm

Some remarks concerning duality of the algorithm are in order at this point. In one sense the algorithm is its own dual since it simultaneously discovers both simple paths and cutsets in the network. However, these are discovered directly in terms of their disjoint implicants rather than their prime sets which must be operated on to form the disjoint implicants. Therefore, the application of other algorithms to the sets $\{C_i\}$ and $\{F_i\}$ would be required to determine the prime set of simple paths and the prime set of cuts. In the sense of the basic operating mode of the algorithm, however, its dual certainly is not embodied in itself. Recall that the algorithm operates by finding a single short path through a degraded network. Its dual then would operate by finding a single minimal cut in a degraded network. The dual algorithm has not been written as of the submission date of this paper but it would only involve a straightforward extension of the same concepts as exposed herein.

The usefulness of the dual algorithm becomes apparent in considering the problem of finding the terminal reliability between arbitrary source and terminal nodes in a network. This problem is easily solved for the case of a complete network since the topology is completely symmetric. Hence the solution for any source-terminal pair is the solution for all source-terminal pairs. The complete network can then be made identical to any network of the same or lower order by removing appropriate nodes. (The order of a network, as used here, is the number of unweighted nodes, not counting source and terminal.) Likewise, the solution for the complete network can be used to find the solution to the lower order network by setting $p = 0$ for those nodes removed. The thrust of the argument is simply that in completely connected networks which are too large to be solved exactly, the upper bound solution can be found more efficiently by the dual algorithm. Combining this with the lower bound solution from the algorithm presented here, very tight bounds could be found efficiently even for networks with node probabilities ~ 0.1.

IV. RESULTS

In this section we indicate the input-output operations of the algorithm through a number of examples. In figure 6 the graph of a completely connected network of order 3 is shown. The algorithm requires the following inputs:

a. Source node label.

b. Terminal node label.

c. Total number of nodes. This is the number of weighted nodes +2 (source and terminal nodes).

d. Maximum allowable number of $\underset{\sim}{T}$ vectors in memory at any one time (see page 13). This is to avoid memory overflow for very large networks.

e. Maximum allowable number of $\underset{\sim}{F}$ vectors (see page 13).

f. Maximum allowable number of $\underset{\sim}{C}$ vectors (see page 13).

g. The connectivity matrix for the network. The matrix for Figure 6 is shown in Table II.

The terminal reliability expression for the network of Figure 6 is shown in Table III. The terminal unreliability expression is shown in Table IV.

A complete network of order 4 is shown in Figure 7. The terminal reliability expression is given in Table V, the terminal unreliability expression in Table VI.

The foregoing examples should give an adequate feel for the manner in which the algorithm operates. Accordingly, in our final Tables VII and VIII we give summary results for the indicated problems. The problem entries in

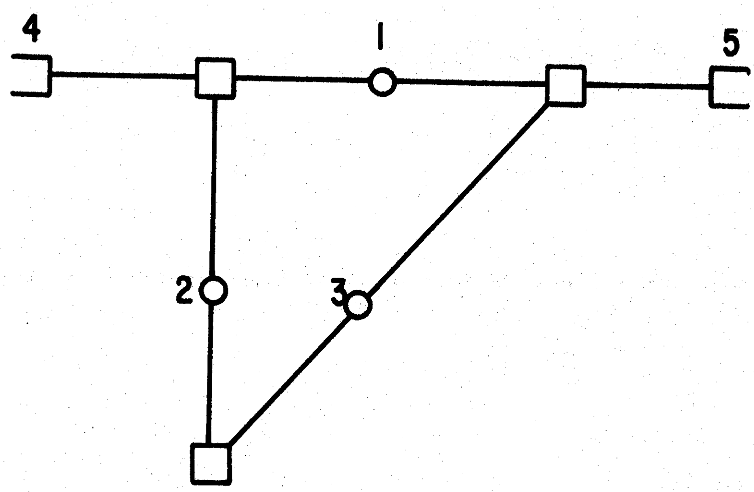

Fig. 6. A Completely Connected Network of Order 3.

Table II. Connectivity Matrix for the Network of Fig. 6.

NODES	1	2	3	4	5
1	0	1	1	1	1
2	1	0	1	1	0
3	1	1	0	0	1
4	1	1	0	0	0
5	1	0	1	0	0

TABLE III

TERMINAL RELIABILITY EXPRESSION FOR THE

NETWORK OF FIGURE 6

NODES				
1	2	3	4	5
1	0	0	0	0
-1	1	1	0	0

Reads: $P_{4-5} = p_1 + q_1p_2p_3$

TABLE IV

TERMINAL UNRELIABILITY EXPRESSION FOR

THE NETWORK OF FIGURE 6

NODES				
1	2	3	4	5
-1	-1	0	0	0
-1	1	-1	0	0

Reads: $Q_{4-5} = q_1q_2 + q_1p_2q_3$

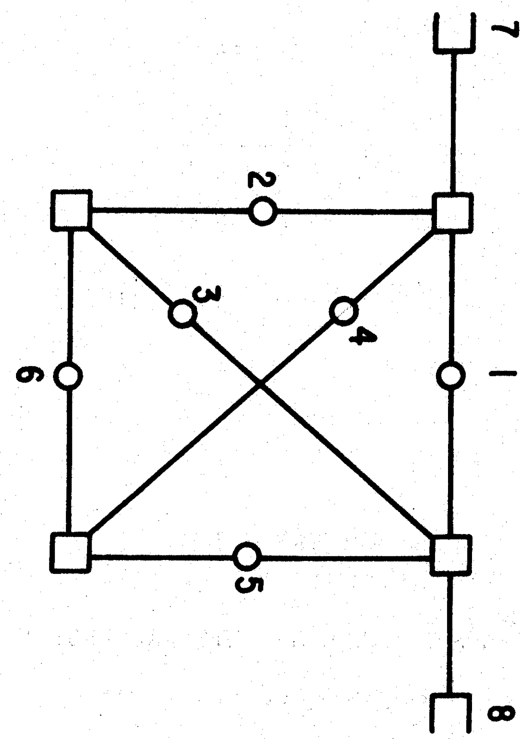

Fig. 7. A Completely Connected Network of Order 4.

TABLE V

TERMINAL RELIABILITY EXPRESSION FOR THE

NETWORK OF FIGURE 7

NODES

1	2	3	4	5	6	7	8
1	0	0	0	0	0	0	0
-1	1	1	0	0	0	0	0
-1	-1	0	1	1	0	0	0
-1	1	-1	1	1	0	0	0
-1	-1	1	1	-1	1	0	0
-1	1	-1	-1	1	1	0	0

Reads: $P_{7-8} = p_1 + q_1p_2p_3 + q_1q_2p_4p_5 + q_1p_2q_3p_4p_5$

$+ q_1q_2p_3p_4q_5p_6 + q_1p_2q_3q_4p_5p_6$

TABLE VI

TERMINAL UNRELIABILITY EXPRESSION FOR THE NETWORK OF FIGURE 7

NODES

1	2	3	4	5	6	7	8
-1	-1	0	-1	0	0	0	0
-1	1	-1	1	-1	0	0	0
-1	-1	-1	1	-1	0	0	0
-1	-1	1	1	-1	-1	0	0
-1	1	-1	-1	-1	0	0	0
-1	1	-1	-1	1	-1	0	0

Reads: $Q_{7-8} = q_1q_2q_4 + q_1p_2q_3p_4q_5 + q_1q_2q_3p_4q_5$

$+ q_1q_2p_3p_4q_5q_6 + q_1p_2q_3q_4q_5$

$+ q_1p_2q_3q_4p_5q_6$

TABLE VII

SUMMARY RESULTS

PROBLEM/ NR. OF WEIGHTED NODES	NR. OF F VECTORS	NR. OF C VECTORS	EXACT/ BOUNDED SOLUTION	SOLUTION TIME (sec)	IF NODE PROBABILITY IS:	THEN TERM. RELIABILITY IS: UPPER/LOWER
Figure 6/3	2	2	Exact	1	0.9	.981000/ .981000
Figure 1/5	5	6	Exact	1	0.9	.978480/ .978480
Figure 7/6	6	6	Exact	1	0.9	.997848/ .997848
Figure 8/10	29	26	Exact	1	0.9	.999795/ .999795
*Figure 10/12	79	148	Exact	5	0.9	.975116/ .975116
*Figure 11/21	924	1821	Exact	36	0.9	.994076/ .994076
**Figure 9/26	100	1	Bounded	17	0.9	.999900/ .997669
	200	3	Bounded	32	0.9	.999857/ .999051
	300	6	Bounded	48	0.9	.999844/ .999515
	8192	2752	Bounded	1008	0.9	.999797/ .999797

*For the purpose of comparison with reference [1], the problems for Fig. 10 and 11 were also solved for the case of node (link) probability = 0.9 applying before series parallel reduction as in the reference. In this case, for Fig. 10, terminal reliability = .912914; for Fig. 11, terminal reliability = .997186.

**Terminal nodes for the solutions pertinent to Fig. 9 are 1 and 28.

TABLE VIII SUMMARY RESULTS

PROBLEM/ NR. OF WEIGHTED NODES	NR. OF F VECTORS	NR. OF C VECTORS	EXACT/ BOUNDED SOLUTION	SOLUTION TIME (sec.)	IF NODE PROBABILITY IS:	THEN TERM. RELIABILITY IS: UPPER/LOWER
Figure 9/26	300	6	Bounded	51	0.1	.343900/ .005590
					0.2	.590396/ .050194
					0.3	.759851/ .168231
					0.4	.870162/ .361452
					0.5	.936890/ .591339
					0.6	.973454/ .797225
					0.7	.991013/ .931371
					0.8	.997971/ .987969
					0.9	.999844/ .999515
Figure 9/26	8192	2752	Bounded	1115	0.1	.163956/ .008827
					0.2	.379602/ .086618
					0.3	.580863/ .280085
					0.4	.743828/ .542600
					0.5	.863822/ .775591
					0.6	.941422/ .918168
					0.7	.981696/ .978720
					0.8	.996582/ .996472
					0.9	.999797/ .999797

Table VIII involved solving for bounded terminal reliability and terminal unreliability expressions for the network of Figure 9, then performing a numerical evaluation of these expressions for the indicated node probabilities. We note a time increase of approximately three seconds and 107 seconds for the additional numerical evaluations. The increase is probably not as much as indicated as all times presented in Tables VII and VIII are taken from the day file and do not represent a strict accounting of CP time devoted to the problems presented.

One other point concerning the problem entries in Table VIII is worth noting. Namely, the upper bounds on terminal reliability are quite poor at low values of node probability. This is one more indication of the potential value of the dual to the algorithm presented here. Such a dual would provide a tighter upper bound to terminal reliability.

V. CONCLUSIONS

In a previous publication [1] an efficient algorithm for the analysis of unreliable communications networks was proposed. The paper then went on to apply the proposed algorithm to the problems shown in Figures 10 and 11. For the problem of Figure 10, the algorithm ran to completion in 112 seconds on an IBM 360/67 computer. For the problem of Figure 11, the algorithm terminated with an approximate solution (lower bound) after 10 minutes of computation time.

This paper reports on an even more efficient algorithm for the analysis of unreliable communications networks. Specifically, the problem of Figure 10 was completely solved in slightly less than 5 seconds on a CDC 6600 computer. The problem of Figure 11 was completely solved in approximately 36 seconds. For CP bound problems the CDC 6600 is from 2 to 5 times faster than the IBM 360/67. We, therefore, estimate the relative efficiency of the algorithm herein to that of reference [1] to be in the range 4 to 11. There is one further advantage to

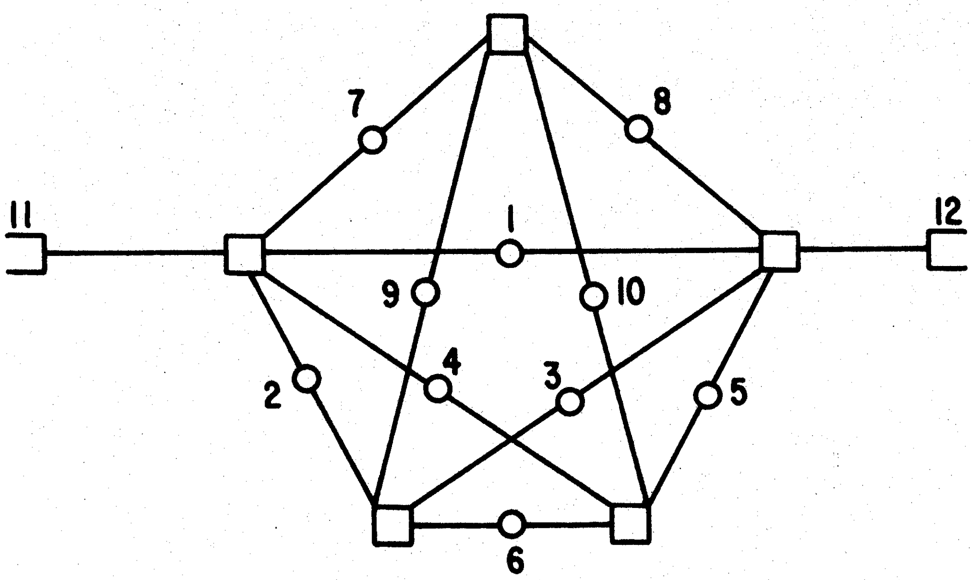

Fig. 8. A Completely Connected Network of Order 5.

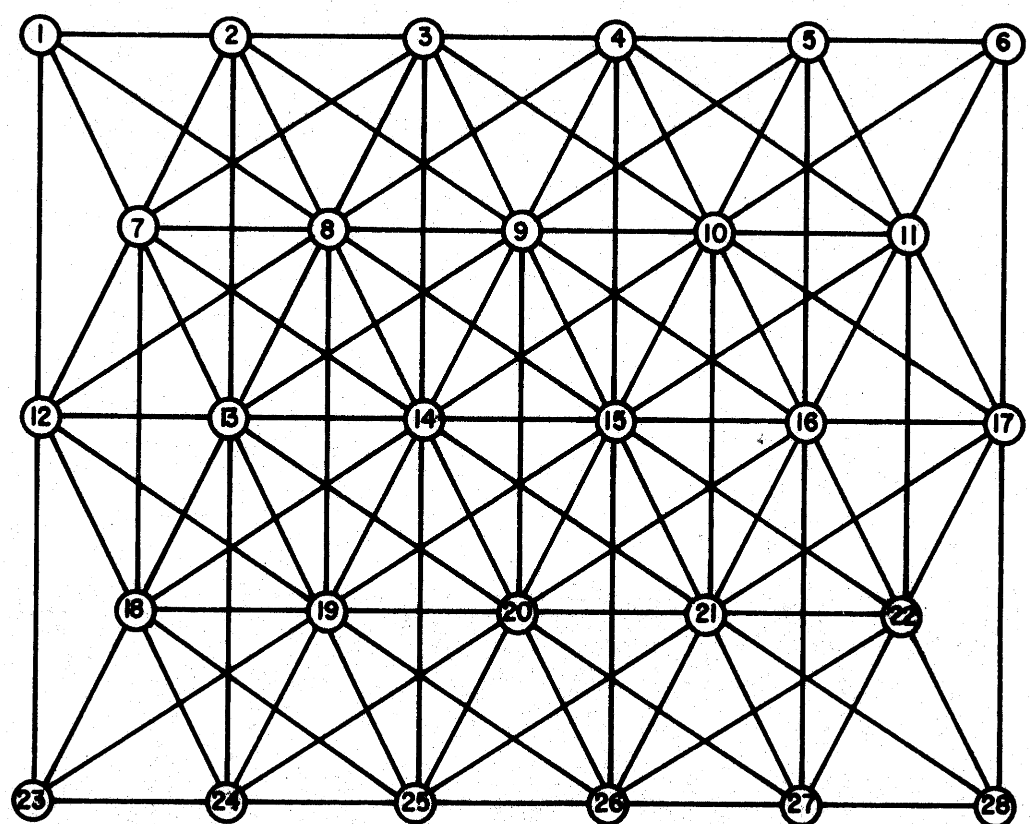

Fig. 9. A Part of the AUTOVON Network.

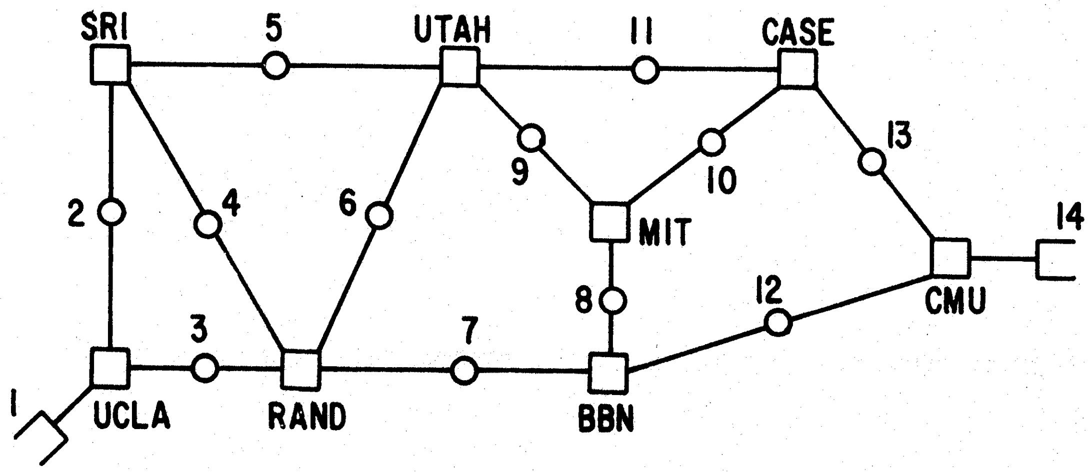

Fig. 10. ARPA Computer Network after
Series - Parallel Reduction

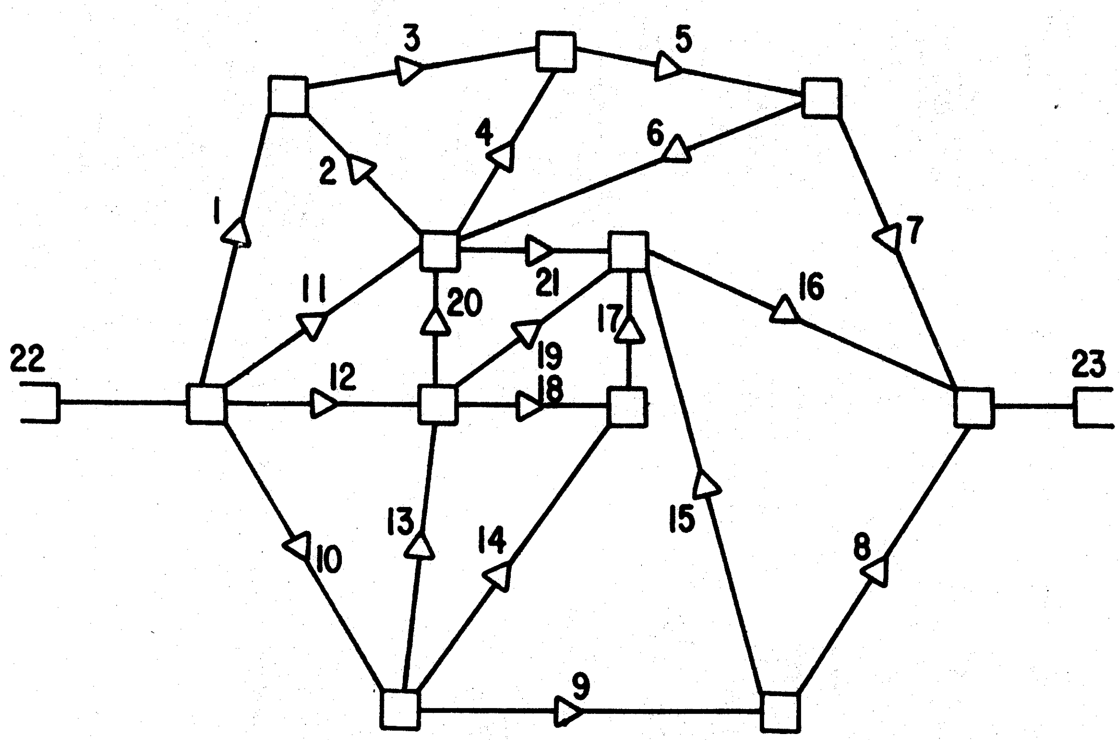

Fig. 11. A More Connected Network

the algorithm proposed herein. Our algorithm simultaneously solves for the probability of success and the probability of failure thereby yielding realistic and very tight bounds (both upper and lower) on extremely complicated networks in which it is not practical to force an exact solution.

BIBLIOGRAPHY

1. Luigi Fratta and Ugo G. Montanari, "A Boolean Algebra Method for Computing the Terminal Reliability in a Communication Network," IEEE Transactions on Circuit Theory, Vol CT-20, No. 3, pp 203-211, May 1973

2. Montgomery Phister, Jr., Logical Design of Digital Computers, Ch. 4, "The Simplification of Boolean Functions and the Veitch Diagram Simplification Method," John Wiley & Sons, Inc., 1961