AD-A039 455 NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF F/G 9/2
SOFTWARE ERROR DETECTION MODELS, VALIDATION TESTS AND PROGRAM C--ETC(U)
NOV 76 N F SCHNEIDEWIND, G T HOWARD

UNCLASSIFIED NPS-52SS76111 NL

1 OF 2
AD A039455



AD A039455

(2)
NW

NPS52SS76111

NAVAL POSTGRADUATE SCHOOL

Monterey, California



SOFTWARE ERROR DETECTION
MODELS, VALIDATION TESTS
AND PROGRAM COMPLEXITY MEASURES

by

N. F. Schneidewind G. T. Howard
M. Kirchgaessner

November 1976

Approved for public release; distribution unlimited

Prepared for:
Naval Air Development Center
Warminster, Pennsylvania

AD No. 1
DDC FILE COPY

DDC
RECEIVED
MAY 13 1977
B

NAVAL POSTGRADUATE SCHOOL
Monterey, California

Rear Admiral Isham Linder
Superintendent

Jack R. Brosting
Provost

The work reported herein was supported by the Naval Air Development Center, Warminster, Pennsylvania.

Reproduction of all or part of this report is authorized.

This report was prepared by:

N. F. Schneidewind
N. F. SCHNEIDEWIND
Professor of Information Science

G. T. Howard
G. T. HOWARD
Associate Professor of
Operations Research

Reviewed by:

D. A. Schrady
D. A. SCHRADY, Dean
Division of Information and
Policy Sciences

Released by:

Robert Fossum
ROBERT FOSSUM
Dean of Research

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: NPS-52SS76111	2. GOVT ACCESSION NO. / 3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): Software Error Detection Models, Validation Tests and Program Complexity Measures.	5. TYPE OF REPORT & PERIOD COVERED: Technical Report.
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): N. F. Schneidewind and G. T. Howard M. Kirchgaessner	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Naval Postgraduate School, Monterey, California 93940	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: N66269/76/RQ/02030
11. CONTROLLING OFFICE NAME AND ADDRESS: Naval Air Development Center, Warminster, Pennsylvania 18974	12. REPORT DATE: November 1976 13. NUMBER OF PAGES: 154
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): 159p.	15. SECURITY CLASS. (of this report): Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for putlic 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)

Software measurement
Program Structure
Software error

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report describes a continuing research effort in software reliability which was first reported in "System Test Methodology," Naval Postgraduate School, Vol I NPS55SS75072A, Vol. II NPS 55SS75072B(1975). The work just completed involved: improvement of the software error simulation model; validation of the software error simulation model; and analysis of program complexity with simulation and analytical models, using 44 Naval Tactical Data System procedures. The results which were achieved are the following:

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)

251 450

4B

(1) all validation tests were passed; however simulation results were generally higher than analytical results and (2) the general direction of the relationship between complexity measures and error detection was as expected; however, considerable variability was exhibited when single independent variables were used. It appeared that a multivariable model involving error detection and several program complexity measures would be more appropriate.

TABLE OF CONTENTS

INTRODUCTION	I-1
DESCRIPTION OF ERROR SIMULATION PROGRAM	II-1
VALIDATION TESTS	III-1
ANALYSIS OF COMPLEXITY MEASURES	IV-1
SUMMARY	V-1
REFERENCES	R-1
APPENDIX A: FLOW CHART OF ERROR SIMULATION PROGRAM (MAIN), AND SUBROUTINE (SEED)	A-1
APPENDIX B: DIRECTED GRAPHS, SIMULATION AND ANALYTICAL RESULTS	B-1

ACCESSION for
NTIS White Section
DDC Buff Section
UNANNOUNCED
JUSTIFICATION
BY
DISTRIBUTION/AVAILABILITY CODES
Dist. AVAIL. and/or SPECIAL
A

I. INTRODUCTION

This report describes and documents the research conducted at the Naval Postgraduate School (NPS) during FY 76-77 on the System Test Methodology project sponsored by the Naval Air Development Center (NADC). The project began in FY 75-76. Previous reports were "System Test Methodology," Vol. I, NPS55Ss75072A and Vol. II, NPS55Ss75072B, July 1975, published by the Naval Postgraduate School [1]. Several publications in the open literature have resulted from this research [2, 3, 4]. This work covered three areas:

a. Software error simulation and analytic models.

b. System test simulation model.

c. Partitioned tests for software error analysis.

Computer programs were developed and provided to NADC for the simulation and analytic models of a. Of the three areas, it was felt a. had the greatest potential for improving software reliability. Consequently, the following efforts were undertaken in FY 76-77:

- Improvement of the software error simulation model.
- Validation of the software error simulation model.
- Analysis of program complexity measures.

These topics are covered in Sections II, III, and IV, respectively. A brief overview of each topic, in the order listed above, will be given here.

- Sections of the simulation program which involved execution and repair times were removed because: (1) current research interests are structural characteristics and error detection properties of programs, rather than timing aspects and (2) CPU time required by these parts of the program were too high in relation to the need for this information. The new error insertion feature (simulates the

possibility of new error insertion as a result of error correction) was removed because this process is not well understood. Therefore, it was difficult to prescribe the appropriate probability distributions and conditions which should govern error insertion in a simulation. User instructions are provided in Section II. A program listing of the simulation model is included with this report.

- Validation tests of the simulation model were performed with respect to: (1) mean number of errors seeded, (2) probability of arc traversal and (3) number of arc and path traversals. Validation tests were conducted with respect to the analytical model. We did not conduct validation tests of either the analytical or simulation model with respect to the error detection process in real programs, although Naval Tactical Data System program structures and error parameters were used in some of our studies. Thus, validation tests were limited to a test of the correctness of our concept of error detection. The analytical model was used as the standard for comparing simulation results. Independent appraisals by NADC and NPS personnel have concluded that the analytical model's equations accurately represent the error detection process as originally conceived. Validation with respect to actual program error detection processes will be attempted in future research.
- The third area involved using the models to study the relationship between program structure and error detection properties. Naval Tactical Data System programs were used for this investigation.

II. DESCRIPTION OF ERROR SIMULATION PROGRAM

Summary

This program is designed to simulate the error detection process in a portion of computer software, represented here as a directed graph. The graph consists of nodes representing merging or branching points in the software and arcs representing the sequences of instructions between the nodes. There are no real limitations on the topology of the graphs which can be handled by the program except that multiple arcs between the same two nodes are not permitted. If such arcs are desired, they can be artifically handled by introducing additional dummy nodes on each of the multiple arcs. The program is currently dimensioned for thirty nodes. This could easily be changed although some formats would also have to be modified.

In the specification of the graph, the user can input the length (number of instructions) of each arc or allow them all to be set internally to the same length of 10.

The locations of the errors in the graph can also be specified by the user, or the program can place the errors by a random process. When this is done random numbers representing the distance between errors are drawn from the exponential distribution. The errors are then placed in the arcs corresponding to these randomly selected instructions.

The MAIN program simulates the process of running an input through the directed graph. Each input finds all errors on its path in a single pass. The single input is terminated when it reaches a node from which no arcs emanate.

The selection of which arc to follow from a given node is made randomly (uniform) using IZ as the random number seed. The seed is used to generate a new random number which serves as the next seed. The seed IZ is not reset to its initial value when the graph is re-seeded with errors. Thus the random process IZ continues to run sequentially until the run ends.

For the graph which is input, the data specifies (among other items)

a) the number of inputs per replication

b) the number of replications per seeding

c) the number of seedings

The main elements of the model are discussed in more detail in the following sections.

1. error seeding
2. branching
3. error finding
4. data input
5. computations performed

1. Error Seeding

The user has the option of planting errors in the arcs of the directed graph or he can allow the program to seed the errors randomly. If the user chooses to let the program place the errors, he must specify the parameter MEANER as part of the input data. This parameter is used by the program as the mean of the distribution which determines the distance between successive errors. It is conveniently interpreted as the mean number of instructions between errors.

Since the arc lengths and distances between errors are treated internally as floating point numbers, it is not necessary that arc lengths be integer, although that is the proper interpretation when the length is measured by the number of instructions. It may be desirable to measure length in some other way related to the complexity of the instructions and the program permits this.

Unless the user specifies the individual arc lengths X(K,J), they will all be set internally to 10 (line 95). The random selection of arc lengths is not a feature of this program but a simple modification would permit this, if desired. If so, a random number seed (say IY) would have to be specified above line 95. To avoid correlation with the other random processes in the program the symbols IX, IW, and IZ should not be used as the seed.

The random seeding of errors is done in the subroutine SEED. The arcs of the graph is assumed to be arranged in "natural order" so that arc (i,j) precedes (k,l) if $i < k$, and arc (i,j) precedes (i,l) if $j < l$. Each arc (i,j) has a specified length X(I,J) and

it is convenient to think of these lengths as being laid out sequentially on the real line beginning at the origin.

Using the seed IX, a random number ER1 is drawn from the exponential distribution with mean 1 and rescaled by multiplying it (in floating point) by the parameter MEANER to yield XER1. Thus the quantity XER1 is a sample from the exponential distribution whose mean is MEANER. The quantity XER1 is used as the distance from the last seeded error to the next. A comparison is made to ascertain in which arc the error should be placed and the process is repeated with a new value of XER1 until the location of the "next" error falls beyond the total length of all arcs combined. At this point the seeding process is complete and the SEED subroutine returns control to the main program.

The variables set in SEED and placed in common with the MAIN program include

ISEED(I,J) = The number of errors seeded in the arc ij for every arc

SVSEED(I,J) = A saved copy of ISEED(I,J)

NINST = total number of instructions

NSEED = total number of errors seeded.

When errors are planted by the user, the subroutine SEED is not called, and the variables NINST and NSEED are not computed.

2. Branching

This section discusses the branching mechanism which governs the path taken by an individual input traversing the directed graph.

Node one is assumed to be the input node to the directed graph and any node having no arcs leaving it is a terminal node. To preclude endless cycling, it is necessary that the graph have at least one terminal node which can be reached from node 1. Except for node 1 no special ordering of nodes is required. An arc can go from any node to any node. There is no special significance to an arc which begins and ends on the same node. Such self-loops may be used to model repetitive processes which are repeated a variable number of times. The number of repetitions is not controlled by the user but is determined by the branching process which randomly selects the next node in the sequence of nodes encountered by a particular input.

The branching process from a particular node is begun by counting the number of arcs NUMSUC which emanate from the current node designated as NODE. Then using the seed IZ, a random number U is selected from the uniform distribution on the interval 0 to 1. The quantity K = 1 + NUMSUC*U is computed. Note that K can assume any integer value from 1 to NUMSUC and that these are all equally likely.

It is convenient to think of the arcs emanating from NODE as being arranged in natural order as the first, second, third, ..., NUMSUC. Then the $K^{\underline{th}}$ arc emanating from NODE is selected as the node to which the input advances.

3. Error Finding

The number of errors placed in an arc ij is ISEED(I,J). When the traversal of a particular arc is simulated by the program, it is assumed that all errors in that arc are detected by that input and corrected before subsequent inputs are run. Thus after traversal of arc ij the variable ISEED(I,J) is set to 0. The variable SVSEED(I,J) is available for replacing the original errors when desired. The total number of errors detected by any input is cumulated and recorded as NFIND (line 166).

Other error finding mechanisms are possible and may prove realistic if sufficient data becomes available to reveal more about the error finding process. For example, it may be more realistic to assume that when an arc is traversed its errors are exposed to detection but may or may not actually be found depending on some random process. An assumption of this type could be accommodated in this model if some modifications are made.

4. Data Input

The required data is described below. Each card or group of cards is described in a separate section.

(a) MINPUT, NUMOUT, NREPET, MEANER, N. The format is (5I5)

MINPUT is the number of different inputs desired within each replication (dimensioned 50)

NUMOUT is the number of replications within each repetition

NREPET is the number of repetitions or re-seedings. With each reseeding NUMOUT replications are performed and in each of those there are MINPUT inputs.

MEANER is the mean distance between seeded errors

N is the number of nodes in the graph (currently dimensioned for 30)

(b) The graph structure is read in as described below. The number of cards is variable but can not exceed N + 1.

Each card has format of (16I5) although typically many fields will not be used.

The first field identifies a node from which arcs emanate.

The second field gives the number of arcs ($\leq$ 14).

The remaining 14 fields (if required) identify the nodes to which these arcs go.

The above information is repeated for each node from which arcs emanate.

This section of data is terminated by a card with 99 punched in columns 4 and 5.

(c) The following cards are optional. If used, they specify the arc lengths.

The format is 2I5, 7(I5,F5.0).

The first field identifies the from node.

The second field specifies the number of arcs.

The next fields are used in pairs and have the following meanings.

first field, identifies a to node.

second field, specifies the arc length.

This section is terminated with a 99 in cols 4 and 5. This card is not optional.

(d) If the user wishes to plant the errors in arcs of his choice instead of letting the program seed the errors randomly, the following cards are used.

The format is (16I5).

Field one specifies a from node

Field two specifies the number of arcs emanating from the node in which errors are to be planted.

The next fields are used in pairs.

- first in pair, identifies the to node
- second in pair, specifies the number of errors in the arc just defined

This section is terminated with 99 in cols 4 and 5. (not optional)

(e) The last card is used to specify an output option. The variable name is MOUT.

The format is (I5)

If a 0 is punched, only summary output is given.

If a 1 is punched, detailed information about the seeding, the paths, etc is given. This should be used only for small values of the product NUMOUT * NREPET.

If this product exceeds 25, the program sets MOUT=0 as protection against extensive output.

5. Computations Performed by the Program

a. As discussed in section A-1 the program simulates MINPUT inputs for each of NUMOUT replications and the entire process is repeated for NREPET seedings. This section describes the computation performed in producing the summary output.

Let i = 1,..., MINPUT

j = 1,..., NUMOUT

k = 1,..., NREPET.

x_{ijk} = number of errors found on input i, replication j of seeding k. For each seeding k a summary is produced giving the average number of errors found for each input and the standard deviation of the number of errors found on each input. These are defined as follows:

INPUT NUMBER i

AVE NUMBER ERRORS FOUND

$$x_{i\cdot k} = \sum_{j=1}^{NUMOUT} x_{ijk} / NUMOUT$$

STD DEV

$$\sigma_{i\cdot k} = \left[\sum_{j=1}^{NUMOUT} (x_{ijk} - x_{i\cdot k})^2 / (NUMOUT-1) \right]^{1/2}.$$

After all NREPET seedings have been analyzed a summary output is produced giving the average number of errors found for each input and the standard deviation of the number of errors found. These quantities are defined as:

SUMMARY FOR INPUT i

AVERAGE ERRORS FOUND

$$x_{i\cdot\cdot} = \sum_{j} \sum_{k} x_{ijk} / (NUMOUT)\cdot(NREPET)$$

STANDARD DEVIATION

$$\sigma_{i\cdot\cdot} = \left[\sum_j \sum_k (x_{ijk} - x_{i\cdot\cdot})^2 / ((\text{NUMOUT}) \cdot (\text{NREPET}) - 1) \right]^{1/2}.$$

b. To assist in relating these computations to the the program, the following section is included.

for each input i

$\text{IFOUND}(\ ) = \sum_j x_{ijk}$ for k given

$\text{SFOUND}(\ ) = \sum_j (x_{ijk})^2$ for k given

$\text{CUMSQR}(\ ) = \sum_j \sum_k (x_{ijk})^2$, the cumulative number of (squared) errors found for all replications and seedings.

$\text{SVAVE}(\ ) = \sum_k \sum_j x_{ijk}/\text{NUMOUT}$

$\text{TAVE} = \sum_j \sum_k x_{ijk}/(\text{NUMOUT}).(\text{NREPET}).$

The quantity $\sigma_{i\cdot\cdot}$, the standard deviation is computed using the following relationships:

$$\sigma_{i\cdot\cdot} = \left[\sum_j \sum_k (x_{ijk} - x_{i\cdot\cdot})^2 / ((\text{NUMOUT}) \cdot (\text{NREPET}) - 1) \right]^{1/2}$$

$$= \left[[\sum_j \sum_k x_{ijk}^2 - (\text{NUMOUT}) \cdot (\text{NREPET}) x_{i\cdot\cdot}^2] / ((\text{NUMOUT}) \cdot (\text{NREPET}) - 1) \right]^{1/2}$$

$$= \left[[\text{CUMSQR}(\ ) - (\text{NUMOUT})(\text{NREPET})(\text{TAVE})^2] / ((\text{NUMOUT})(\text{NREPET}) - 1) \right]^{1/2}.$$

III. VALIDATION TESTS

This section describes validation tests of the simulation model; the analytical model [1] was used as the standard. The test results are preceded by a brief description of the analytical model. Hypothesis tests were conducted of: (1) mean number of errors seeded, (2) probability of arc traversal and (3) numbers of arc and path traversals. Tests of mean number of detected errors were not performed because an efficient method of computing the variance from the analytical model was not available. The standard deviation of detected errors was needed to calculate confidence limits. Although the simulation model computes an estimate of the variance, this calculation could not be used because, without the analytical model variance, it could not be validated. However, a notable achievement has been the development of an algorithm for computing the variance of number of errors detected from the analytical model [5]. Work is proceeding on obtaining a solution in closed form. When a computer program is available for the closed form solution, validation tests will be conducted of simulation model mean number of detected errors and the variance.

Analytical Model

The analytical model [1] computes the exptected number of detected errors for each input. Designating the original expected number of errors in arc ij as μ_{ij}, for the arc between nodes i and j and p_{ij} as the probability of traversing arc ij one or more times, the expected number of detected errors in arc ij for the first input is $\mu_{ij}p_{ij}$. The expected number of errors remaining in arc ij after the first input is $\mu_{ij}(1-p_{ij})$. The expected number of errors detected on the second input is $\mu_{ij}(1-p_{ij})p_{ij}$. The expected number of errors detected on the $k^{\underline{th}}$ input is

$$e(k) = \mu_{ij}(1-p_{ij})^{k-1}p_{ij} \quad . \tag{1}$$

When the expected number of detected errors is added over all arcs, we have

$$E(k) = \sum_{ij} \mu_{ij}(1-p_{ij})^{k-1}p_{ij} \tag{2}$$

for the total expected number of detected errors on the $k^{\underline{th}}$ input. The initial μ_{ij} can be interpreted as the mean number of errors per arc which is originally present, in which case the $\mu_{ij} = \bar{\mu}_{ij}$ are equal for all arcs, or as a specified number of errors in each arc. In the latter case, the μ_{ij} will be different. When comparing simulation and analytical results, $\bar{\mu}_{ij}$ is used in (1) if the simulation is repeated for a number of seedings and μ_{ij} is used in (1) if only one seeding is used.

When an actual program is available for analysis, the number of source statements in an arc s_{ij} and the mean number of source statements between errors M (obtained from historical module error records) can be used to obtain the μ_{ij} in (2) by the computation $\mu_{ij} = s_{ij}/M$. Then (2) becomes

$$E(k) = \left(\sum_{ij} s_{ij}(1-p_{ij})^{k-1} p_{ij} \right)/M. \qquad (3)$$

If it is desired to calculate (3) as the fraction of original errors detected, then (3) is divided by U, the expected number of original errors in a program. We obtain $U = S/M$, where S is the total number of statements in the program. If the program is a procedure of a larger module, S is the number of statements in the procedure and M is the mean number of errors between statements of the module. Thus when calculated on a fractional basis, (3) becomes

$$E(k)/U = \left(\sum_{ij} s_{ij}(1-p_{ij})^{k-1} p_{ij} \right)/S. \qquad (4)$$

It is important to use (4) when comparing detected errors from different size programs, since we would expect to find more errors in larger programs.

The probability of traversing arc ij is computed by multiplying the probability of reaching node i by the branch probability for arc ij, $1/n$, where n is the number of arcs emanating from node i. The probability of reaching node i is the sum of the traversal probabilities of arcs which enter node i.

Navy Tactical Data System (NTDS) program listings were converted to directed graphs. After constructing the graphs, the numbers of nodes, arcs, paths and source statements were recorded. Simulation and analytical results were obtained for 44 procedures in two modules. The data were obtained in order to compare simulation and analytical results and to analyze complexity

measures (see Section IV). Although any values of M could have been used for these purposes, the values were computed from S/U where S and U were actual numbers of module source statements and total reported errors, respectively. In the model, the distance between errors refers to statements executed and not statements on the program listing. M is equal to the model mean distance when module testing starts (all U errors are present) and all statements are executed once (S statements).

Appendix B contains the directed graphs and tables pertaining to the simulation and analytical solutions for 31 procedures of one module and 13 procedures of a second module of the NTDS. The graphs and tables show numbers of nodes, arcs, paths and source statements. Simulation and analytical results are shown for mean number and mean fraction of errors detected and probability of arc traversal for every arc of every directed graph for a single input. Simulation results were obtained for 100 repetitions and 100 replications where a repetition is an error seeding and a replication is a path. One hundred replications were used for each of 100 repetitions.

Validation Tests

a. Mean Number of Errors Seeded.

Errors are seeded in the simulation model with exponentially distributed distances (number of source statement) between errors. This is equivalent to a Poisson distribution of number of errors seeded per arc, with the mean number seeded being proportional to arc length. The total number of errors seeded in the directed graph is also Poisson distributed with mean S/M and standard deviation $(S/M)^{1/2}$. Since the sample was N = 100 seedings, the normal approximation of the Poisson was used. Since the variance is known, the test statistic:

$Z = \left(\bar{X} - (S/M) \right) / (S/MN)^{1/2}$, where $\bar{X}$ is the mean number of errors seeded over N seedings with $M = 21$ statements between errors. A two sided test was used with $\alpha = .05$. Eight Module 1 NTDS procedures were tested for error seeding separately and independently of the results shown in Appendix B for error detection.

H_0: $\mu = S/M$

H_1: $\mu \neq S/M$

Reject H_0 if $|Z| > 1.96$

The results of the hypothesis tests are shown in Table III-1.

TABLE III-1

Mean Error Seeding Tests

Procedure	S	S/M	$\bar{X}$	$(S/M)^{1/2}$	$\lvert Z \rvert$
8	10	.476	.550	.690	1.072
14	9	.429	.430	.655	.015
25	8	.381	.400	.617	.308
34	15	.714	.780	.845	.781
39	17	.810	.840	.900	.333
47	12	.571	.680	.756	1.442
48	13	.619	.700	.787	1.0292
53	11	.524	.630	.724	1.464

Although H_0 would be accepted in each of the above tests, the simulation error seeding was consistently high.

Another test involved a graph with a single input and a single exit node. The arc joining these nodes had a length of 10 and the parameter MEANER was set to 1 so that the expected number of seeded errors was 10. The subroutine SEED was called 1000 times to seed errors in this arc, and the mean number of errors seeded was 9.995. This test was conducted by running a 1000 inputs through the graph and since each input traverses the single arc the number of errors found is the same as the number seeded $\left(Z = (10 - 9.9995)/(10 \cdot 1000)^{1/2} = .00005 \right)$.

b. Probability of Arc Traversal

Traversals on a given arc or path are independent and the probability of traversal is constant on successive trials. The number of traversals in an arc ij is binomially distributed. The probability of arc traversal is P_{ij} and the relative frequency of traversal obtained from simulation is P'_{ij}, so that $E(P'_{ij}) = P_{ij}$ and $V(P'_{ij}) = P_{ij}(1-P_{ij})/N$, where N is the number of independent trails (100 replications x 100 repetitions = 10,000 trials). Since a normal approximation can be used when N is this large and the variance is known, the test statistic

$$Z = \left(P'_{ij} - P_{ij}\right) / \left(P_{ij}(1-P_{ij})N\right)^{1/2}$$

was employed for a two sided test with $\alpha = .05$. Eight procedures listed in Appendix B (different procedures than used in seeding tests) were randomly selected. A branch node of each of the eight procedures and its outgoing arcs were also randomly selected.

H_0: $\mu = P_{ij}$

H_1: $\mu \neq P_{ij}$

Reject H_0 if $|Z| > 1.96$

The results of the hypothesis tests are shown in Table III-2. The hypothesis H_0 would be accepted in each case. Simulation and analytical results are close, as can be seen by examining Appendix B.

TABLE III-2

Probability of Arc Traversal Tests

Module/ Procedure	Arc	P'_{ij}	P_{ij}	$\left((P_{ij})(1-P_{ij})\right)^{1/2}$	$\lvert Z \rvert$
1/28	4,5	.1256	.1250	.331	.181
1/28	4,6	.1235	.1250	.331	.453
1/29	2,3	.5014	.5000	.500	.280
1/29	2,6	.4986	.5000	.500	.280
1/44	5,6	.2458	.2500	.433	.970
1/44	5,k	.2419	.2500	.433	1.871
1/49	4,5	.1260	.1250	.331	.302
1/49	4,8	.1232	.1250	.331	.544
1/60	6,7	.1225	.1250	.331	.755
1/60	6,17	.1213	.1250	.331	1.118
2/23	2,3	.4947	.5000	.500	1.060
2/23	2,4	.5053	.5000	.500	1.060
2/48	5,6	.1869	.1875	.390	.154
2/48	5,7	.1850	.1875	.390	.641
2/86	5,12	.2524	.2500	.433	.554
2/86	5,20	.2472	.2500	.433	.647

c. Numbers of Arc and Path Traversals

The branching mechanism was tested by including in the program a traversal counter which records the number of times each arc is traversed during a program run. The correct functioning of the counter was confirmed by obtaining detailed output (MOUT = 1) for several different graphs and manually confirming that the count corresponded to the detailed output.

Several runs were made to test the actual traversal count against the expected number. Two specific tests are reported below:

1. The graph with a single input node connected to each of 4 terminal nodes was used in a run with 1 input, 4000 replications, and 1 seeding. The expected number of traversals on each arc is 1000, and the traversal counter showed 994, 1004, 1006, 996 as the observed frequencies. These values easily pass a chi square test at the 99% confidence level ($\chi^2 = .104$, $\chi^2_{3,.99} = 11.34$).

2. A second test used a graph with node 1 connected to nodes 2 and 3, node 3 connected to nodes 4 and 5, and node 5 connected to nodes 6 and 7. In addition 10 inputs were used to test if the branching mechanism works for multiple inputs. The test used 100 replications and 2 seedings. The graph has a total of four paths from input to termination. The expected number of traversals for the path terminating at each node is shown below along with the actual number of traversals produced in this test. These values also pass the chi square test at the 99% confidence level ($\chi^2 = .419$, $\chi^2_{3,.99} = 11.34$).

path ending at	2	4	6	7
expected traversals	1000	500	250	250
actual traversals	1011	491	253	245

IV. ANALYSIS OF COMPLEXITY

One objective of the research was to identify complexity measures which could be used to estimate the difficulty of detecting errors in a program. If these measures could be identified, they would prove useful in program design and testing. Complexity measures would be used during program design to avoid structures which are difficult to test and during testing to allocate resources to testing on the basis of estimated difficulty of error detection.

Four complexity measures were evaluated: numbers of nodes (Nn), arcs, (Na), paths (Np) and source statements (S). A brief explanation will be given of the significance of these measures as indices of difficulty of error detection. These measures were developed for 31 procedures of one NTDS module and 13 procedures of another NTDS module. A procedure is a subset of a module.

Nodes

Nodes can be categorized as follows:

a. Start. Signifies program beginning. No entry and one or more exits.

b. Terminal. Signifies program end. One or more entries and no exit.

c. Branch. Single entry and multiple exits.

d. Merge. Multiple entries and single exit.

e. Merge and Branch. Multiple entries and exits.

f. Transfer. Single entry and exit. Commonly used to signify a call to a sub-procedure and to indicate the return point in the calling procedure.

g. Dummy. A special case of f. An artificial node in the simulation for handling parallel arcs.

h. Transient. Nodes which delineate the beginning and ending of sub-procedures.

As compared to a structure with a start node, terminal node and one arc connecting the two, the addition of nodes in categories c, d, and e will add to the number of paths and decrease the probability of reaching certain arcs. Thus, error detection is affected. Transfer nodes do not affect the structure of a program, but do add to its complexity because these nodes correspond to a point of transfer of control to a sub-procedure and a return to the calling procedure. Transient nodes were not counted in Nn because transient nodes are part of a sub-procedure; however, the transient nodes were necessary in order to complete certain paths in the directed graph. Dummy nodes were not counted as part of Nn because they were used only to specify a directed graph in the model and did notcontribute to program complexity. All other nodes were counted. It would be useful in future work to develop an additional measure consisting only of nodes involving branching, since the presence of these nodes may cause the probabilities of arc traversal in the outgoing arcs to be less than the probabilities in incoming arcs, thus increasing the difficulty of error detection in paths which contain the outgoing arcs.

Arcs

Transient arcs are arcs contained in sub-procedures. These arcs were not counted in Na. However, these arcs were needed to complete certain paths in the directed graphs. An entire sub-procedure was represented by two nodes (start and end nodes) and one arc. This was done because it was infeasible to represent in the model an entire module of approximately 2000 source statements and many interconnected procedures. Secondly, the point of view was adopted that sub-procedures called by procedures have been checked out and any errors detected reside in the main procedure. Thus the sub-procedure

was treated as a zero source statement (no errors were seeded) arc. This treatment corresponds to viewing the test of a procedure as a unit test in which it is assumed that interacting units (sub-procedures) are working; the focus is on finding errors in the main procedure. An important facet of testing is the integration test in which the focus is on the interactions between units. The model could be employed in this vein by conceiving of each procedure as an arc and the nodes as branch points to entire procedures. It was not necessary to depict transient nodes and arcs, since these components do not affect the error finding mechanism in the model. However, it was considered important to document the actual program structure rather than the structure required by the model.

Dummy arcs were not counted in Na because they were used only to represent parallel arcs in the model and did not contribute to program complexity.

An increase in the number of arcs increases the number of paths and, hence, makes error finding more difficult.

Paths

We define a path as a series of connected nodes and arcs which begins with a start node and ends with a terminal node. A definitional problem arises in the case of paths which contain cycles. A graph containing cycles has an infinite number of paths. However, for our purposes, traversals in cycles were counted the minimum number of times. A cycle contained in a path was traversed one time. This treatment of paths is consistent with the model assumption that all errors in an arc are detected on the first traversal. Cycles in the model do not represent DO loops in a program, where the program is forced to iterate in a loop a

specified number of times. Rather, a cycle contains nodes and arcs which may be revisited.

As the number of paths in a program increases, error finding becomes more difficult because the number of areas in the program which must be searched increases. It should be noted that number of paths is not independent of number of nodes and arcs; number of paths is a function of number of nodes and arcs.

Source Statements

The number of source statements S will affect error detection in the model because an assumption of the model is that the number of original errors in a program is proportional to the number of source statements. Based on this assumption, error detection would increase as program size increases, all other factors (number of paths) being equal. In order to not mask the effect of other factors, the number of errors detected on the first input E(1), as a fraction of the original number of errors U, is used instead of E(1). The relationship between error detection and S is complex because error detection depends on both U (a function of S) and structural properties (paths).

Structural properties are not independent of number of source statements. Since few programs are written as one line of code or a single arc, the number of nodes, arcs and paths will, in general, increase with number of source statements.

Analysis of Complexity Measures

Values of Nn, Na, Np, S and E(1)/U (analytical solution) are tabulated in Table IV-1. Plots of E(1)/U versus Nn, Na, Np and S

are shown in Figures IV-1 to IV-4, respectively. Points are plotted for both Module 1 and Module 2. Since the data were shown to be highly non-linear when plotted on a linear scale, the data were plotted first on semi-log and then on log-log scales (Figures IV-1 to IV-4) to see whether a straight line would emerge indicative of a non-linear relationship. As seen in the figures, the data are still scattered on a log-log scale. In many cases there are multiple values of errors detected for a given value of complexity. This result suggests that error detection is a function of multiple complexity measures. However, lack of independence among variables makes the identification of a multi-variate relationship difficult. Other measures, such as path length, connectivity and reachability, may prove more illuminating as indices of error detection.

In order to more clearly indicate the relationship between error detection and a single complexity measure, multiple values of $E(1)/U$ were averaged (for three or more values) and plotted in Figures IV-5 and IV-6 for paths and nodes, respectively. These curves provide a better indication of the inverse relationship between error detection and complexity. However, the reduced sample which results from averaging is inadequate for fitting an equation to the data.

TABLE IV-1

COMPLEXITY MEASURES

Module/ Procedure	Nn	Na	Np	S	E(1)/U
1/2	12	21	26	37	.1700
1/8	8	11	3	10	.6000
1/11	6	7	3	8	.5938
1/14	4	5	4	9	.7222
1/19	17	25	7	22	.3295
1/22	18	26	11	30	.3458
1/25	8	10	2	8	.6875
1/28	12	16	4	24	.6145
1/29	20	28	13	47	.6463
1/30	7	10	5	10	.3563
1/34	11	15	3	15	.9167
1/35	12	16	3	11	.8636
1/36	17	24	3	31	.4032
1/39	15	24	10	17	.3526
1/44	15	24	7	21	.4256
1/47	12	16	4	12	.7500
1/48	14	21	7	13	.6538
1/49	13	19	7	19	.2829
1/53	11	18	9	11	.4531
1/57	22	31	12	26	.2726
1/60	22	34	16	24	.3317
1/75	17	24	8	20	.6800
1/76	13	18	5	19	.4803
1/77	10	16	9	10	.6000
1/79	19	28	3	23	.3804
1/81	6	8	3	7	.5000
1/87	14	20	6	25	.4700
1/91	18	28	9	14	.2514
1/92	4	5	3	9	.4444
1/93	23	33	12	34	.2289
1/95	16	26	10	19	.2325

TABLE IV-1 (cont'd)

COMPLEXITY MEASURES

Module/ Procedure	Nn	Na	Np	S	E(1)/U
2/15	9	12	5	10	.4750
2/23	8	10	3	12	.6667
2/41	9	13	12	13	.5192
2/46	23	32	18	32	.3212
2/47	22	34	36	33	.2060
2/48	12	18	14	17	.4651
2/79	9	12	5	11	.5455
2/86	20	29	6	33	.3775
2/90	11	17	8	23	.2446
2/99	19	27	10	23	.4620
2/122	12	18	6	20	.4313
2/137	10	15	11	23	.3505
2/149	16	24	9	35	.3964

Legend

Nn: number of nodes, with dummy nodes (nodes associated with dummy parallel arcs) and transient nodes (nodes associated with called procedures) eliminated.

Na: number of arcs, with dummy arcs and transient arcs (arcs associated with called procedures) eliminated.

Np: number of paths, with paths involving cycles counted minimum number of times.

S: number of source language statements.

E(1)/U: expected fraction errors detected on first imput obtained with analytical model.

K&E LOGARITHMIC 46 7283
2 X 2.7 CYCLES MADE IN U.S.A.
KEUFFEL & ESSER CO.

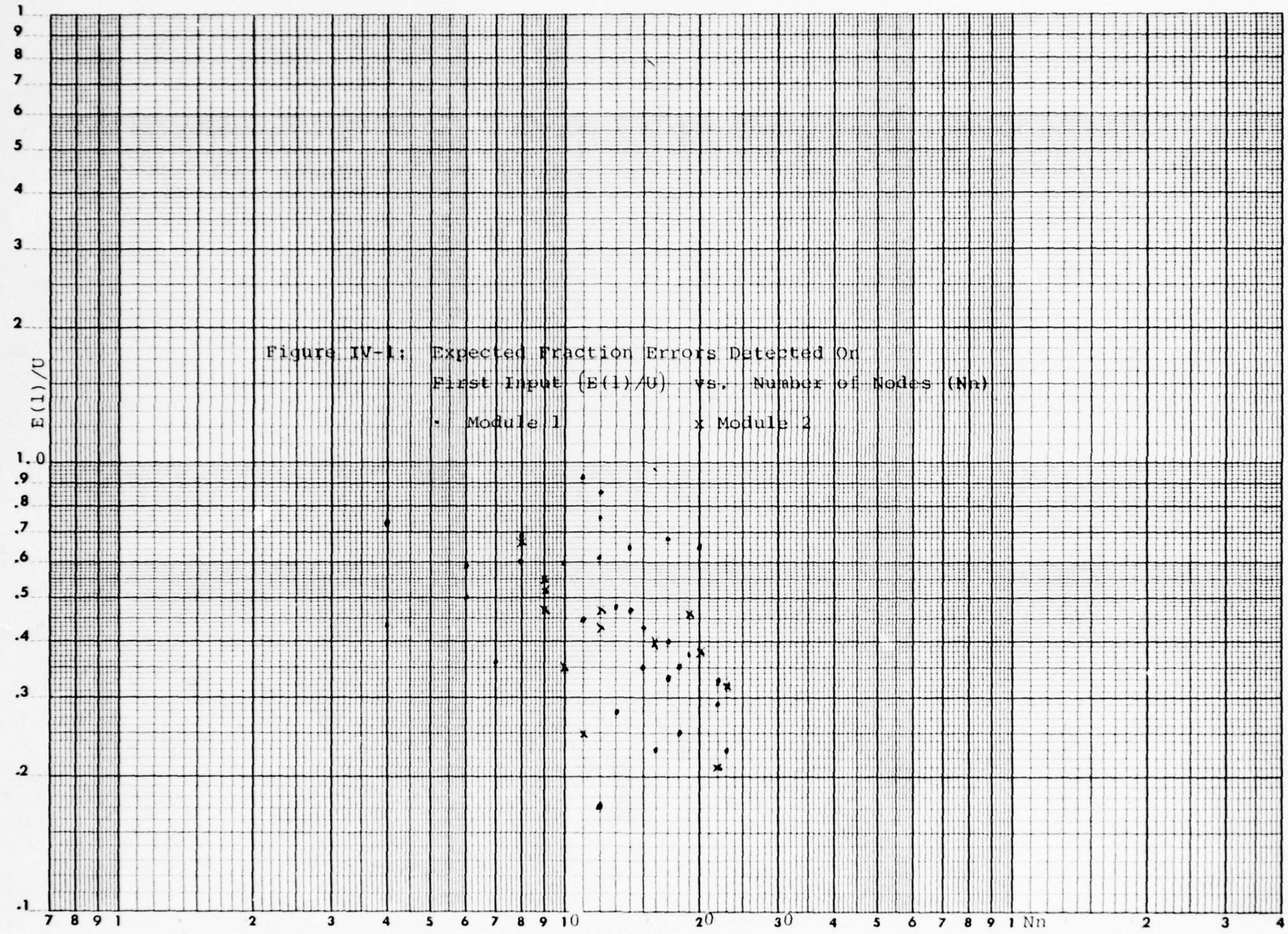
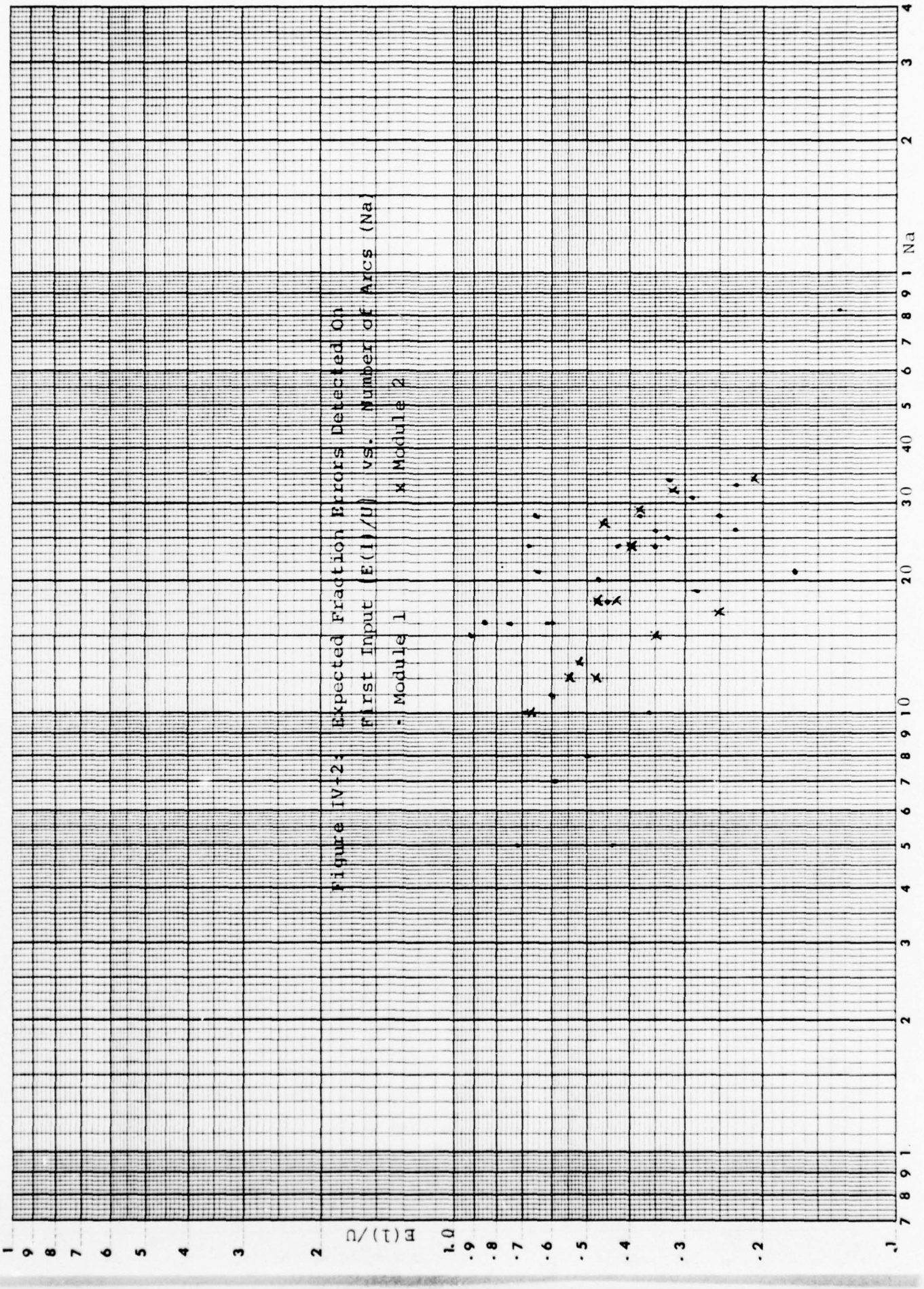


Figure IV-1: Expected Fraction Errors Detected On First Input (E(1)/U) vs. Number of Nodes (Nn)

• Module 1 x Module 2


Figure IV-2: Expected Fraction Errors Detected On First Input (E(1)/U) vs. Number of Arcs (Na)
Module 1
x Module 2
E(1)/U
Na
1.0
.9
.8
.7
.6
.5
.4
.3
.2
.1
1
2
3
4
5
6
7
8
9
10
20
30
40

K•Σ LOGARITHMIC 46 7283
2 X 2 7 CYCLES MADE IN U.S.A.
KEUFFEL & ESSER CO.

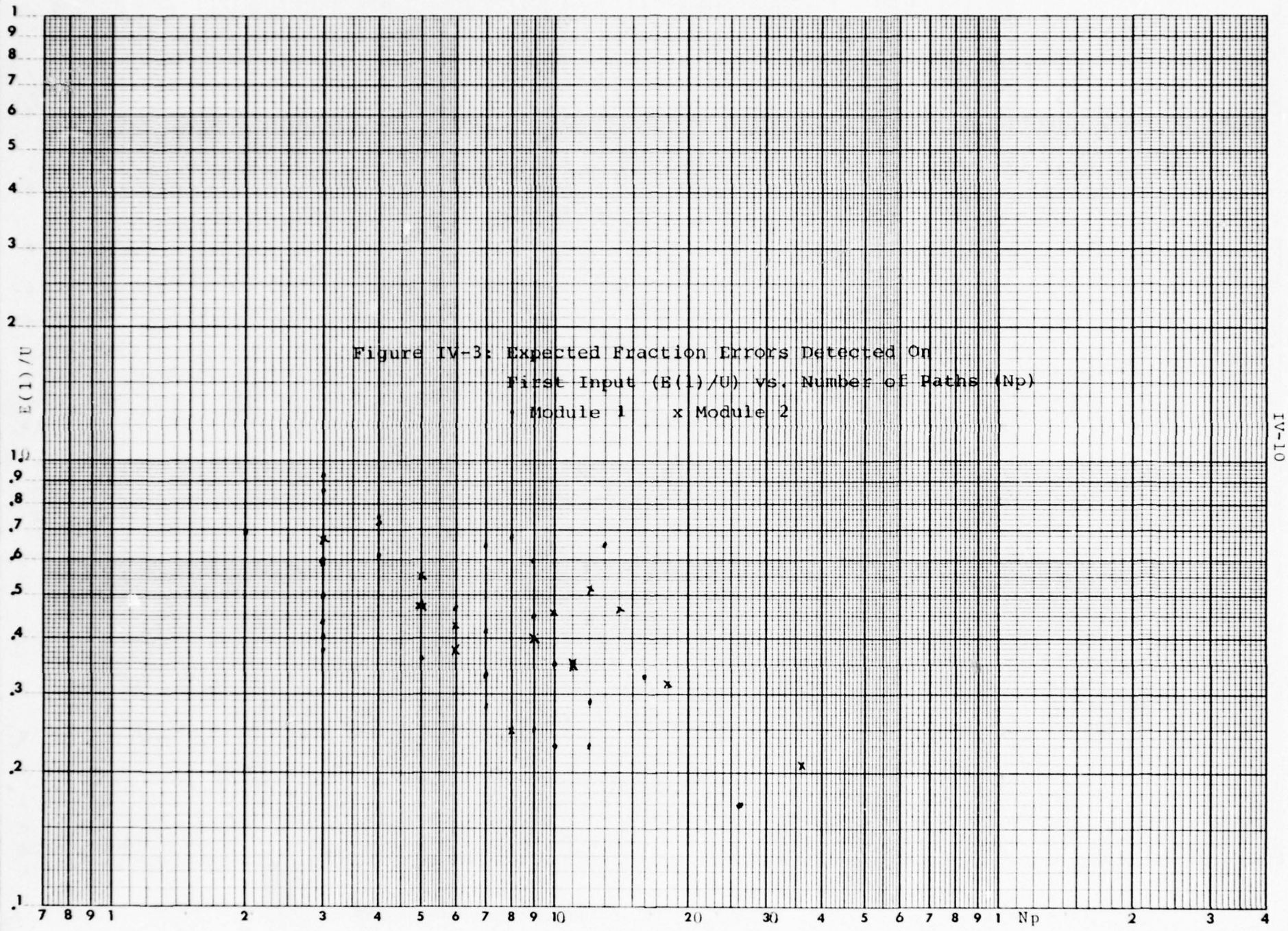


Figure IV-3: Expected Fraction Errors Detected On
First Input (E(1)/U) vs. Number of Paths (Np)
• Module 1 x Module 2

K•E LOGARITHMIC 46 7283
2 X 2 7 CYCLES MADE IN U.S.A.
KEUFFEL & ESSER CO.

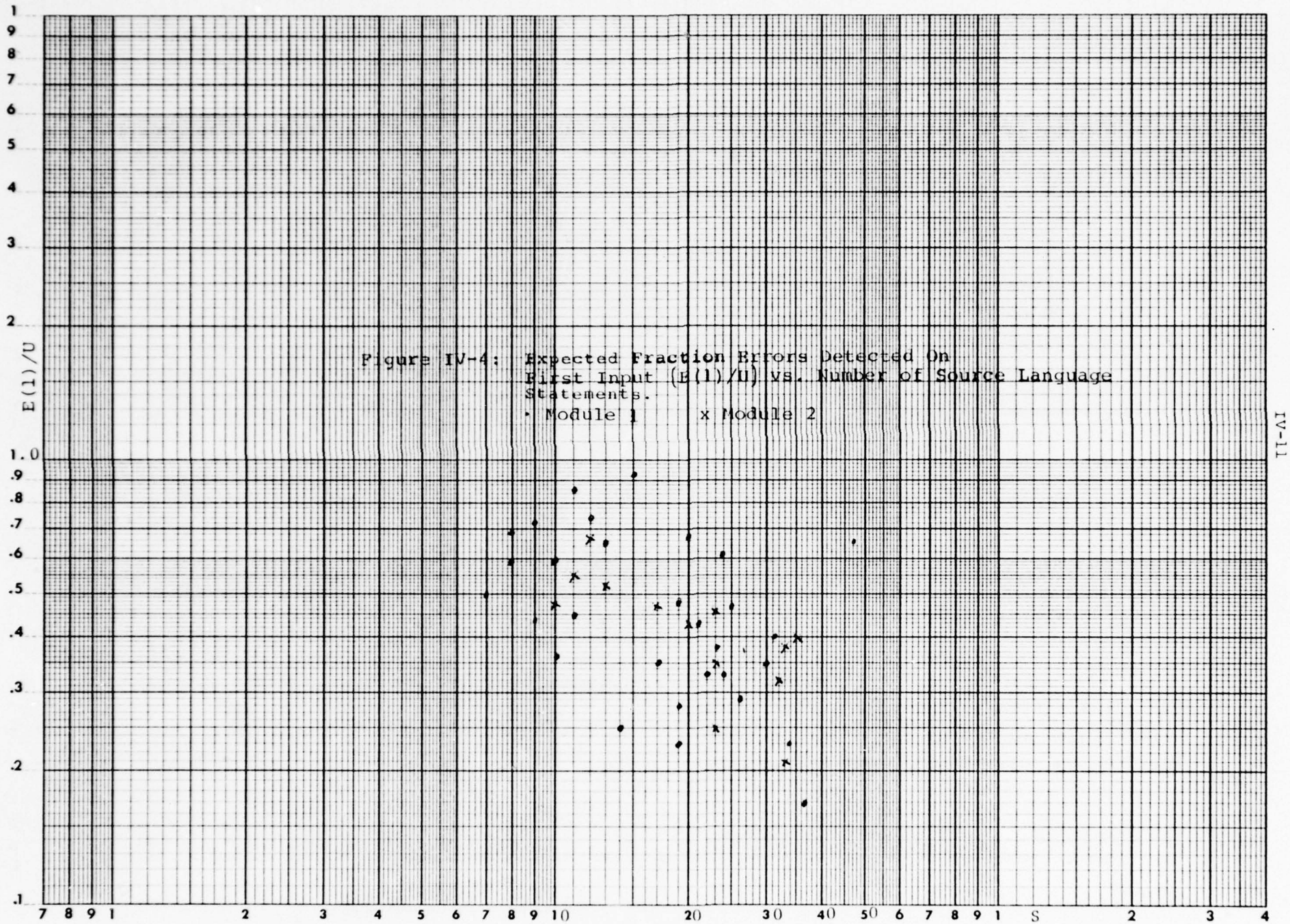


Figure IV-4: Expected Fraction Errors Detected On First Input (E(1)/U) vs. Number of Source Language Statements.
• Module 1 x Module 2

K&E LOGARITHMIC 46 7283
2 X 2 7 CYCLES MADE IN U.S.A.
KEUFFEL & ESSER CO.

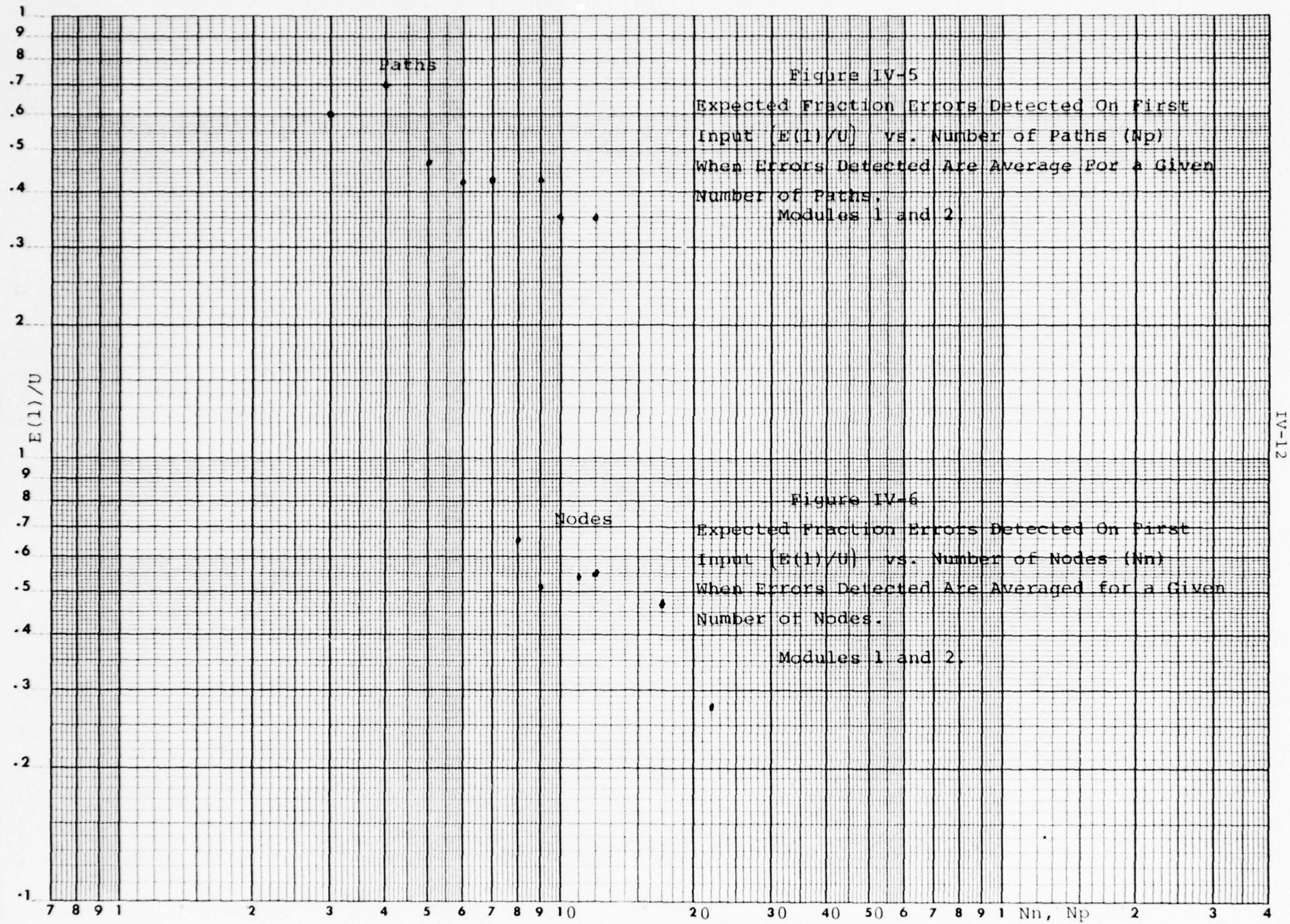


Figure IV-5

Expected Fraction Errors Detected On First Input [E(1)/U] vs. Number of Paths (Np) When Errors Detected Are Average For a Given Number of Paths.

Modules 1 and 2.

Figure IV-6

Expected Fraction Errors Detected On First Input [E(1)/U] vs. Number of Nodes (Nn) When Errors Detected Are Averaged for a Given Number of Nodes.

Modules 1 and 2.

V. SUMMARY

This report covered the following areas:

- Description of the modified error simulation model
- Validation tests of the simulation model
- Analysis of program complexity measures

A description of the revised simulation model was presented in Section II. The data input format was described in Section II-4. A flowchart appears in Appendix A.

Validation tests on the simulation model were described in Section III. Some of the tests used data shown in Appendix B. These data show: directed graphs, properties of the directed graphs, simulation solutions and analytical solutions for 44 NTDS procedures. Validation tests were conducted for error seeding and arc/path traversal. All hypothesis (validation) tests were passed. However, simulation results were consistently higher than analytical results.

Program complexity measures were analyzed in Section IV. Four complexity measures -- numbers of nodes, arcs, paths and source statements -- were plotted against fraction expected detected errors obtained from the analytical solution. The data were obtained from the NTDS procedures of Appendix B. Although the direction of the plots (error detection inversely related to complexity measures) was as expected, considerable variability in the data were exhibited. The variability indicated error detection was a function of several complexity measures instead of one. There were multiple values of error detection for many values of complexity measure. When these values were averaged the effect of a single complexity measure became

clearer. However, the reduced sample size made quantitative analysis infeasible. Although the four factors are potentially useful as measures of complexity, no single factor stood out as being a major determinant of error detection.

REFERENCES

[1] G. H. Bradley, G. T. Howard, N. F. Schneidewind, T. F. Green, and G. W. Montgomery, "System Test Methodology," Vol. 1, Naval Postgraduate School, NPS55Ss75072A and Vol. II, NPS55Ss75072B, July 1975.

[2] G. H. Bradley, T. F. Green, G. T. Howard and N. F. Schneidewind, "Structure and Error Detection in Computer Software," Proceedings AIIE Conference, pp. 54-59, 1975.

[3] N. F. Schneidewind and T. F. Green, "Simulation of Error Detection in Computer Programs," Proceedings of the Symposium on the Simulation of Computer Systems, National Bureau of Standards, pp. 101-105, 1975.

[4] T. F. Green, N. F. Schneidewind, G. T. Howard, and T. Pariseau, "Program Structures, Complexity and Error Characteristics," Proceedings of the Computer Engineering Conference, Microwave Research Institute, Polytechnic Institute of New York, 1976.

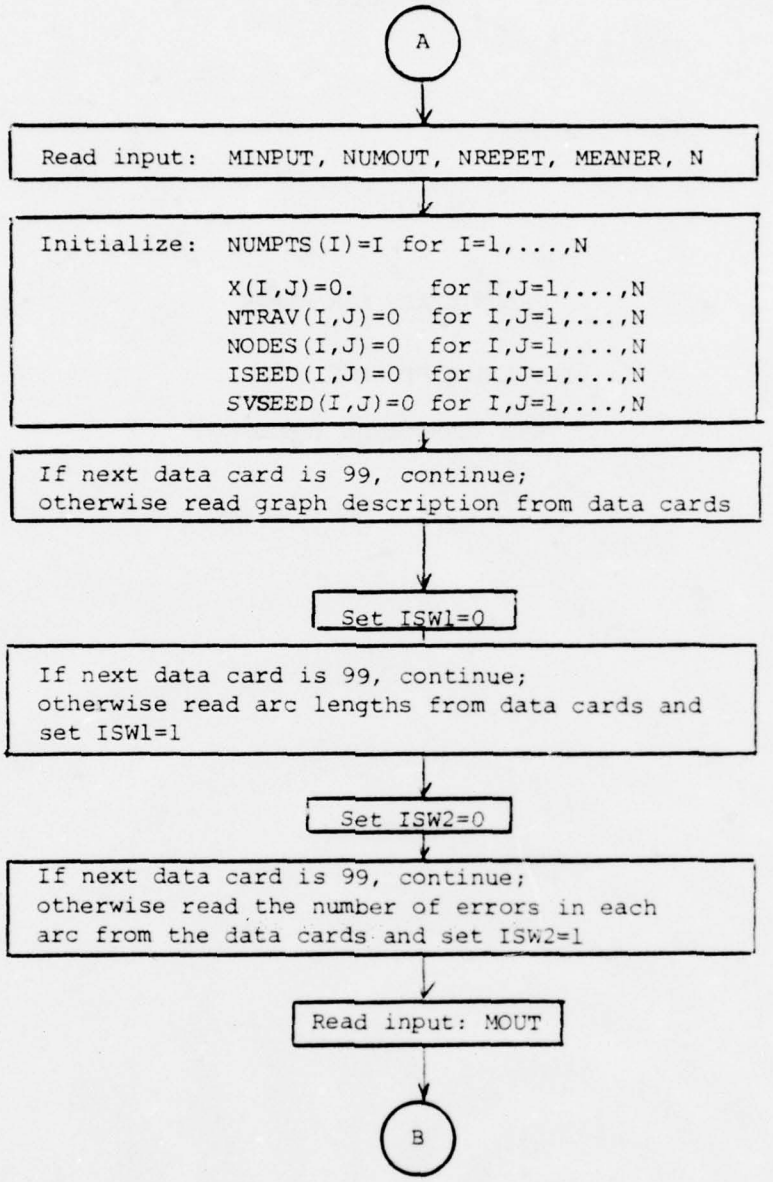
[5] S. Litwin and R. J. Pariseau, "Variance of the Number of Errors Detected During a Set of Random Passes Through an Error Laden Graph," Naval Air Development Center Technical Memorandum, 52TM 76-STS-001, 6 April 1976.

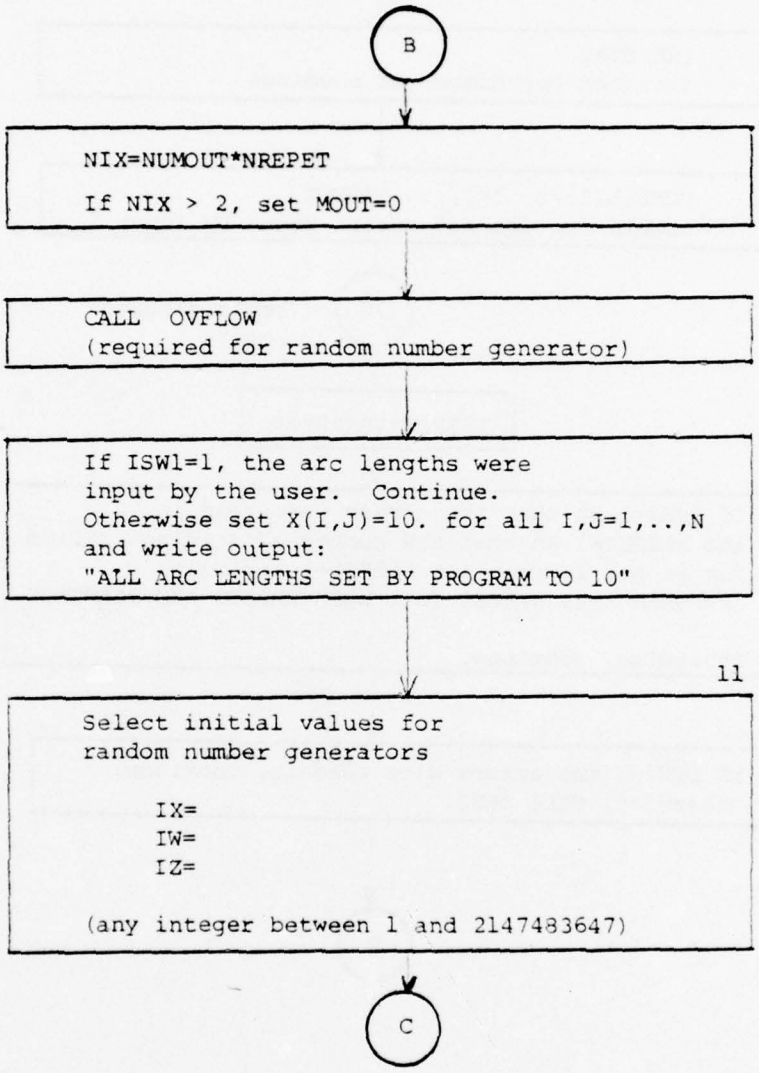
APPENDIX A

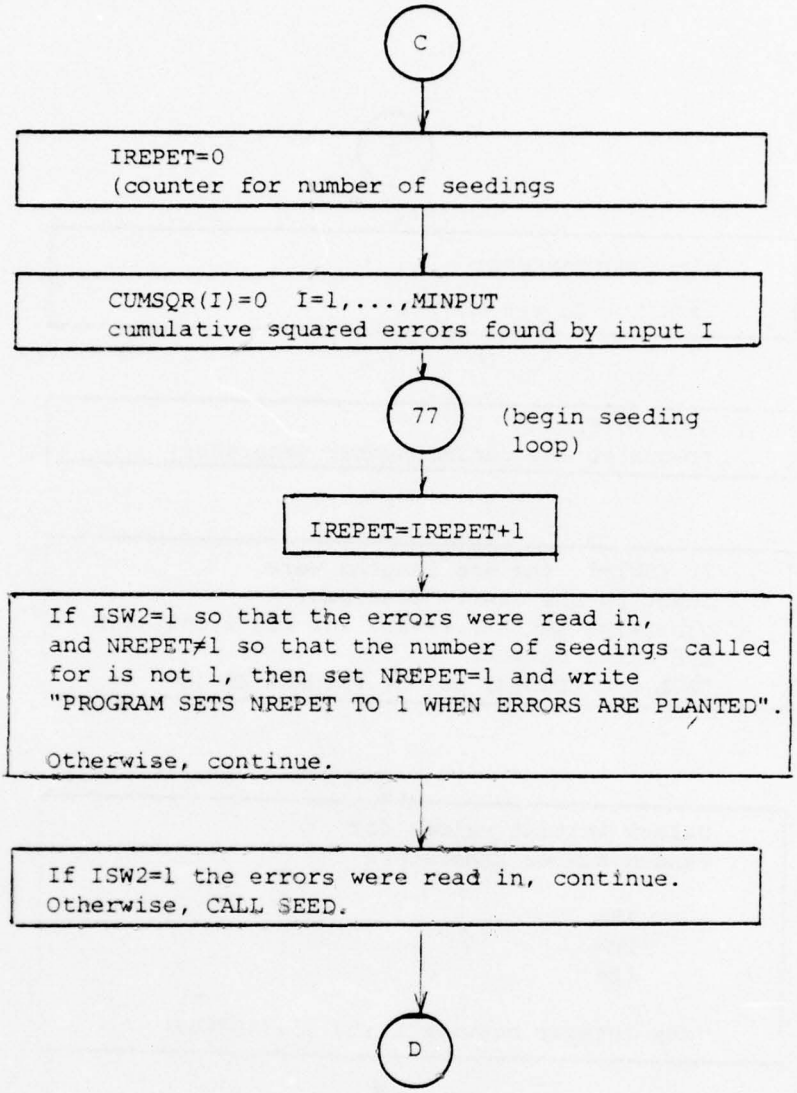
FLOW CHART FOR ERROR

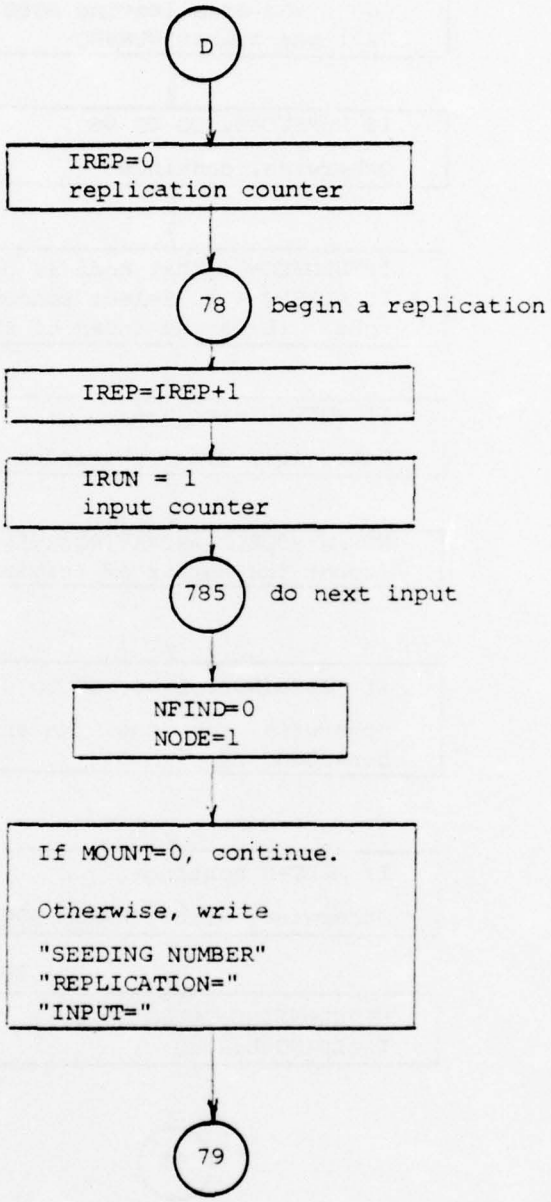
SIMULATION PROGRAM (MAIN),

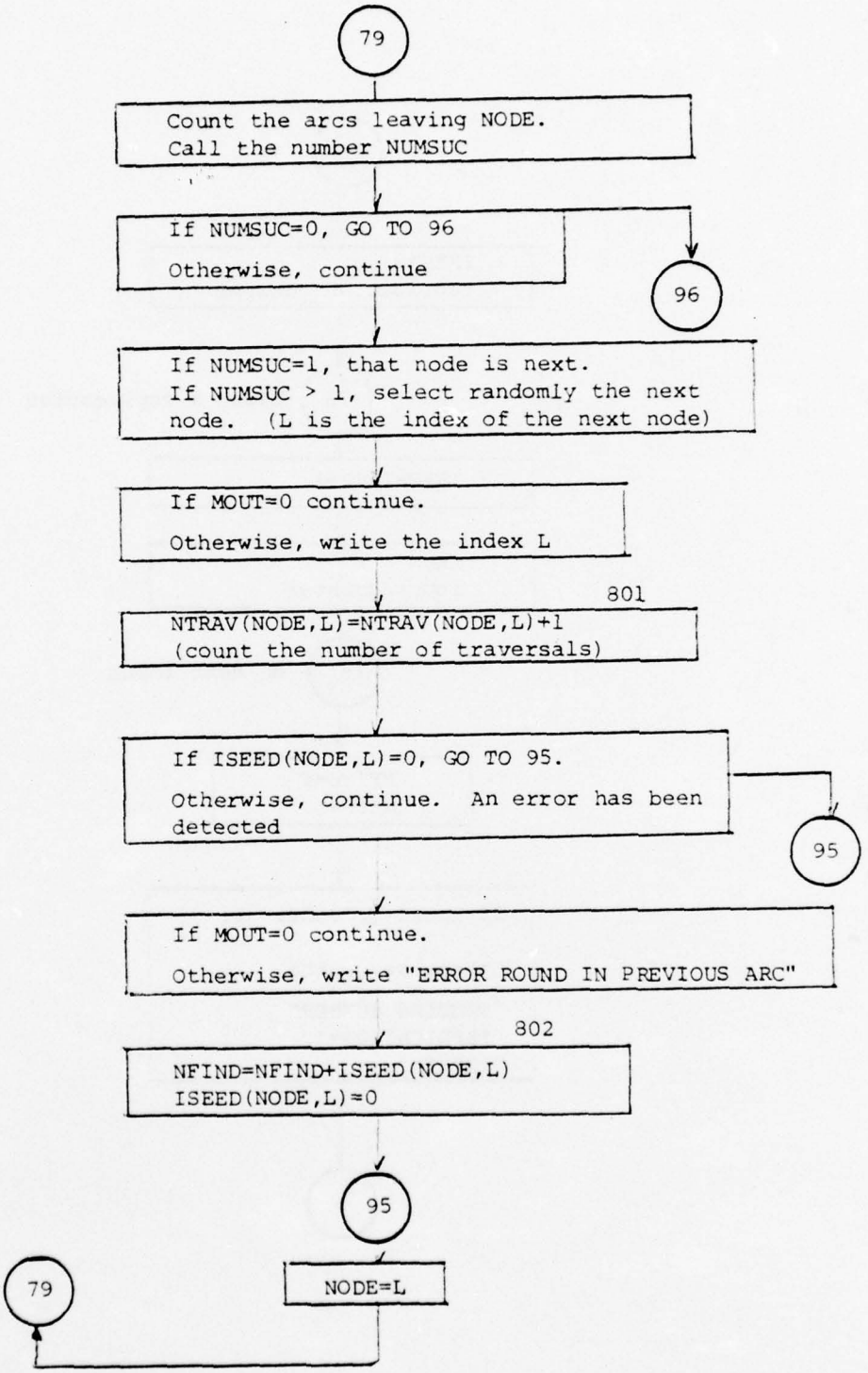
AND SUBROUTINE (SEED)

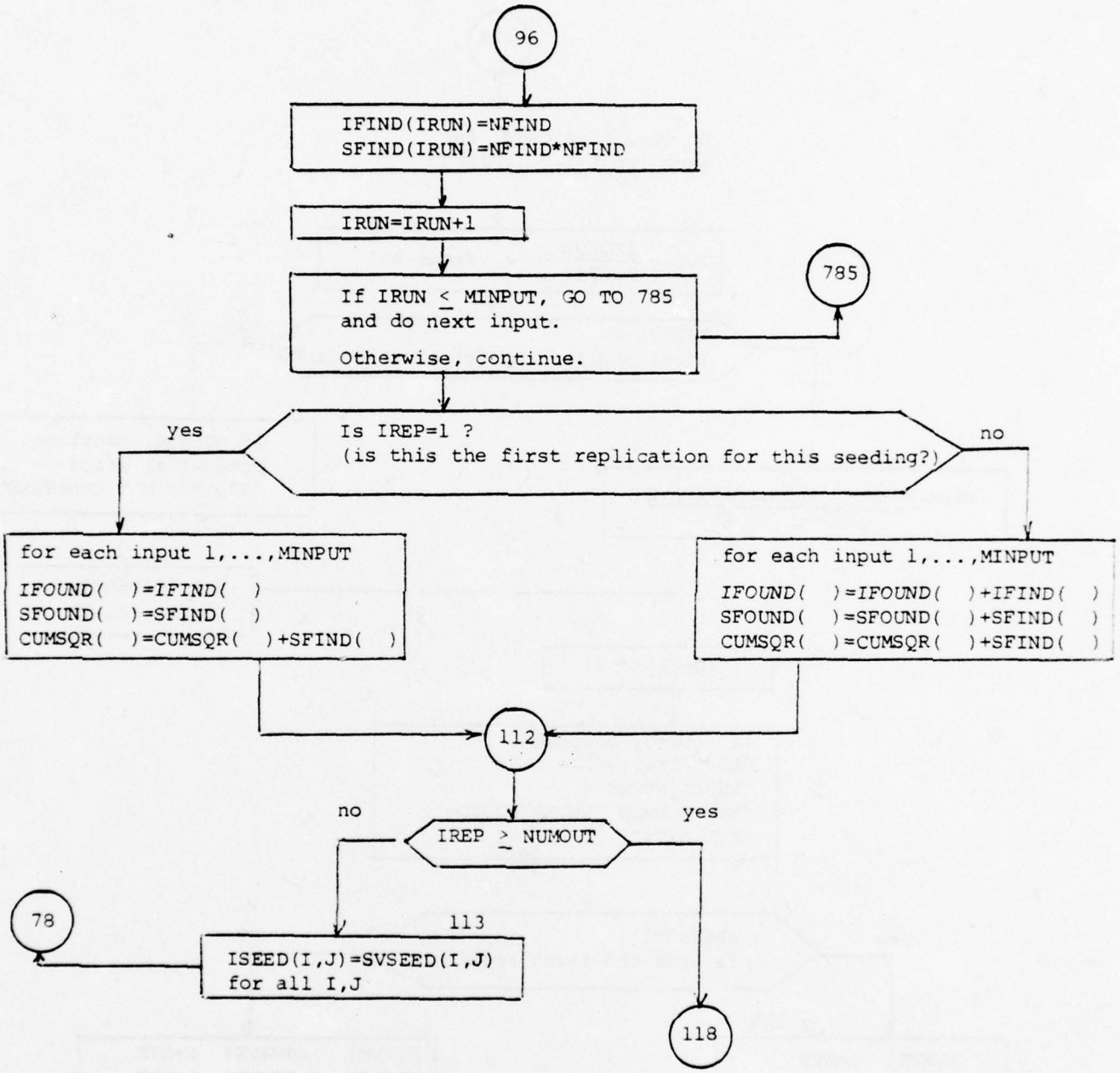

A
Read input: MINPUT, NUMOUT, NREPET, MEANER, N
Initialize: NUMPTS(I)=I for I=1,...,N
X(I,J)=0. for I,J=1,...,N
NTRAV(I,J)=0 for I,J=1,...,N
NODES(I,J)=0 for I,J=1,...,N
ISEED(I,J)=0 for I,J=1,...,N
SVSEED(I,J)=0 for I,J=1,...,N
If next data card is 99, continue;
otherwise read graph description from data cards
Set ISW1=0
If next data card is 99, continue;
otherwise read arc lengths from data cards and
set ISW1=1
Set ISW2=0
If next data card is 99, continue;
otherwise read the number of errors in each
arc from the data cards and set ISW2=1
Read input: MOUT
B

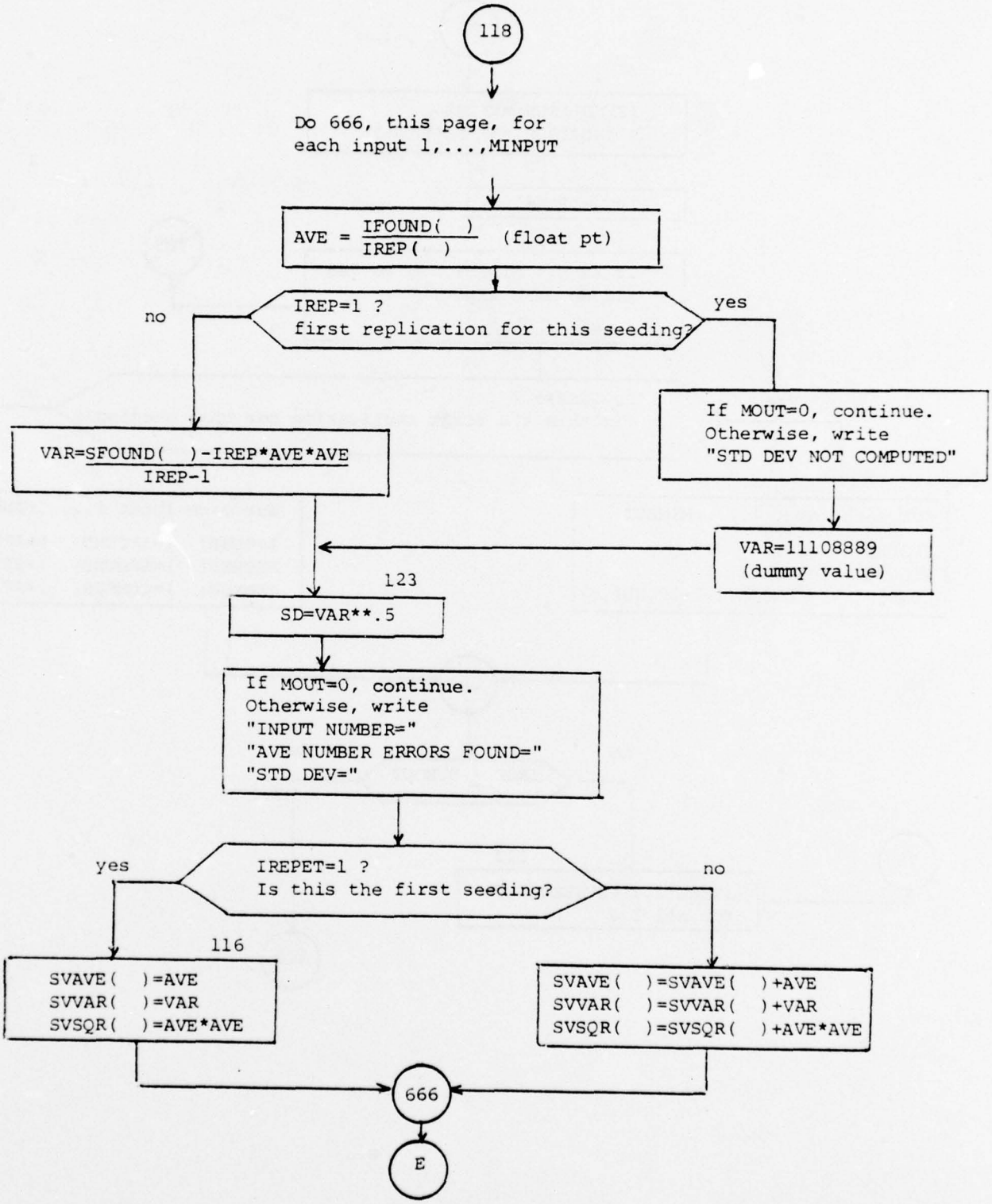

B
NIX=NUMOUT*NREPET
If NIX > 2, set MOUT=0
CALL OVFLOW
(required for random number generator)
If ISW1=1, the arc lengths were
input by the user. Continue.
Otherwise set X(I,J)=10. for all I,J=1,..,N
and write output:
"ALL ARC LENGTHS SET BY PROGRAM TO 10"
11
Select initial values for
random number generators
IX=
IW=
IZ=
(any integer between 1 and 2147483647)
C

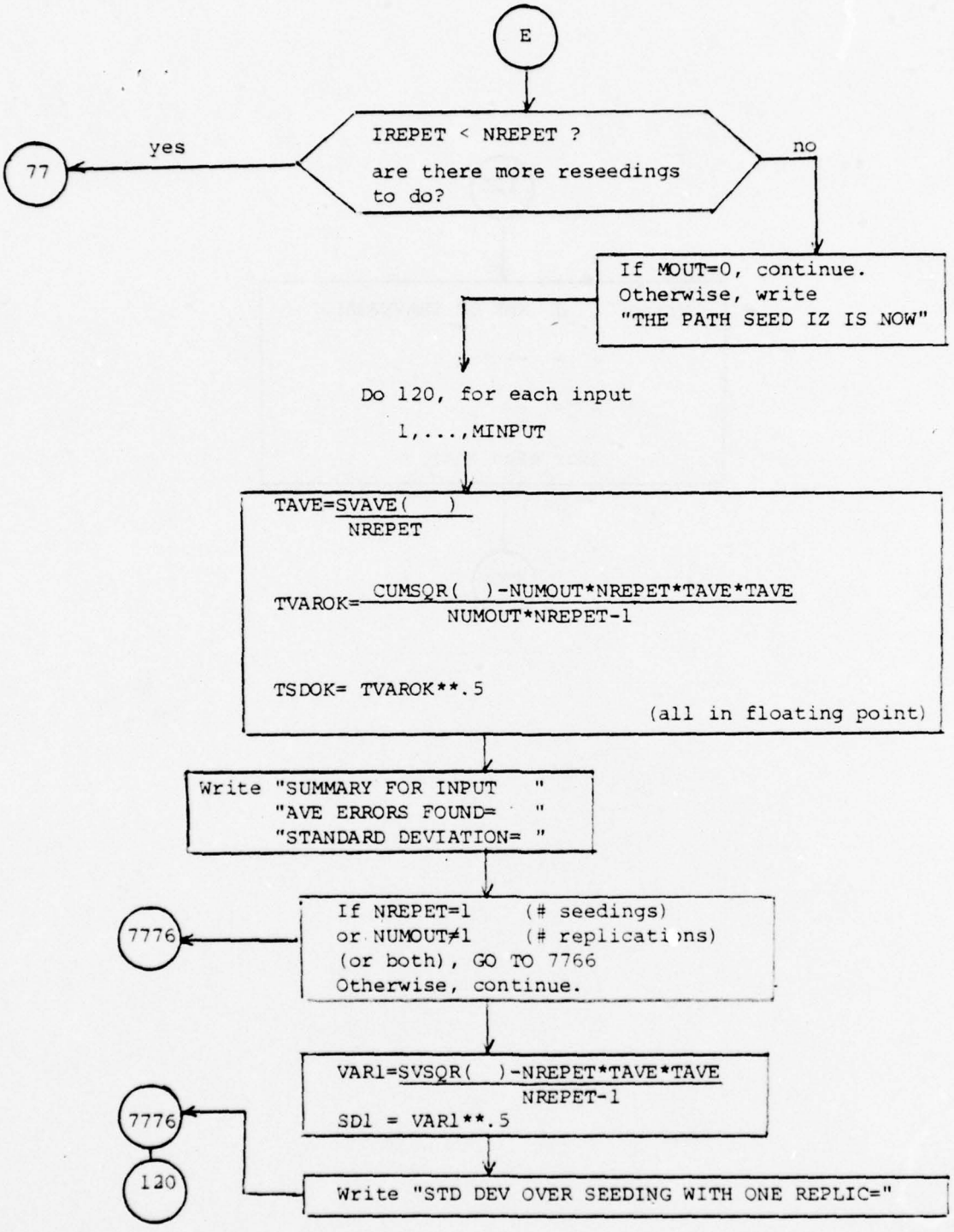

C
IREPET=0
(counter for number of seedings
CUMSQR(I)=0 I=1,...,MINPUT
cumulative squared errors found by input I
77
(begin seeding loop)
IREPET=IREPET+1
If ISW2=1 so that the errors were read in, and NREPET≠1 so that the number of seedings called for is not 1, then set NREPET=1 and write "PROGRAM SETS NREPET TO 1 WHEN ERRORS ARE PLANTED".
Otherwise, continue.
If ISW2=1 the errors were read in, continue.
Otherwise, CALL SEED.
D

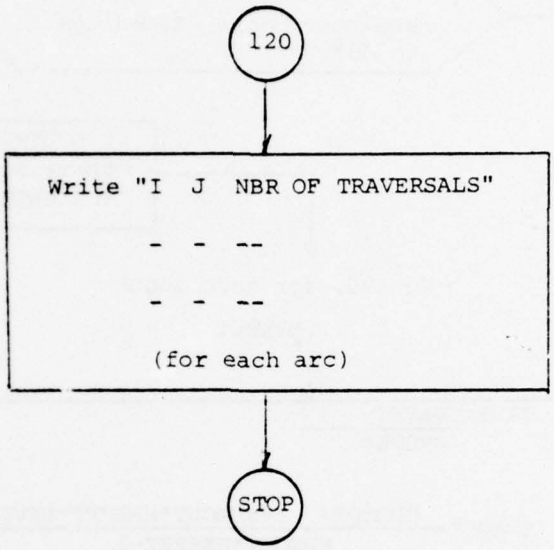

D
IREP=0
replication counter
78
begin a replication
IREP=IREP+1
IRUN = 1
input counter
785
do next input
NFIND=0
NODE=1
If MOUNT=0, continue.
Otherwise, write
"SEEDING NUMBER"
"REPLICATION="
"INPUT="
79


79
Count the arcs leaving NODE.
Call the number NUMSUC
If NUMSUC=0, GO TO 96
Otherwise, continue
96
If NUMSUC=1, that node is next.
If NUMSUC > 1, select randomly the next node. (L is the index of the next node)
If MOUT=0 continue.
Otherwise, write the index L
801
NTRAV(NODE,L)=NTRAV(NODE,L)+1
(count the number of traversals)
If ISEED(NODE,L)=0, GO TO 95.
Otherwise, continue. An error has been detected
95
If MOUT=0 continue.
Otherwise, write "ERROR ROUND IN PREVIOUS ARC"
802
NFIND=NFIND+ISEED(NODE,L)
ISEED(NODE,L)=0
95
NODE=L
79

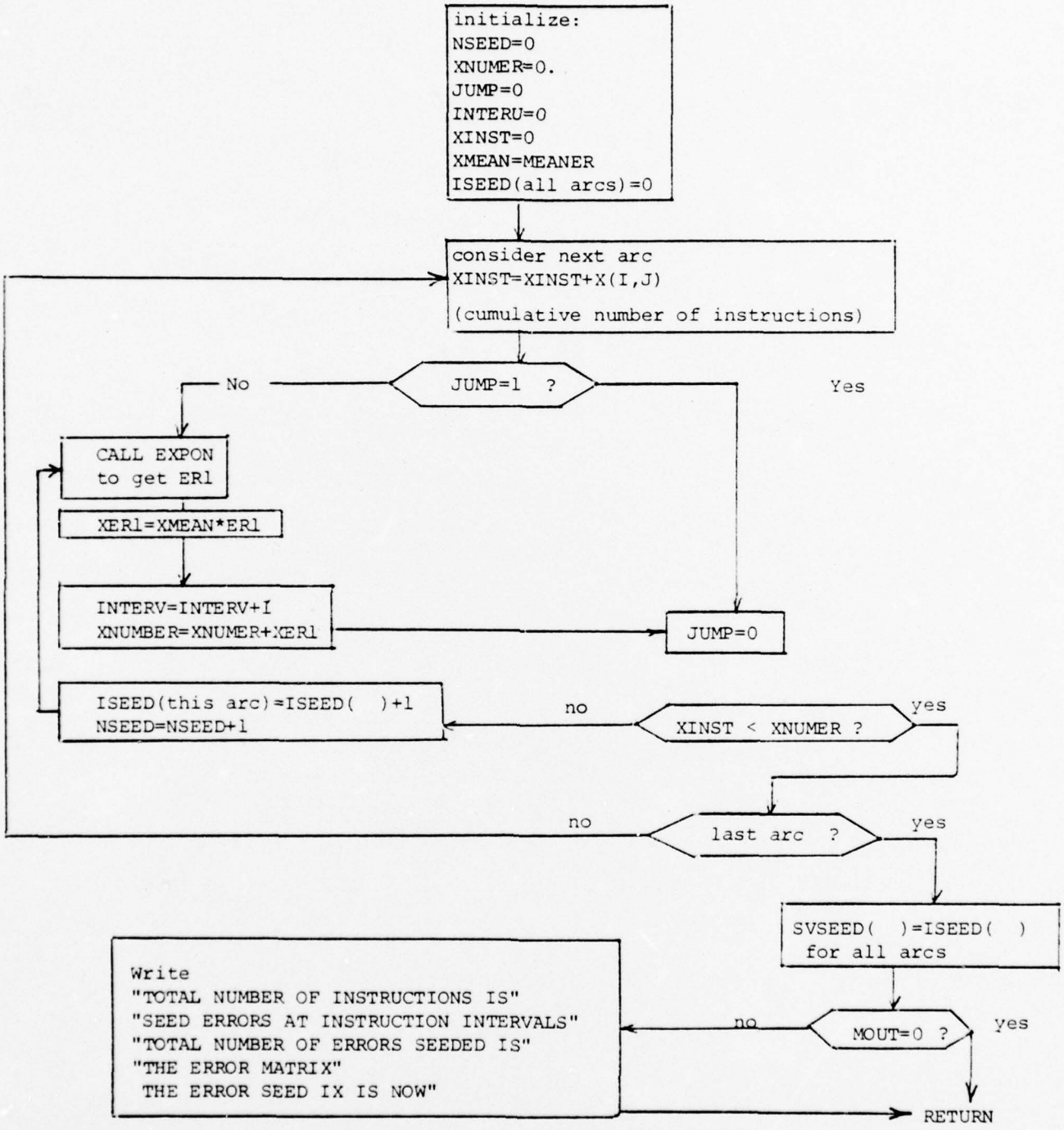

96
IFIND(IRUN)=NFIND
SFIND(IRUN)=NFIND*NFIND
IRUN=IRUN+1
785
If IRUN ≤ MINPUT, GO TO 785
and do next input.
Otherwise, continue.
yes
Is IREP=1 ?
(is this the first replication for this seeding?)
no
for each input 1,...,MINPUT
IFOUND( )=IFIND( )
SFOUND( )=SFIND( )
CUMSQR( )=CUMSQR( )+SFIND( )
for each input 1,...,MINPUT
IFOUND( )=IFOUND( )+IFIND( )
SFOUND( )=SFOUND( )+SFIND( )
CUMSQR( )=CUMSQR( )+SFIND( )
112
no
IREP ≥ NUMOUT
yes
78
113
ISEED(I,J)=SVSEED(I,J)
for all I,J
118


118
Do 666, this page, for each input 1,...,MINPUT
AVE = IFOUND( ) / IREP( ) (float pt)
IREP=1 ? first replication for this seeding?
no
yes
VAR=SFOUND( )-IREP*AVE*AVE / IREP-1
If MOUT=0, continue. Otherwise, write "STD DEV NOT COMPUTED"
VAR=11108889 (dummy value)
123
SD=VAR**.5
If MOUT=0, continue. Otherwise, write "INPUT NUMBER=" "AVE NUMBER ERRORS FOUND=" "STD DEV="
IREPET=1 ? Is this the first seeding?
yes
no
116
SVAVE( )=AVE
SVVAR( )=VAR
SVSQR( )=AVE*AVE
SVAVE( )=SVAVE( )+AVE
SVVAR( )=SVVAR( )+VAR
SVSQR( )=SVSQR( )+AVE*AVE
666
E


E
IREPET < NREPET ?
are there more reseedings to do?
yes
77
no
If MOUT=0, continue.
Otherwise, write
"THE PATH SEED IZ IS NOW"
Do 120, for each input
1,...,MINPUT
TAVE=SVAVE( )/NREPET
TVAROK=(CUMSQR( )-NUMOUT*NREPET*TAVE*TAVE)/(NUMOUT*NREPET-1)
TSDOK= TVAROK**.5
(all in floating point)
Write "SUMMARY FOR INPUT "
"AVE ERRORS FOUND= "
"STANDARD DEVIATION= "
7776
If NREPET=1 (# seedings)
or NUMOUT≠1 (# replications)
(or both), GO TO 7766
Otherwise, continue.
VAR1=(SVSQR( )-NREPET*TAVE*TAVE)/(NREPET-1)
SD1 = VAR1**.5
7776
120
Write "STD DEV OVER SEEDING WITH ONE REPLIC="


120
Write "I J NBR OF TRAVERSALS"
- - --
- - --
(for each arc)
STOP

SUBROUTINE SEED



APPENDIX B

DIRECTED GRAPHS, SIMULATION AND ANALYTICAL RESULTS

1. Notation used in Appendix B

i: node i

j: node j

P'_{ij}: relative frequency of traversing arc ij one or more times (simulation model)

P_{ij}: probability of traversing arc ij one or more times (analytical model).

S_{ij}: number of source statements in arc ij.

$E(1) = \sum_{ij} P_{ij} S_{ij}/M$: expected number of errors detected on first input, obtained from analytical model, where M is the mean number of source statements between errors (M = 21 for Module 1 and M = 51 for Module 2).

$U = S/M$: expected number of errors in a program, where S is the number of source statements in a program.

$E'(1)/U = \sum_{ij} P_{ij} S_{ij}/S$: expected fraction of number of errors detected on first input.

E'(1): mean number of errors detected on first input obtained from simulation model.

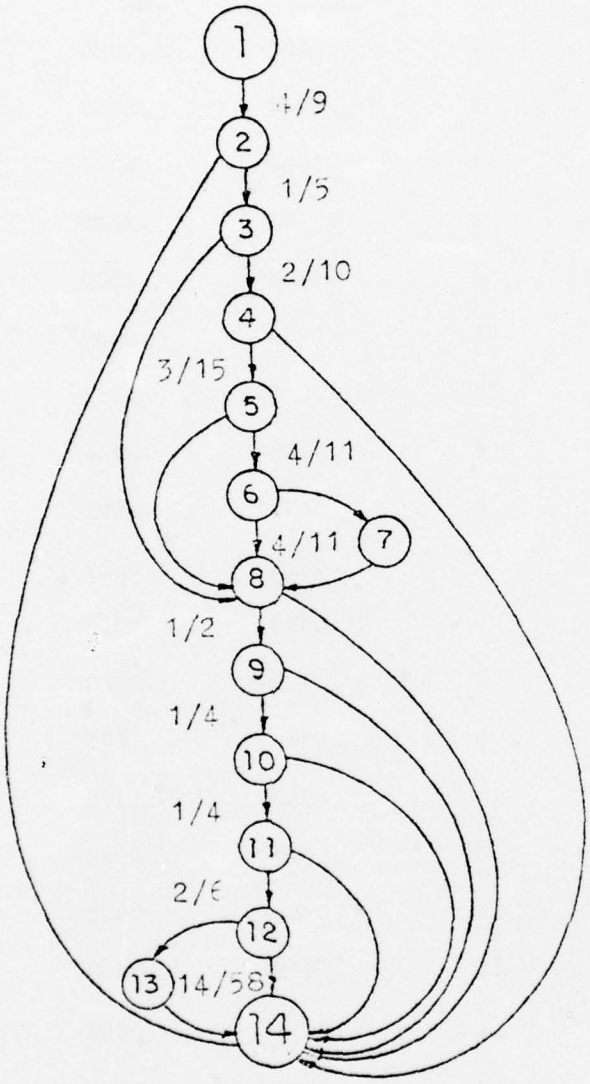
$E'(1)/U = (E'(1)/S)M$: mean fraction of number of errors detected on first input (given as a percentage at the bottom of directed graphs).

*: indicates that P'_{ij} has no meaning because ratio of number of traversals in arc ij to total number of input traversals is greater than one, because arc ij is part of a cycle.

2. Notes.

- The number of source statements (x) and number of machine instructions (y) in an arc are indicated by (x/y) alongside the arc. No number or zero means there are zero statements/instructions in an arc. (These are transfer of control arcs). In some cases the number of machine instructions was not available.
- Since the simulation model does not accomodate parallel arcs, a dummy node was inserted in each parallel arc.
- Nodes associated with sub-procedures (entry and exit nodes) are designated by letters. The entire sub-procedure is indicated by a dotted line and is counted as one arc.

Module: 1 Procedure No.: 2



Number of nodes:	14
Number of arcs:	23
Number of paths:	26
Number of source stmts.:	37
Average error found:	0.3144
Percentage errors found:	17.84

Module 1 Procedure 2

100 Replications 100 Repetitions

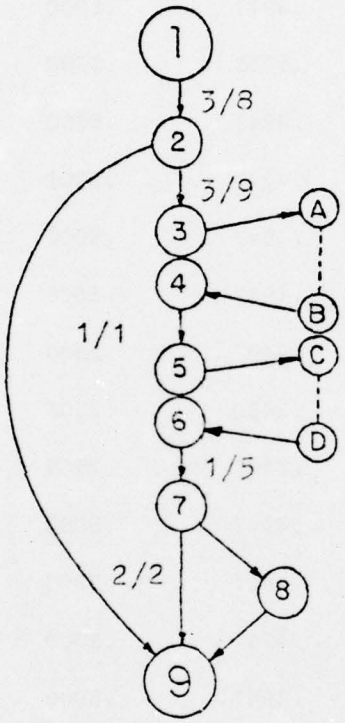
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	4
2	3	.4948	.5000	1
2	14	.5052	.5000	0
3	4	.2480	.2500	2
3	8	.2468	.2500	0
4	5	.1250	.1250	3
4	14	.1230	.1250	0
5	6	.0645	.0625	4
5	8	.0605	.0625	0
6	7	.0315	.0313	0
6	8	.0330	.0313	4
7	8	.0315	.0313	0
8	9	.1860	.1875	1
8	14	.1858	.1875	0
9	10	.0949	.0938	1
9	14	.0911	.0938	0
10	11	.0475	.0469	1
10	14	.0474	.0469	0
11	12	.0233	.0234	2
11	14	.0242	.0234	0
12	13	.0101	.0117	0
12	14	.0132	.0117	14
13	14	.0101	.0117	0
				37

$\sum_{ij} p_{ij} s_{ij} = 6.2890$

E(1) = .2995

E(1)/U = .1700

Module: 1 Procedure No.: 8



Number of nodes:	13
Number of arcs:	14
Number of paths:	3
Number of source stmts.:	10
Average error found:	0.2523
Percentage errors found:	52.98

Module 1 Procedure 8

100 Replications 100 Repetitions

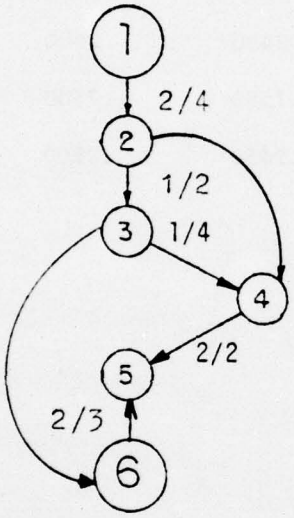
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	3
2	3	.4947	.5000	3
2	9	.5053	.5000	0
3	A	.4947	.5000	0
4	5	.4947	.5000	1
5	C	.4947	.5000	0
6	7	.4947	.5000	1
7	8	.2497	.2500	0
7	9	.2450	.2500	2
8	9	.2497	.2500	0
A	B	.4947	.5000	0
B	4	.4947	.5000	0
C	D	.4947	.5000	0
D	6	.4947	.5000	0
				10

$\sum_{ij} p_{ij} s_{ij} = 6.0000$

E(1) = .2857

E(1)/U = .6000

Module: 1 Procedure No.: 11



Number of nodes:	6
Number of arcs:	7
Number of paths:	3
Number of source stats.:	8
Average error found:	0.1974
Percentage errors found:	51.82

Module 1 Procedure 11

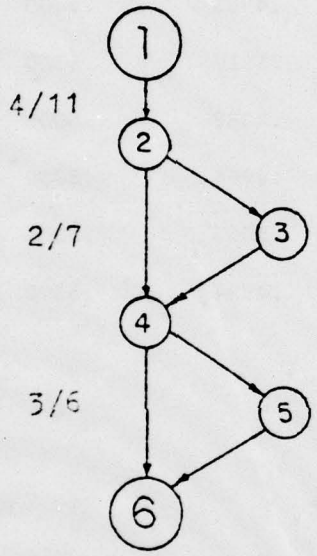
		100 Replications	100 Repetitions	
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4947	.5000	1
2	4	.5053	.5000	0
3	4	.2497	.2500	1
3	6	.2450	.2500	0
4	5	.7550	.7500	2
6	5	.2450	.2500	2
				8

$\sum_{ij,} p_{ij} s_{ij} = 4.75$

E(1) = .2262

E(1)/U = .5938

Module: 1 Procedure No.: 14



Number of nodes:	6
Number of arcs:	7
Number of paths:	4
Number of source stmts.:	9
Average error found:	0.2586
Percentage errors found:	60.34

Module 1 Procedure 14

100 Replications 100 Repetitions

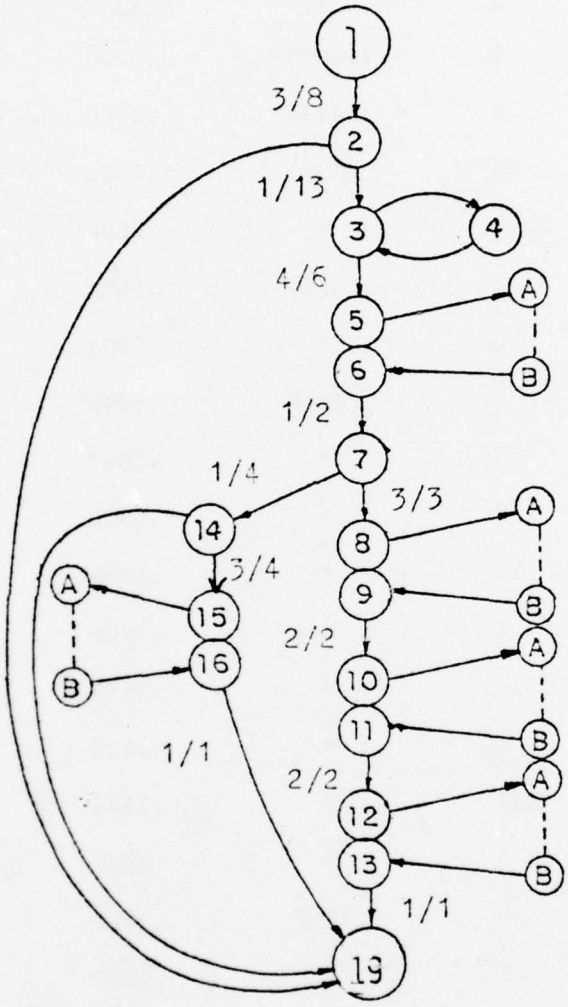
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	4
2	3	.4983	.5000	0
2	4	.5017	.5000	2
3	4	.4983	.5000	0
4	5	.4994	.5000	0
4	6	.5006	.5000	3
5	6	.4994	.5000	0
				9

$\sum_{ij} p_{ij} s_{ij} = 6.5000$

E(1) = .3095

E(1)/U = .7222

Module: 1 Procedure No.: 19



Number of nodes:	19
Number of arcs:	26
Number of paths:	7
Number of source stmts.:	22
Average error found:	0.2885
Percentage errors found:	27.54

Module 1 Procedure 19

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	3
2	3	.4948	.5000	1
2	19	.5052	.5000	0
3	4	*	.2500	0
3	5	.4948	.5000	4
4	3	*	.2500	0
5	A	.4948	.5000	0
6	7	*	.1667	1
7	8	*	.0909	3
7	14	*	.1000	1
8	A	*	.0909	0
9	10	*	.1538	2
10	A	*	.1538	0
11	12	*	.1538	2
12	A	*	.1538	0
13	19	.2206	.2222	1
14	15	*	.0500	3
14	19	.0560	.0556	0
15	A	*	.0500	0
16	19	.2182	.2222	1
A	B	*	.5000	0
B	6	*	.1667	0

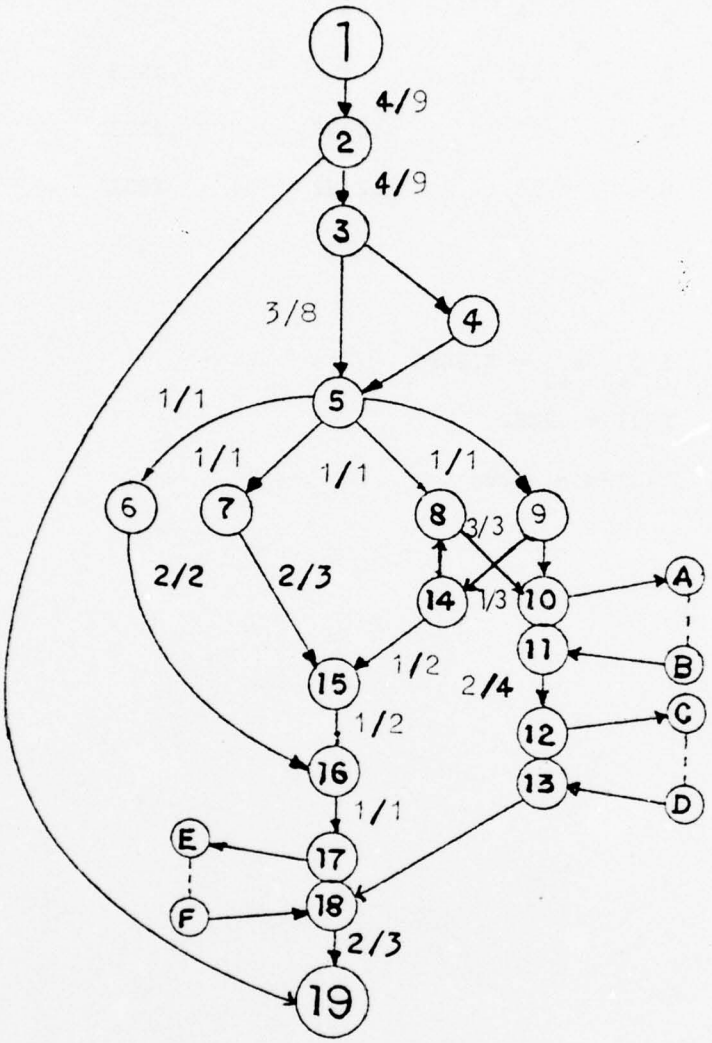
B	9	*	.1538	0
B	11	*	.1538	0
B	13	.2206	.2222	0
B	16	.2182	.2222	0
				22

$\Sigma\ p_{ij}\ s_{ij} = 7.2490$

$E(1) = .3452$

$E(1)/u = .3295$

Module: 1 Procedure No.: 22



Number of nodes:	25
Number of arcs:	30
Number of paths:	11
Number of source stmts.:	30
Average error found:	0.4105
Percentage errors found:	28.7%

Module 1 Procedure 22

100 Replications 100 Repetitions

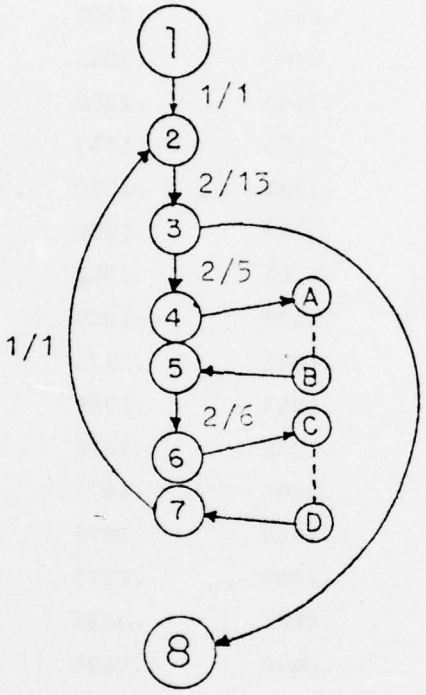
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	4
2	3	.4950	.5000	4
2	19	.5050	.5000	0
3	4	.2490	.2500	0
3	5	.2460	.2500	3
4	5	.2490	.2500	0
5	6	.1280	.1250	1
5	7	.1179	.1250	1
5	8	.1234	.1250	1
5	9	.1257	.1250	1
6	16	.1280	.1250	2
7	15	.1179	.1250	2
8	10	.1885	.1875	3
9	14	.1257	.1250	1
10	A	.1885	.1875	0
11	12	.1885	.1875	2
12	C	.1885	.1875	0
13	18	.1885	.1875	0
14	8	.0651	.0625	0
14	15	.0606	.0625	1
15	16	.1785	.1875	1
16	17	.3065	.3125	1
17	E	.3065	.3125	0
18	19	.4950	.5000	2
A	B	.1885	.1875	0
B	11	.1885	.1875	0
C	D	.1885	.1875	0
D	13	.1885	.1875	0
E	F	.3065	.3125	0
F	18	.3065	.3125	0
				30

$\sum_{ij} p_{ij} s_{ij} = 10.375$

E(1) = .4940

E(1)/U = .3458

Module: 1 Procedure No.: 25



Number of nodes:	12
Number of arcs:	12
Number of paths:	2
Number of source stmts.:	8
Average error found:	0.2324
Percentage errors found:	61.01

Module 1 Procedure 25

100 Replications 100 Repetitions

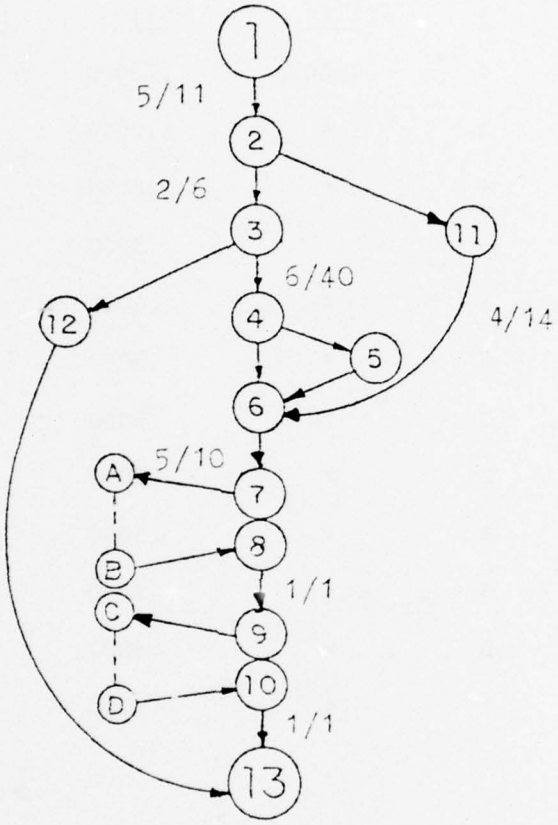
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	1
2	3	*	1.0000	2
3	4	*	.5000	2
3	8	*	.5000	0
4	A	*	.5000	0
5	6	*	.5000	2
6	C	*	.5000	0
7	2	*	.5000	1
A	B	*	.5000	0
B	5	*	.5000	0
C	D	*	.5000	0
D	7	*	.5000	0
				8

$\sum_{ij} p_{ij} s_{ij} = 5.5$

E(1) = .2619

E(1)/U = .6875

Module: 1 Procedure No.: 28



Number of nodes:	17
Number of arcs:	19
Number of paths:	4
Number of source stmts.:	24
Average error found:	0.6400
Percentage errors found:	56.00

Module 1 Procedure 28

100 Replications 100 Repetitions

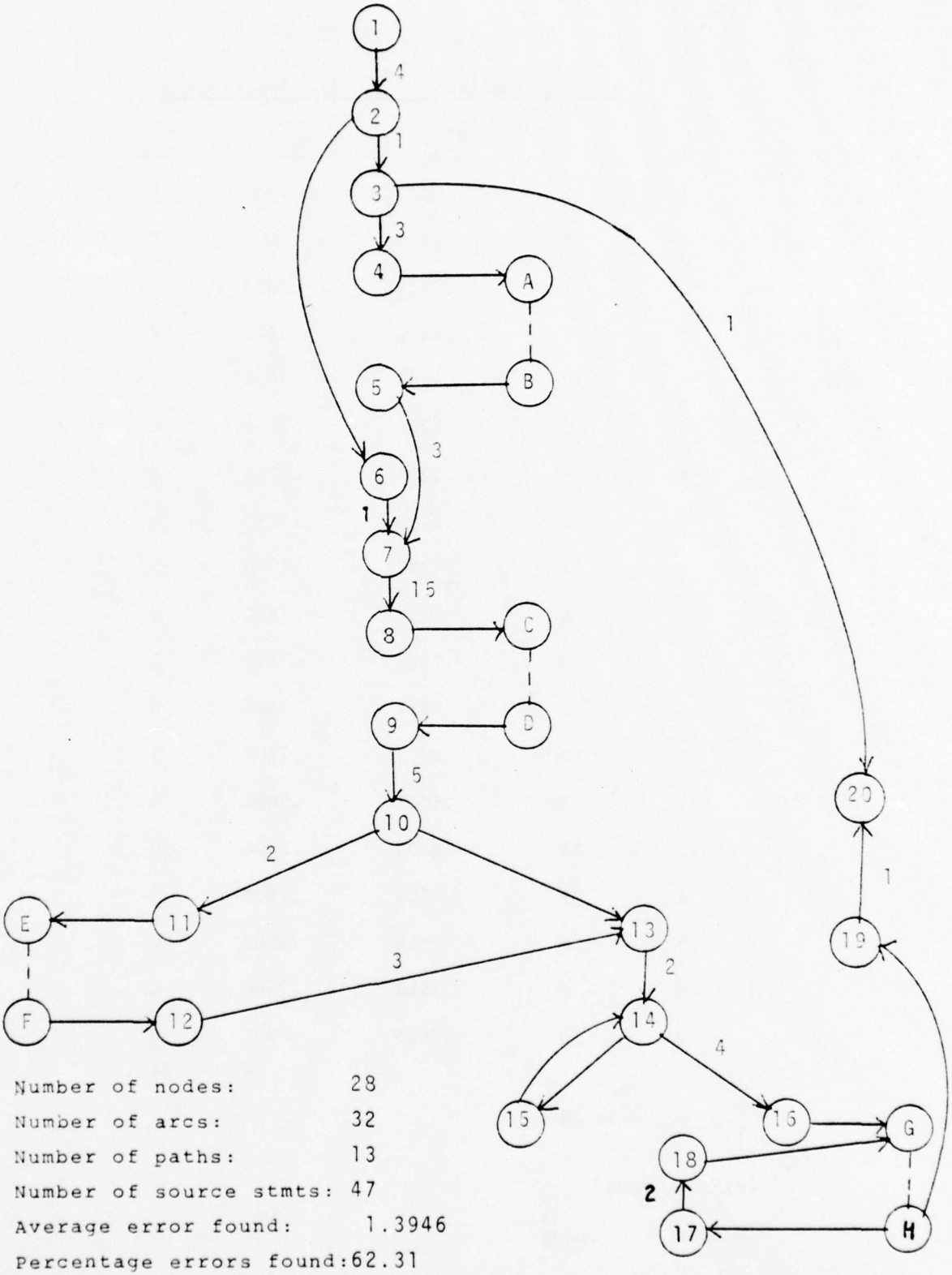
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	5
2	3	.4975	.5000	2
2	11	.5025	.5000	0
3	4	.2491	.2500	6
3	12	.2484	.2500	0
4	5	.1256	.1250	0
4	6	.1235	.1250	0
5	6	.1256	.1250	0
6	7	.7516	.7500	5
7	A	.7516	.7500	0
8	9	.7516	.7500	1
9	C	.7516	.7500	0
10	13	.7516	.7500	1
11	6	.5025	.5000	4
12	13	.2484	.2500	0
A	B	.7516	.7500	0
B	8	.7516	.7500	0
C	D	.7516	.7500	0
D	10	.7516	.7500	0
				24

$\sum_{ij} p_{ij} s_{ij} = 14.75$

E(1) = .7024

E(1)/U = .6145

Module: 1 Procedure No.: 29



Number of nodes: 28
Number of arcs: 32
Number of paths: 13
Number of source stmts: 47
Average error found: 1.3946
Percentage errors found: 62.31

Module 1 Procedure No. 29

		100 Replications	100 Repetitions	
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	4
2	3	.5014	.5000	
2	6	.4986	.5000	0
3	4	.2499	.2500	3
3	20	.2515	.2500	1
4	A	.2499	.2500	0
5	7	.2499	.2500	3
6	7	.4986	.5000	1
7	8	.7485	.7500	15
8	C	.7485	.7500	0
9	10	.7485	.7500	5
10	11	.3682	.3750	2
10	13	.3793	.3750	0
11	E	.3692	.3750	0
12	13	.3692	.3750	3
13	14	.7485	.7500	2
14	15	*	.3750	0
14	16	.7485	.7500	4
15	14	*	.3750	0
16	G	.7485	.7500	0
17	18	.7445	.7500	2
18	G	.7445	.7500	0
19	20	.7485	.7500	1
A	B	.2499	.2500	0
B	5	.2499	.2500	0
C	D	.7485	.7500	0
D	9	.7485	.7500	0
E	F	.3692	.3750	0
F	12	.3692	.3750	0
G	H	*	.7500	0
H	17	.7445	.7500	0
H	19	.7485	.7500	0
				47

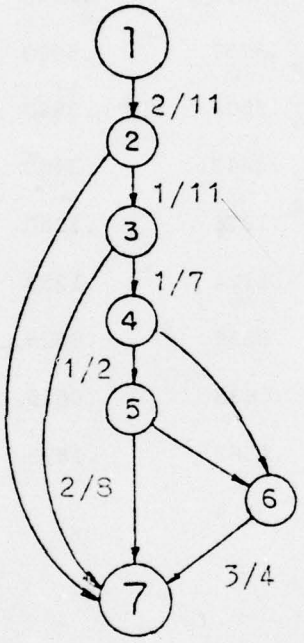
(continued)

$$\sum_{ij} p_{ij} S_{ij} = 30.375$$

$E(1) = 1.4464$

$E(1)/U = .6463$

Module: 1 Procedure No.: 30



Number of nodes:	7
Number of arcs:	10
Number of paths:	5
Number of source stmts.:	10
Average error found:	0.1649
Percentage errors found:	34.63

Module 1 Procedure 30

100 Replications 100 Repetitions

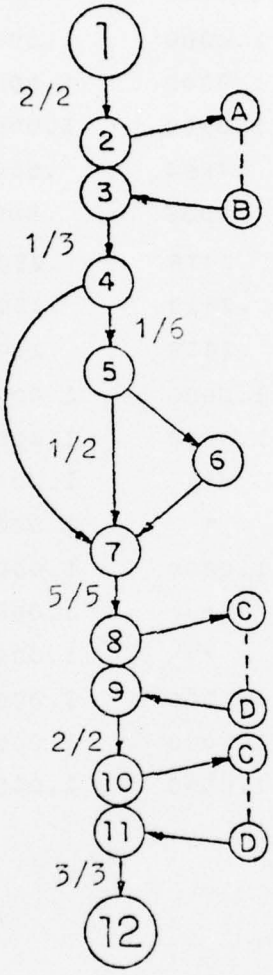
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4943	.5000	1
2	7	.5057	.5000	0
3	4	.2500	.2500	1
3	7	.2443	.2500	0
4	5	.1276	.1250	1
4	6	.1224	.1250	0
5	6	.0638	.0625	0
5	7	.0638	.0625	2
6	7	.1862	.1875	3
				10

$\sum_{ij} p_{ij} s_{ij} = 3.5625$

E(1) = .1696

E(1)/U = .3563

Module: 1 Procedure No.: 34



Number of nodes:	16
Number of arcs:	18
Number of paths:	3
Number of source stmts.:	15
Average error found:	0.5465
Percentage errors found:	76.51

Module 1 Procedure No. 34

100 Replications 100 Repetitions

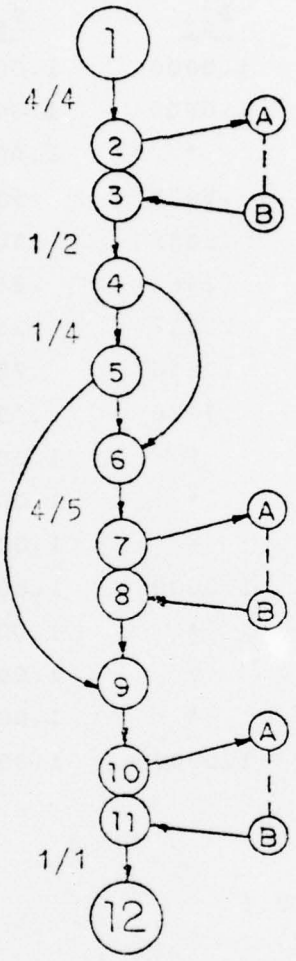
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	A	1.0000	1.0000	0
3	4	1.0000	1.0000	1
4	5	.4994	.5000	1
4	7	.5006	.5000	0
5	6	.2475	.2500	0
5	7	.2519	.2500	1
6	7	.2475	.2500	0
7	8	1.0000	1.0000	5
8	C	1.0000	1.0000	0
9	10	*	1.0000	2
10	C	*	1.0000	0
11	12	1.0000	1.0000	3
C	D	*	1.0000	0
D	9	*	1.0000	0
D	11	1.0000	1.0000	0
A	B	1.0000	1.0000	0
B	3	1.0000	1.0000	0
				15

$\sum_{ij} p_{ij} s_{ij} = 13.75$

E(1) = .6548

E(1)/U = .9167

Module: 1 Procedure No.: 35



Number of nodes:	14
Number of arcs:	17
Number of paths:	3
Number of source stmts.:	11
Average error found:	0.3576
Percentage errors found:	68.27

Module 1 Procedure No. 35

100 Replications 100 Repetitions

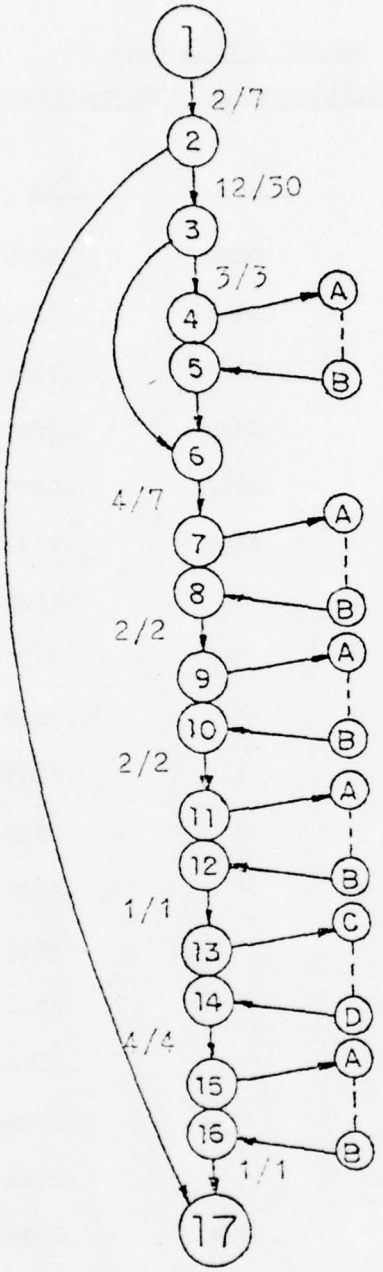
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	4
2	A	1.0000	1.0000	0
3	4	*	1.0000	1
4	5	.5075	.5000	1
4	6	.5055	.5000	0
5	6	.2563	.2500	0
5	9	.2512	.2500	0
6	7	.7618	.7500	4
7	A	.7618	.7500	0
8	9	*	1.0000	0
9	10	*	1.0000	0
10	A	*	1.0000	0
11	12	1.0000	1.0000	1
A	B	*	1.0000	0
B	3	*	1.0000	0
B	8	*	1.0000	0
B	11	1.0000	1.0000	0
				11

$\sum_{ij} p_{ij} s_{ij} = 9.5$

E(1) = .4524

E(1)/U = .8636

Module: 1 Procedure No.: 36



Number of nodes:	21
Number of arcs:	26
Number of paths:	3
Number of source stmts.:	31
Average error found:	0.5203
Percentage errors found:	35.25

Module 1 Procedure 36

100 Replications 100 Repetitions

		p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4964	.5000	12
2	17	.5036	.5000	0
3	4	.2449	.2500	3
3	6	.2515	.2500	0
4	A	,2449	.2500	0
5	6	*	.2500	0
6	7	*	.2500	4
7	A	*	.2500	0
8	9	*	.2500	2
9	A	*	.2500	0
10	11	*	.2500	2
11	A	*	.2500	0
12	13	*	.2500	1
13	C	*	.2500	0
14	15	*	.2500	4
15	A	*	.2500	0
16	17	.4964	.5000	1
A	B	*	.5000	0
B	5	*	.2500	0
B	8	*	.2500	0
B	10	*	.2500	0
B	12	*	.2500	0
B	16	.4964	.5000	0

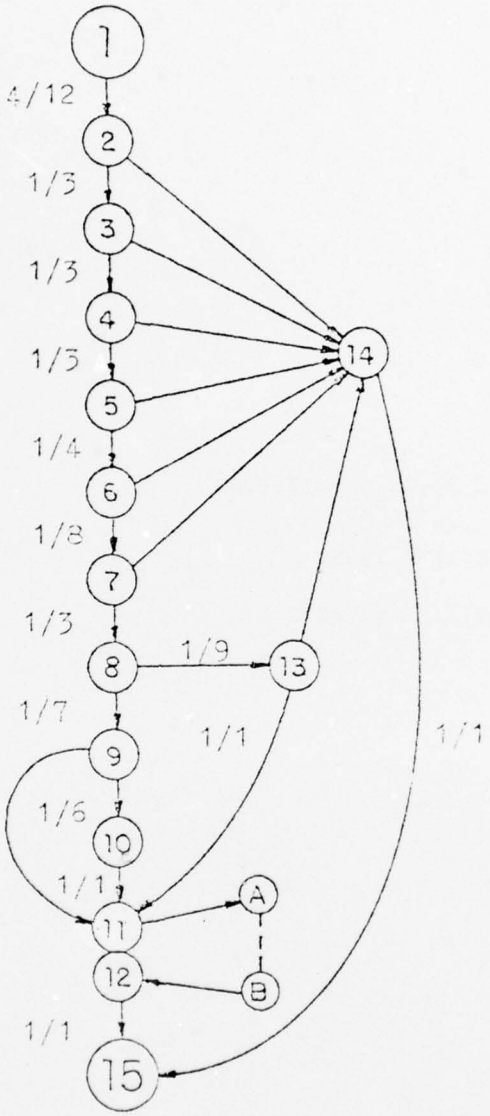
C	D	*	.2500	0
D	14	*	.2500	0
				31

$\Sigma P_{ij} S_{ij} = 12.5000$

$E(1) = .5952$

$E(1)/u = .4032$

Module: 1 Procedure No.: 39



Number of nodes:	17
Number of arcs:	25
Number of paths:	10
Number of source stmts.:	17
Average error found:	0.2637
Percentage errors found:	32.57

Module 1 Procedure 39

100 Replications 100 Repetitions

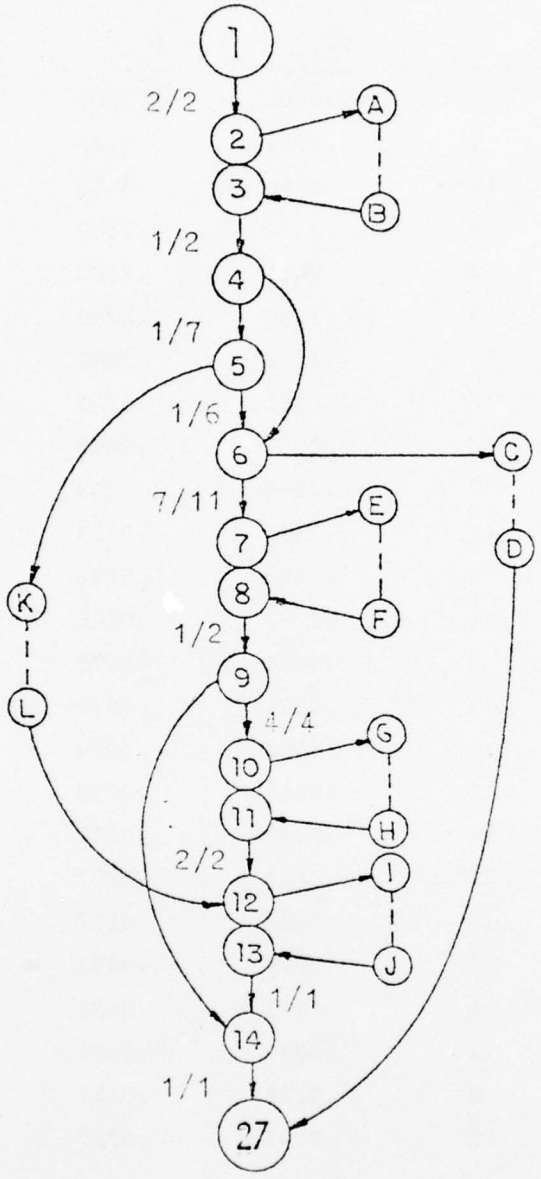
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	4
2	3	.4954	.5000	1
2	14	.5046	.5000	0
3	4	.2490	.2500	1
3	14	.2464	.2500	0
4	5	.1258	.1250	1
4	14	.1232	.1250	0
5	6	.0631	.0625	1
5	14	.0627	.0625	0
6	7	.0299	.0313	1
6	14	.0332	.0313	0
7	8	.0162	.0156	1
7	14	.0137	.0156	0
8	9	.0090	.0078	1
8	13	.0072	.0078	1
9	10	.0040	.0039	1
9	11	.0050	.0039	0
10	11	.0040	.0039	1
11	A	.0123	.0117	0
12	15	.0123	.0117	1
14	15	.9877	.9883	1
13	11	.0033	.0039	1
13	14	.0039	.0039	0
A	B	.0123	.0117	0
B	12	.0123	.0117	0
				17

$\sum_{ij} p_{ij} s_{ij} = 6.0117$

E(1) = .2863

E(1)/U = .3536

Module: 1 Procedure No.: 44



Number of nodes:	27
Number of arcs:	30
Number of paths:	7
Number of source stmts.:	21
Average error found:	0.3554
Percentage errors found:	35.54

Module 1 Procedure 44

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	A	1.0000	1.0000	0
3	4	1.0000	1.0000	1
4	5	.4877	.5000	1
4	6	.5123	.5000	0
5	6	.2458	.2500	1
5	K	.2419	.2500	0
6	7	.3846	.3750	7
6	C	.3735	.3750	0
7	E	.3946	.3750	0
8	9	.3846	.3750	1
9	10	.1952	.1875	4
9	14	.1894	.1875	0
10	G	.1952	.1875	0
11	12	.1952	.1875	2
12	I	.4371	.4375	0
13	14	.4371	.4375	1
14	27	.6265	.6250	1
G	H	.1952	.1875	0
H	11	.1952	.1875	0
A	B	1.0000	1.0000	0
B	3	1.0000	1.0000	0
C	D	.3735	.3750	0
D	27	.3735	.3750	0

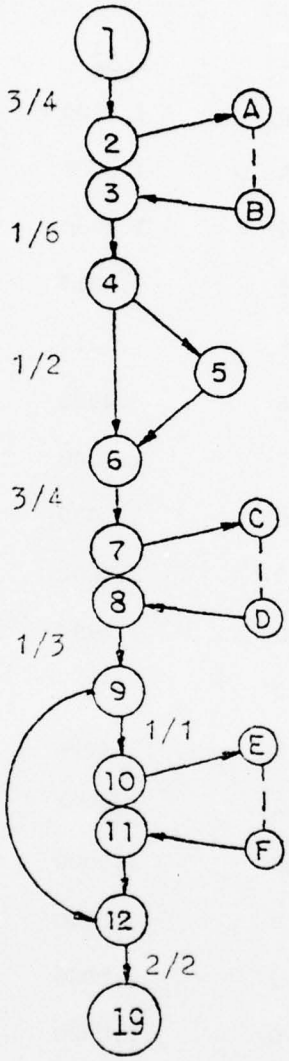
E	F	.3735	.3750	0
F	8	.3846	.3750	0
I	J	.4371	.4375	0
J	13	.4371	.4375	0
K	L	.2419	.2500	0
L	12	.2419	.2500	0
				21

$\Sigma\ p_{ij}\ s_{ij} = 8.9375$

E(1) = .4256

E(1)/u = .4256

Module: 1 Procedure No.: 47



Number of nodes:	19
Number of arcs:	20
Number of paths:	4
Number of source stmts.:	12
Average error found:	0.4231
Percentage errors found:	74.04

Module 1 Procedure 47

100 Replications 100 Repetitions

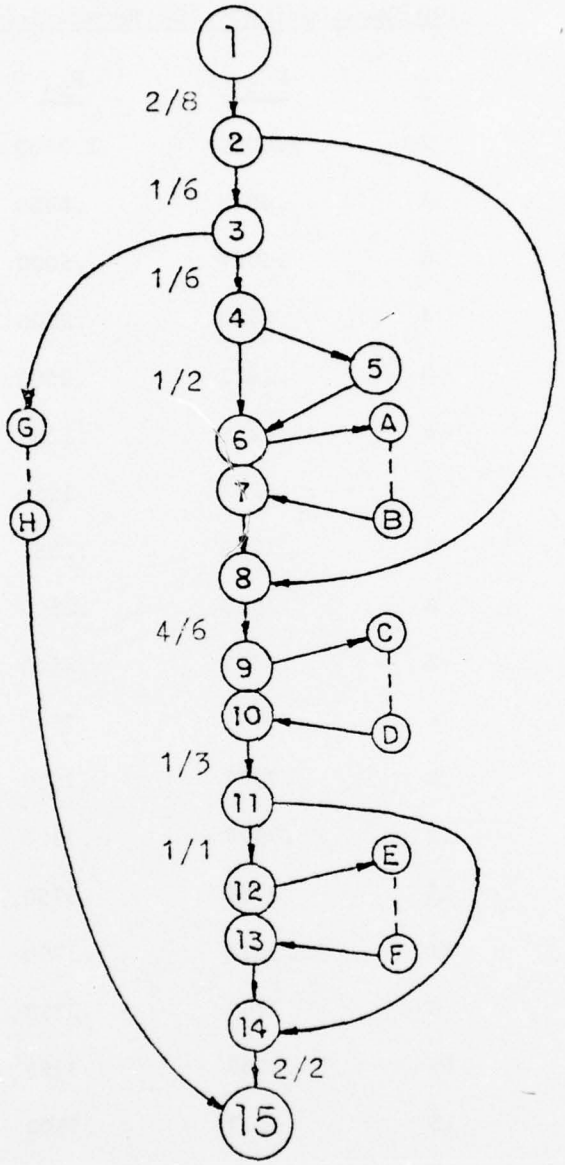
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	3
2	A	1.0000	1.0000	0
3	4	1.0000	1.0000	1
4	5	.4913	.5000	0
4	6	.5017	.5000	1
5	6	.4983	.5000	3
6	7	1.0000	1.0000	0
7	C	1.0000	1.0000	1
8	9	1.0000	1.0000	1
9	10	.4994	.5000	0
9	12	.5006	.5000	0
10	E	.4994	.5000	0
11	12	.4994	.5000	2
12	19	1.0000	1.0000	0
A	B	1.0000	1.0000	0
B	3	1.0000	1.0000	0
C	D	1.0000	1.0000	0
D	8	1.0000	1.0000	0
E	F	.4994	.5000	0
F	11	.4994	.5000	0
				12

$\Sigma\ p_{ij}\ s_{ij} = 9.0000$

$E(1) = .4286$

$E(1)/u = .7500$

Module: 1 Procedure No.: 48



Number of nodes:	23
Number of arcs:	26
Number of paths:	7
Number of source stmts.:	13
Average error found:	0.3287
Percentage errors found:	53.10

Module 1 Procedure 48

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4943	.5000	1
2	8	.5057	.5000	0
3	4	.2520	.2500	1
3	G	.2423	.2500	0
4	5	.1288	.1250	0
4	6	.1232	.1250	1
5	6	.1288	.1250	0
6	A	.2520	.2500	0
7	8	.2520	.2500	0
8	9	.7577	.7500	4
9	C	.7577	.7500	0
10	11	.7577	.7500	1
11	12	.3766	.3750	1
11	14	.3811	.3750	0
12	E	.3766	.3750	0
13	14	.3766	.3750	0
14	15	.7577	.7500	2
A	B	.2520	.2500	0
B	7	.2520	.2500	0
C	D	.7577	.7500	0
D	10	.7577	.7500	0
E	F	.3766	.3750	0

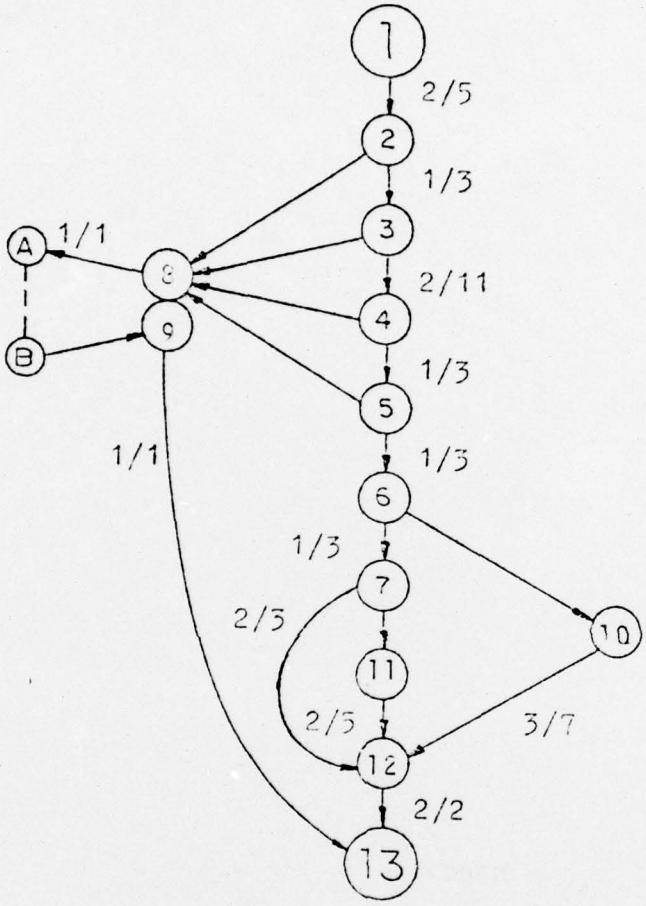
F	13	.3766	.3750	0
G	H	.2423	.2500	0
H	15	.2423	.2500	0
				13

$\Sigma\ p_{ij}\ s_{ij} = 8.5000$

E (1) = .4048

E(1)/u = .6538

Mcdule: 1 Procedure No.: 49



Number of nodes:	15
Number of arcs:	20
Number of paths:	7
Number of source stmts.:	19
Average error found:	0.2217
Percentage errors found:	24.50

Module 1 Procedure 49

100 Replications 100 Repetitions

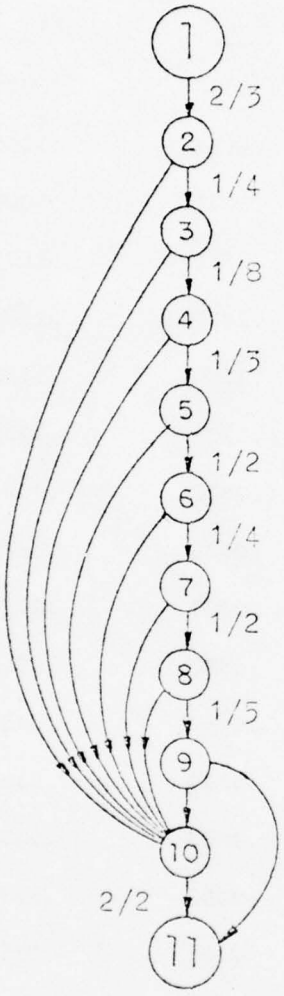
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4959	.5000	1
2	8	.5041	.5000	0
3	4	.2492	.2500	2
3	8	.2467	.2500	0
4	5	.1260	.1250	1
4	8	.1232	.1250	0
5	6	.0632	.0625	1
5	8	.0628	.0625	0
6	7	.0302	.0313	1
6	10	.0330	.0313	0
7	11	.0164	.0156	0
7	12	.0138	.0156	2
8	A	.9368	.9375	1
9	13	.9368	.9375	1
10	12	.0330	.0313	3
11	12	.0164	.0156	2
12	13	.0632	.0625	2
A	B	.9368	.9375	0
B	9	.9368	.9375	0
				19

$\Sigma\ p_{ij}\ s_{ij} = 5.3751$

$E(1) = .2560$

$E(1)/u = .2829$

Module: 1 Procedure No.: 53



Number of nodes:	11
Number of arcs:	18
Number of paths:	9
Number of source stmts.:	11
Average error found:	0.1876
Percentage errors found:	35.81

Module 1 Procedure No. 53

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4946	.5000	1
2	10	.5054	.5000	0
3	4	.2486	.2500	1
3	10	.2460	.2500	0
4	5	.1258	.1250	1
4	10	.1228	.1250	0
5	6	.0632	.0625	1
5	10	.0626	.0625	0
6	7	.0299	.0313	1
6	10	.0333	.0313	0
7	8	.0161	.0156	1
7	10	.0138	.0156	0
8	9	.0092	.0078	1
8	10	.0069	.0078	0
9	10	.0040	.0039	0
9	11	.0052	.0039	0
10	11	.9948	.9961	2
				11

$$\sum_{ij} p_{ij} s_{ij} = 4.9844$$

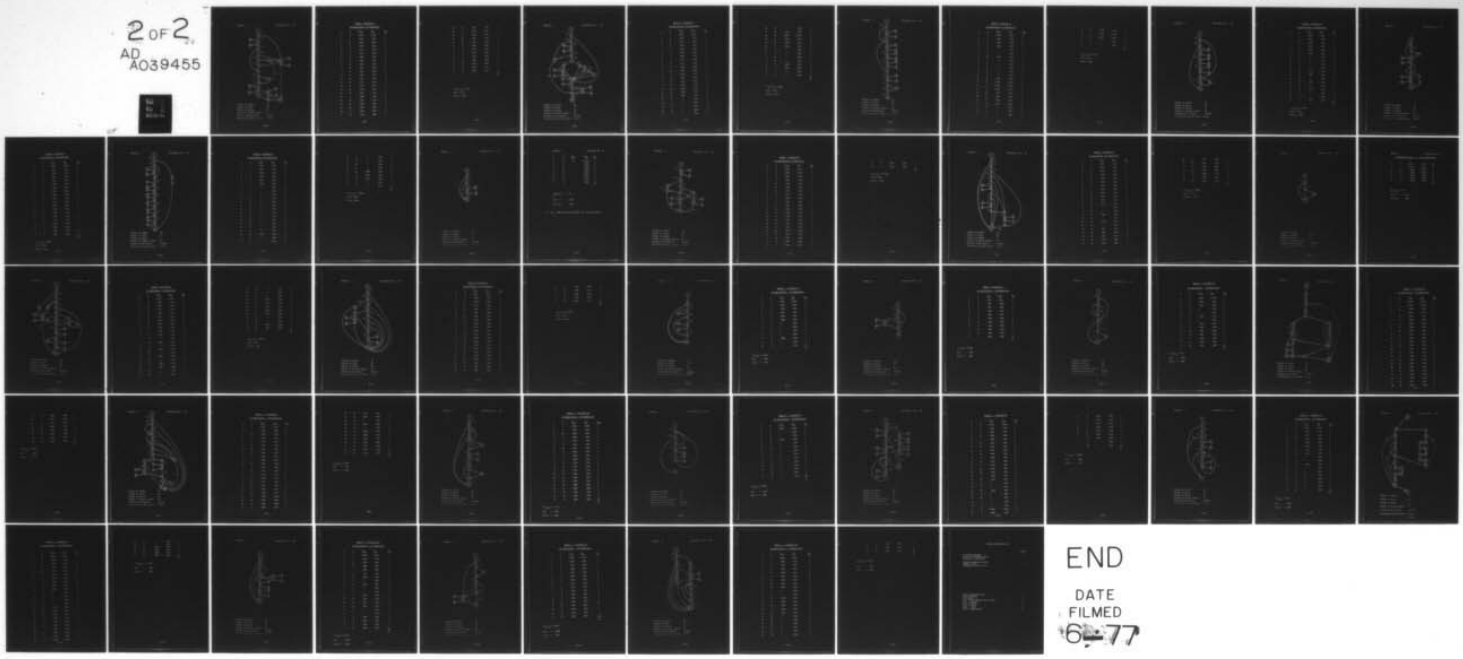
$E(1) = .2374$

$E(1)/U = .4531$

AD-A039 455 NAVAL POSTGRADUATE SCHOOL MONTEREY CALIF F/G 9/2
SOFTWARE ERROR DETECTION MODELS, VALIDATION TESTS AND PROGRAM C--ETC(U)
NOV 76 N F SCHNEIDEWIND, G T HOWARD

UNCLASSIFIED NPS-52SS76111 NL

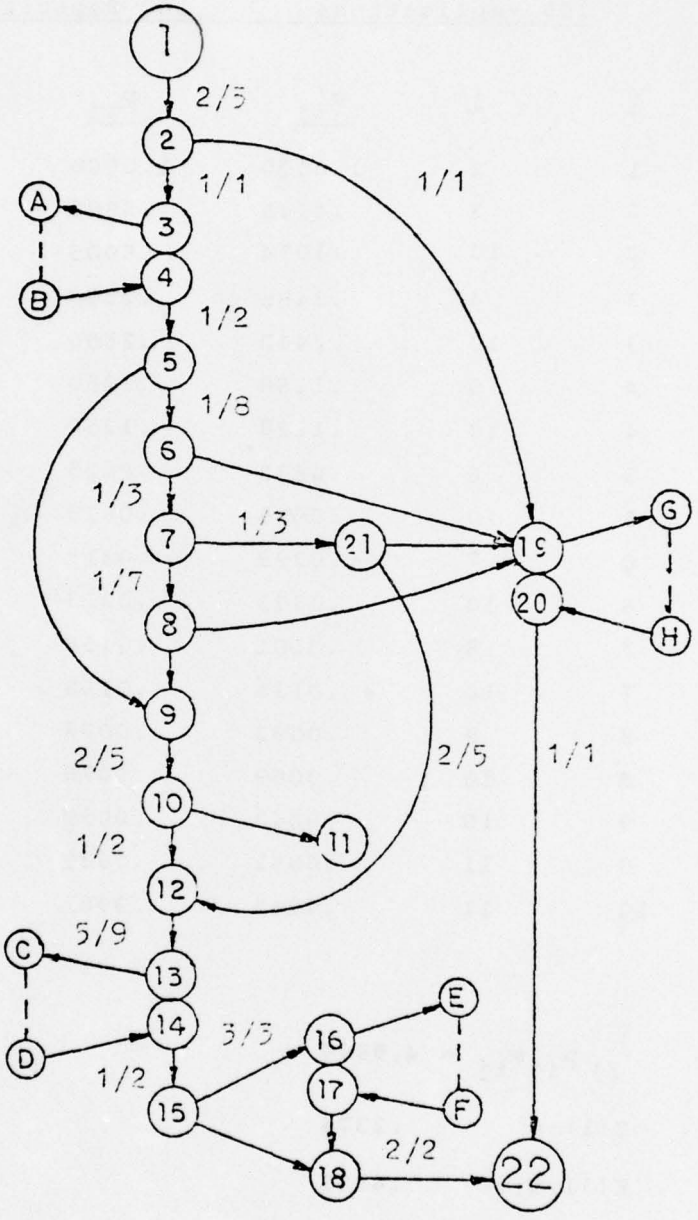
2 OF 2
AD A039455



END

DATE FILMED
6-77

Module: 1 Procedure No.: 57



Number of nodes:	30
Number of arcs:	35
Number of paths:	12
Number of source stmts.:	26
Average error found:	0.2910
Percentage errors found:	23.50

Module 1 Procedure 57

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4972	.5000	1
2	19	.5028	.5000	1
3	A	.4972	.5000	0
4	5	.4972	.5000	1
5	6	.2511	.2500	1
5	9	.2461	.2500	0
6	7	.1228	.1250	1
6	19	.1283	.1250	0
7	8	.0621	.0625	1
7	21	.0607	.0625	1
8	9	.0309	.0313	0
8	19	.0312	.0313	0
9	10	.2770	.2813	2
10	11	.1407	.1407	0
10	12	.1363	.1407	1
12	13	.1644	.1720	5
13	C	.1644	.1720	0
14	15	.1644	.1720	1
15	16	.0842	.0860	3
15	18	.0802	.0860	0
16	E	.0842	.0860	0
17	18	.0842	.0860	0

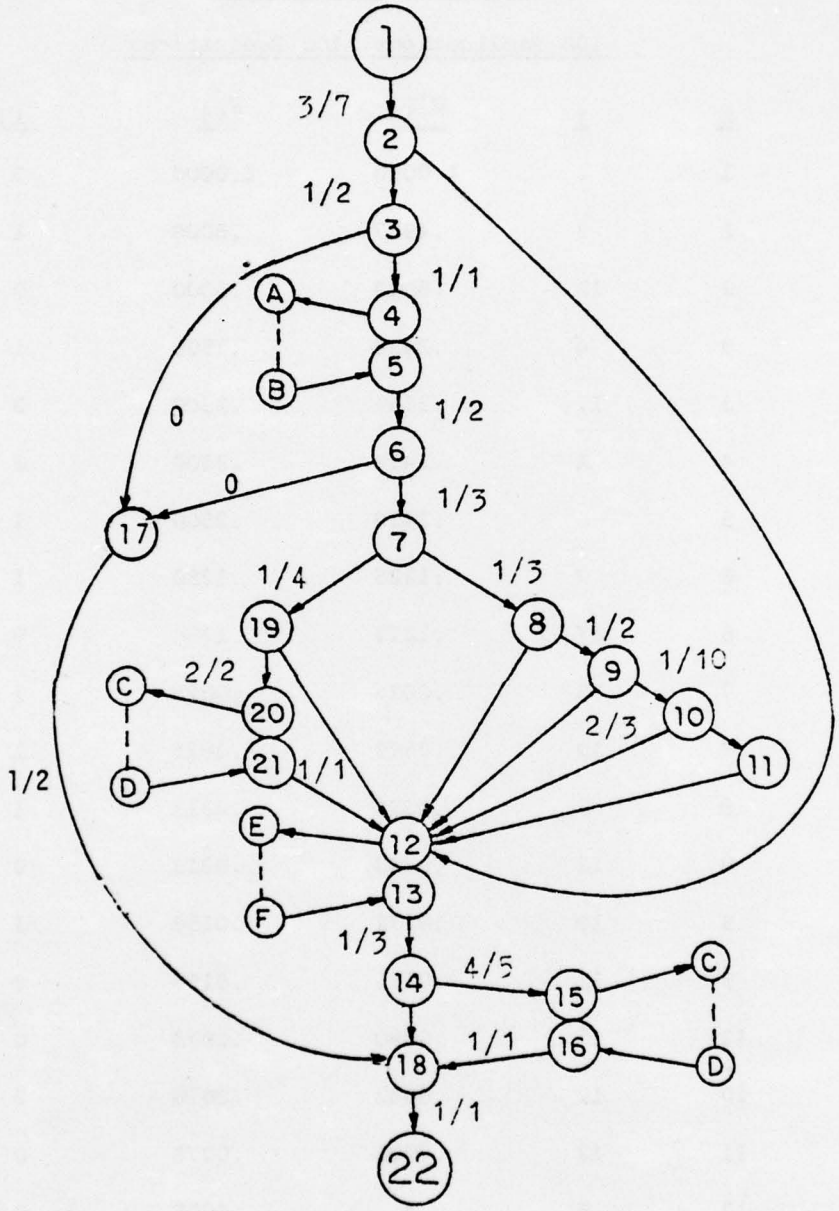
18	22	.1644	.1720	2
19	G	.6949	.6875	0
20	22	.6949	.6875	1
21	12	.0281	.0313	2
21	19	.0326	.0313	0
E	F	.0842	.0860	0
F	17	.0842	.0860	0
A	B	.4972	.5000	0
B	4	.4972	.5000	0
G	H	.6949	.6875	0
H	20	.6949	.6875	0
C	D	.1644	.1720	0
D	14	.1644	.1720	0
				26

$\Sigma\ p_{ij}\ s_{ij} = 7.0874$

$E(1) = .3375$

$E(1)/u = .2726$

Mcdule: 1 Procedure No.: 60



Number of nodes:	28
Number of arcs:	37
Number of paths:	16
Number of source stmts.:	24
Average error found:	0.3336
Percentage errors found:	29.19

Module 1 Procedure 60

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	3
2	3	.4987	.5000	1
2	12	.5013	.5000	0
3	4	.2438	.2500	1
3	17	.2549	.2500	0
4	A	.2438	.2500	0
5	6	.2438	.2500	1
6	7	.1225	.1250	1
6	17	.1213	.1250	0
7	8	.0626	.0625	1
7	19	.0599	.0625	1
8	9	.0303	.0313	1
8	12	.0323	.0313	0
9	10	.0172	.0156	1
9	12	.0131	.0156	0
10	11	.0090	.0078	0
10	12	.0082	.0078	2
11	12	.0090	.0078	0
12	E	*	.6095	0
13	14	*	.6095	1
14	15	*	.3047	4
14	18	*	.4063	0
15	C	*	.3047	0

16	18	*	.2188	1
17	18	.3762	.3750	1
18	22	1.0000	1.0000	1
19	12	.0295	.0313	0
19	20	.0304	.0313	2
20	C	.0304	.0313	0
21	12	*	.1641	1
C	D	*	.3282	0
D	16	*	.2188	0
D	21	*	.1461	0
A	B	.2438	.2500	0
B	5	.2438	.2500	0
E	F	*	.6095	0
F	13	*	.6095	0
				24

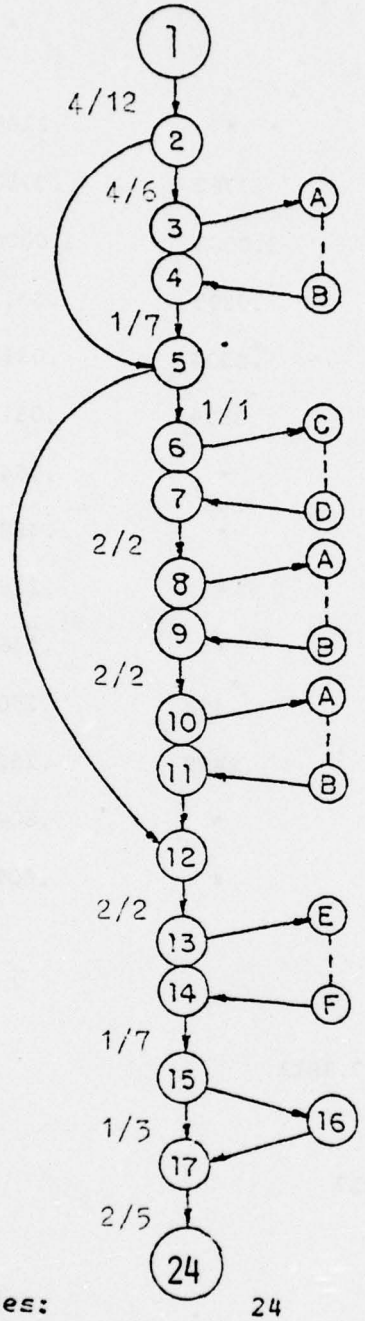
$\sum p_{ij} s_{ij} = 7.9613$

E(1) = .3791

E(1)/u = .3317

Module: 1

Procedure No.: 75



Number of nodes:	24
Number of arcs:	28
Number of paths:	8
Number of source stmts.:	20
Average error found:	0.5433
Percentage errors found:	57.05

Module 1 Procedure 75

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	4
2	3	.5020	.5000	4
2	5	.4980	.5000	0
3	A	.5020	.5000	0
4	5	*	.3750	1
5	6	*	.3750	1
5	12	.5066	.5000	0
6	C	*	.3750	0
7	8	*	.3750	2
8	A	*	.3750	0
9	10	*	.3000	2
10	A	*	.3000	0
11	12	.4934	.5000	0
12	13	1.0000	1.0000	2
13	E	1.0000	1.0000	0
14	15	1.0000	1.0000	1
15	16	.4937	.5000	0
15	17	.5063	.5000	1
16	17	.4937	.5000	0
17	24	1.0000	1.0000	2
A	B	*	.7500	0
B	4	*	.3750	0
B	9	*	.3000	0

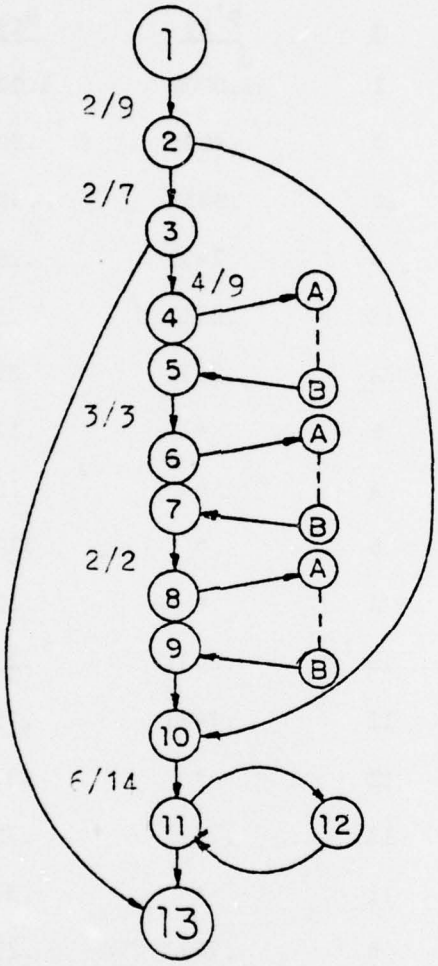
B	11	.4934	.5000	0
E	F	1.0000	1.0000	0
F	14	1.0000	1.0000	0
C	D	*	.3750	0
D	7	*	.3750	0
				20

$\Sigma\ p_{ij}\ s_{ij} = 13.6000$

E(1) = .6476

E(1)/u = .6800

Module: 1 Procedure No.: 76



Number of nodes:	15
Number of arcs:	19
Number of paths:	5
Number of source stmts.:	19
Average error found:	0.3893
Percentage errors found:	43.03

Module 1 Procedure 76

100 Replications 100 Repetitions

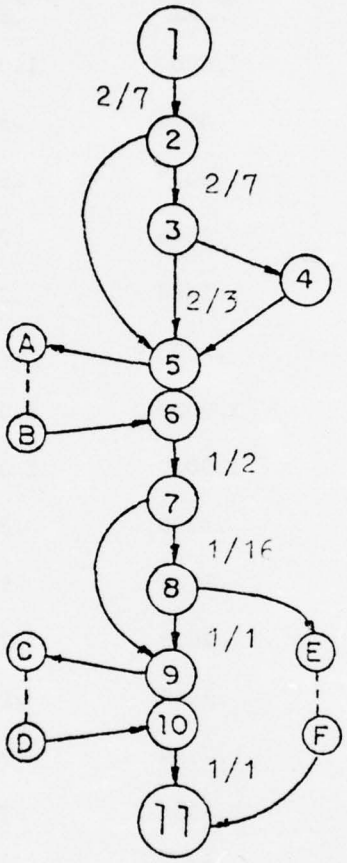
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4941	.5000	2
2	10	.5059	.5000	0
3	4	.2431	.2500	4
3	13	.2510	.2500	0
4	A	.2431	.2500	0
5	6	*	.1250	3
6	A	*	.1250	0
7	8	*	.1250	2
8	A	*	.1250	0
9	10	.2431	.2500	0
10	11	.7490	.7500	6
11	12	*	.3750	0
11	13	.7490	.7500	0
12	11	*	.3750	0
A	B	.7493	.2500	0
B	5	*	.1250	0
B	7	*	.1250	0
B	9	.2431	.2500	0
				19

$\Sigma\ p_{ij}\ s_{ij} = 9.1250$

E(1) = .4345

E(1)/u = .4803

Module: 1 Procedure No.: 77



Number of nodes:	17
Number of arcs:	20
Number of paths:	9
Number of source stmts.:	10
Average error found:	0.2425
Percentage errors found:	50.93

Module 1 Procedure 77

100 Replications 100 Repetitions

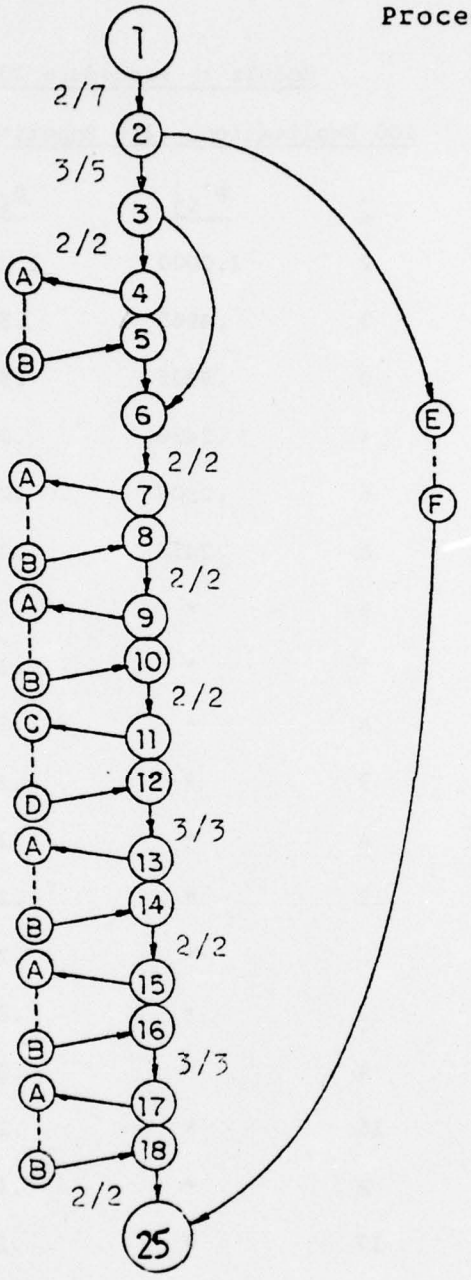
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4972	.5000	2
2	5	.5028	.5000	0
3	4	.2506	.2500	0
3	5	.2466	.2500	2
4	5	.2506	.2500	0
5	A	1.0000	1.0000	0
6	7	1.0000	1.0000	1
7	8	.4979	.5000	1
7	9	.5021	.5000	0
8	9	.2513	.2500	1
8	E	.2466	.2500	0
9	C	.7534	.7500	0
10	11	.7534	.7500	1
C	D	.7534	.7500	0
D	10	.7534	.7500	0
A	B	1.0000	1.0000	0
B	6	1.0000	1.0000	0
E	F	.2466	.2500	0
F	11	.2466	.2500	0
				10

$\Sigma\ p_{ij}\ s_{ij} = 6.0000$

E(1) = .2857

E(1)/u = .6000

Module: 1 Procedure No.: 79



Number of nodes:	25
Number of arcs:	31
Number of paths:	3
Number of source stmts.:	23
Average error found:	0.3628
Percentage errors found:	33.13

Module 1 Procedure 79

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4965	.5000	3
2	E	.5035	.5000	0
3	4	.2458	.2500	2
3	6	.2507	.2500	0
4	A	.2458	.2500	0
5	6	*	.2500	0
6	7	*	.3750	2
7	A	*	.3750	0
8	9	*	.2500	2
9	A	*	.2500	0
10	11	*	.2500	2
11	C	*	.2500	0
12	13	*	.2500	3
13	A	*	.2500	0
14	15	*	.2500	2
15	A	*	.2500	0
16	17	*	.2500	3
17	A	*	.2500	0
18	25	.4965	.5000	2
A	B	*	.5000	0
B	5	*	.2500	0

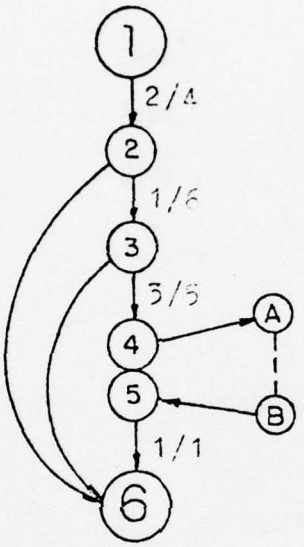
B	8	*	.2500	0
B	10	*	.2500	0
B	14	*	.2500	0
B	16	*	.2500	0
B	18	.4965	.5000	0
E	F	.5035	.5000	0
F	25	.5035	.5000	0
C	D	*	.2500	0
D	12	*	.2500	0
				23

$\Sigma\ p_{ij}\ s_{ij} = 8.7500$

$E(1) = .4167$

$E(1)/u = .3804$

Module: 1 Procedure No.: 81



Number of nodes:	8
Number of arcs:	9
Number of paths:	3
Number of source stmts.:	7
Average error found:	0.1449
Percentage errors found:	43.47

Module 1 Procedure No. 81

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	(1)	1.0000	2
2	3		.5000	1
2	6		.5000	0
3	4		.2500	3
3	6		.2500	0
4	A		.2500	0
A	B		.2500	0
B	5		.2500	0
5	6		.2500	1
				7

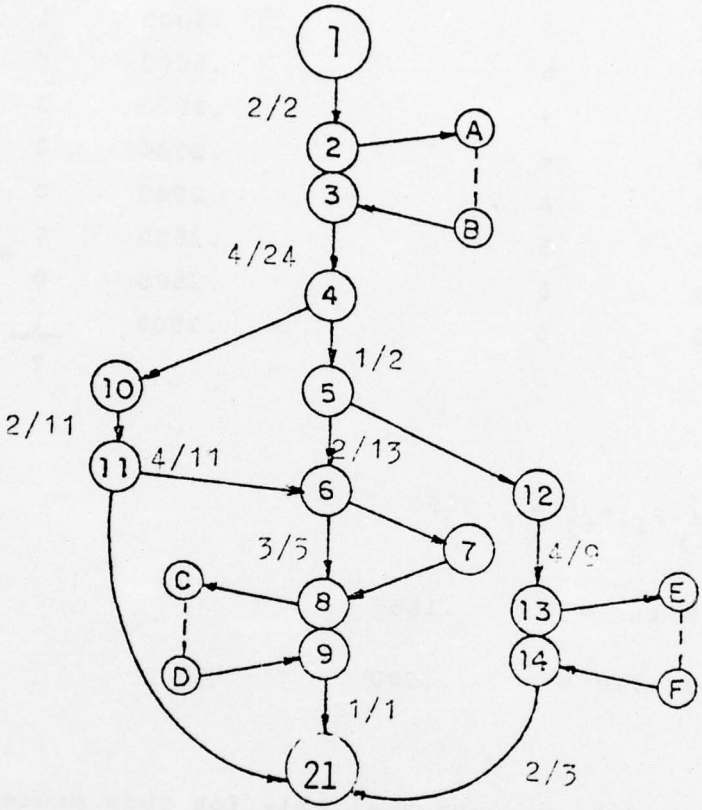
$\sum_{ij} p_{ij} s_{ij} = 3.50$

E(1) = .1667

E(1)/U = .5000

(1) p'_{ij} data was not available for this procedure.

Module: 1 Procedure No.: 87



Number of nodes:	21
Number of arcs:	24
Number of paths:	6
Number of source stmts.:	25
Average error found:	0.5029
Percentage errors found:	42.24

Module 1 Procedure 87

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	A	1.0000	1.0000	0
3	4	1.0000	1.0000	4
4	5	.4915	.5000	1
4	10	.5085	.5000	0
5	6	.2438	.2500	2
5	12	.2477	.2500	0
6	7	.2572	.2500	0
6	8	.2444	.2500	3
7	8	.2572	.2500	0
8	C	.5016	.5000	0
9	21	.5016	.5000	1
10	11	.5085	.5000	2
11	6	.2578	.2510	4
11	21	.2507	.2500	0
12	13	.2477	.2500	4
13	E	.2477	.2500	0
14	21	.2477	.2500	2
E	F	.2477	.2500	0
F	14	.2477	.2500	0
A	B	1.0000	1.0000	0
B	3	1.0000	1.0000	0

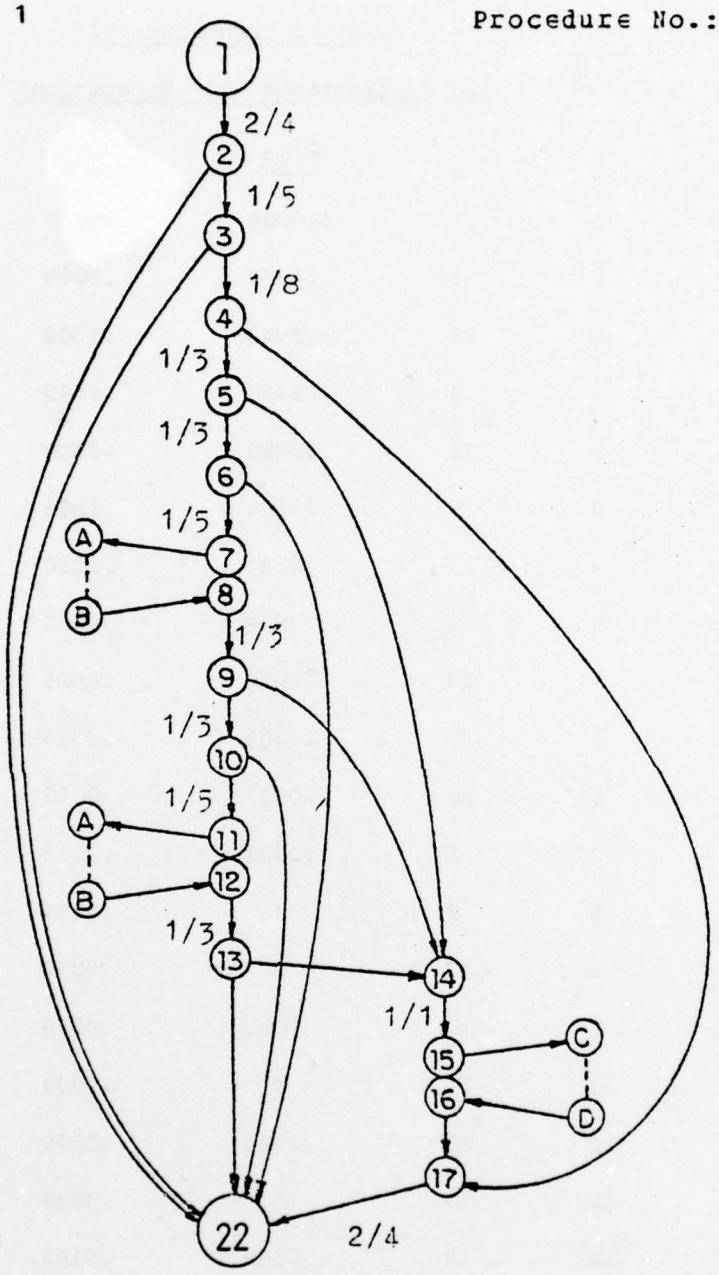
C	D	.5016	.5000	0
D	9	.5016	.5000	0
				25

$\Sigma\ p_{ij}\ s_{ij} = 11.7500$

E(1) = .5595

E(1)/u = .4700

Module: 1 Procedure No.: 91



Number of nodes:	22
Number of arcs:	30
Number of paths:	9
Number of source stmts.:	14
Average error found:	0.1438
Percentage errors found:	21.57

Module 1 Procedure 91

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4957	.5000	1
2	22	.5043	.5000	0
3	4	.2494	.2500	1
3	22	.2463	.2500	0
4	5	.1254	.1250	1
4	17	.1240	.1250	0
5	6	.0628	.0625	1
5	14	.0626	.0625	0
6	7	.0301	.0313	1
6	22	.0327	.0313	0
7	A	.0301	.0313	0
8	9	*	.0157	1
9	10	*	.0078	1
9	14	.0085	.0078	0
10	11	*	.0039	1
10	22	.0054	.0039	0
11	20	*	.0039	0
12	13	.0162	.0153	1
13	14	.0082	.0078	0
13	22	.0080	.0078	0
14	15	.0793	.0860	1
15	18	.0793	.0860	0

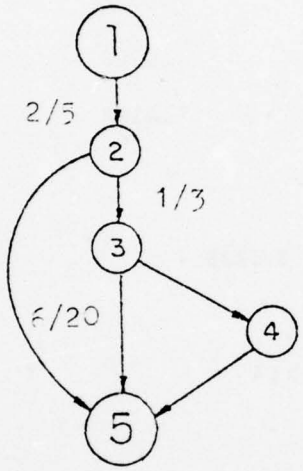
16	17	.0793	.0860	0
17	22	.2033	.2110	2
C	D	.0793	.0860	0
D	16	.0793	.0860	0
A	B	.0348	.0313	0
B	8	.0186	.0157	0
B	12	.0162	.0157	0
				14

$\Sigma\ p_{ij}\ s_{ij} = 3.5195$

E(1) = .1676

E(1)/u = .2514

Module: 1 Procedure No.: 92



Number of nodes:	5
Number of arcs:	6
Number of paths:	3
Number of source stmts.:	9
Average error found:	0.1837
Percentage errors found:	42.86

Module 1 Procedure No. 92

500 Replications 500 Repetitions

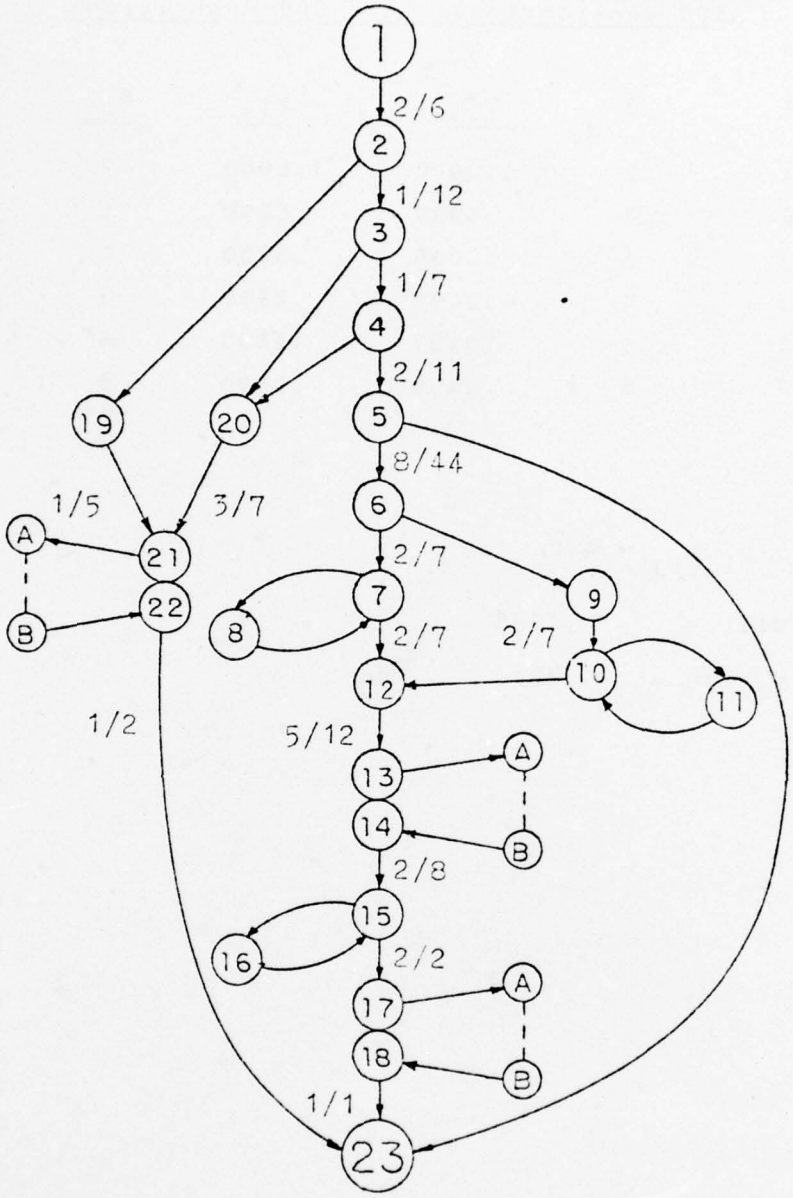
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4995	.5000	1
2	5	.5005	.5000	0
3	4	.2499	.2500	0
3	5	.2497	.2500	6
4	5	.2499	.2500	0
				9

$\sum_{ij} p_{ij} s_{ij} = 4.0$

E(1) = .1905

E(1)/U = .4444

Module: 1 Procedure No.: 93



Number of nodes:	25
Number of arcs:	34
Number of paths:	12
Number of source stmts.:	34
Average error found:	0.3972
Percentage errors found:	24.53

Module 1 Procedure 93

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4996	.5000	1
2	19	.5004	.5000	0
3	4	.2537	.2500	1
3	20	.2459	.2500	0
4	5	.1301	.1250	2
4	20	.1236	.1250	0
5	6	.0659	.0625	8
5	23	.0642	.0625	0
6	7	.0339	.0313	2
6	9	.0320	.0313	0
7	8	*	.0157	0
7	12	.0339	.0313	1
8	7	*	.0157	0
9	10	.0320	.0313	2
10	11	*	.0157	0
10	12	.0320	.0313	0
11	10	*	.0157	0
12	13	.0659	.0625	5
13	A	.0659	.0625	0
14	15	*	.3125	2
15	16	*	.1563	0

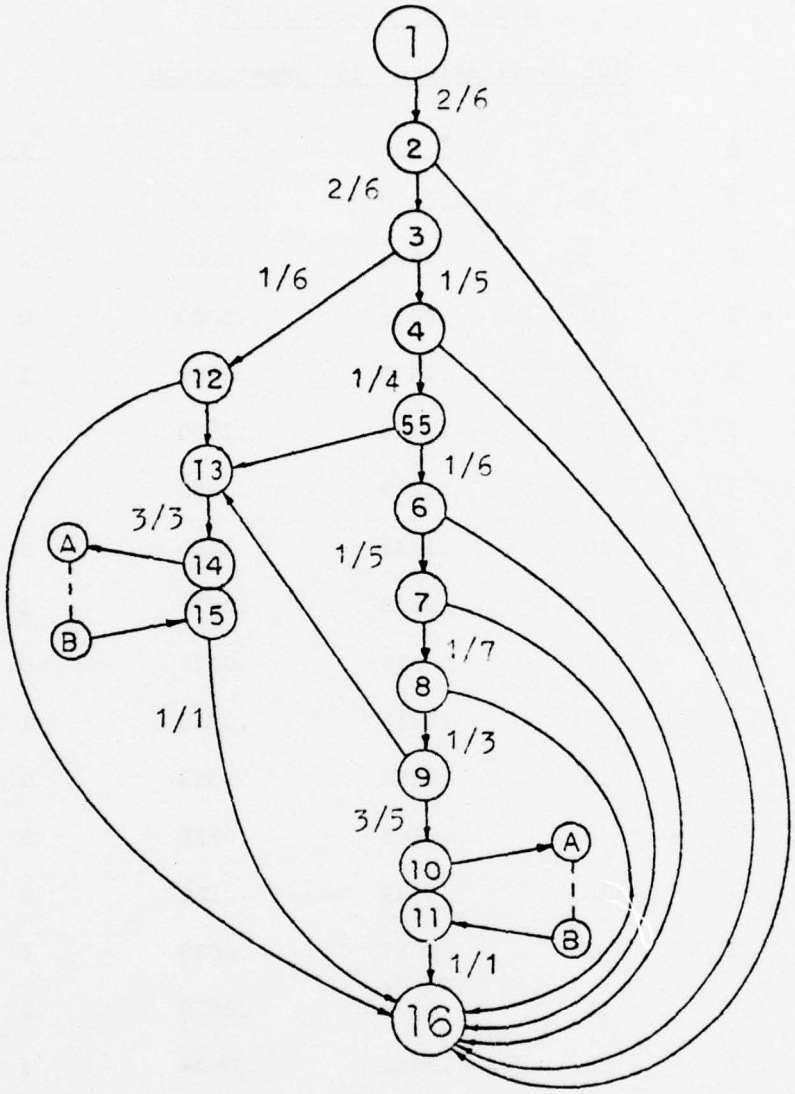
15	17	*	.3125	2
16	15	*	.1563	0
17	A	*	.3125	0
18	23	.4629	.4688	1
19	21	.5004	.5000	1
20	21	.3695	.3750	3
21	A	.8699	.8750	0
22	23	.4729	.4688	1
A	B	*	.9375	0
B	14	*	.3125	0
B	18	.4629	.4688	0
B	22	.4729	.4688	0
				34

$\Sigma\ p_{ij}\ s_{ij} = 7.7814$

$E(1) = .3705$

$E(1)/u = .2289$

Module: 1 Procedure No.: 95



Number of nodes:	18
Number of arcs:	27
Number of paths:	10
Number of source stmts.:	19
Average error found:	0.1822
Percentage errors found:	20.14

Module 1 Procedure 95

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4964	.5000	2
2	16	.5036	.5000	0
3	4	.2470	.2500	1
3	12	.2494	.2500	1
4	5	.1229	.1250	1
4	16	.1241	.1250	0
5	6	.0633	.0625	1
5	13	.0596	.0625	0
6	7	.0315	.0313	1
6	16	.0318	.0313	0
7	8	.0166	.0156	1
7	16	.0149	.0156	0
8	9	.0087	.0078	1
8	16	.0079	.0078	0
9	10	.0042	.0039	3
9	13	.0045	.0039	0
10	A	.0042	.0039	0
11	16	.0978	.0977	1
12	13	.1255	.1250	0
12	16	.1239	.1250	3
13	14	.1896	.1914	0

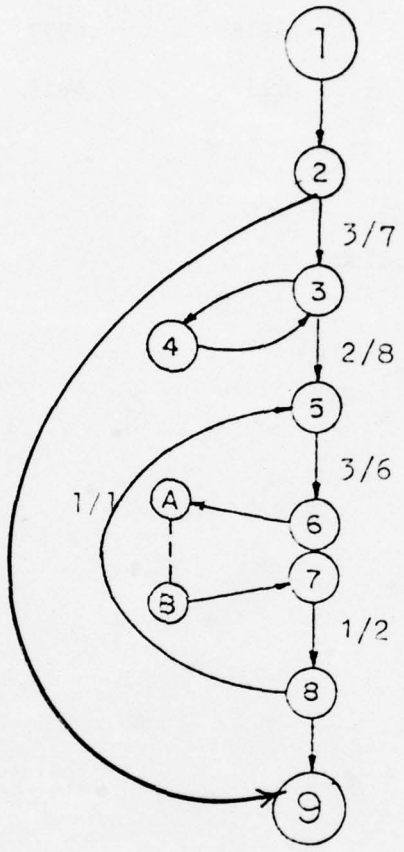
14	A	.1896	.1914	1
15	16	.0960	.0977	0
A	B	.1938	.1953	0
B	11	.0978	.0977	0
B	15	.0960	.0977	0
				19

$\sum p_{ij} s_{ij} = 4.4180$

$E(1) = .2104$

$E(1)/u = .2325$

Module: 2 Procedure No. 15



Number of nodes:	11
Number of arcs:	13
Number of paths:	5
Number of source stmts.:	10
Average error found:	0.0836
Percentage errors found:	42.64

MODULE 2 Procedure 15

100 Replications 100 Repetitions

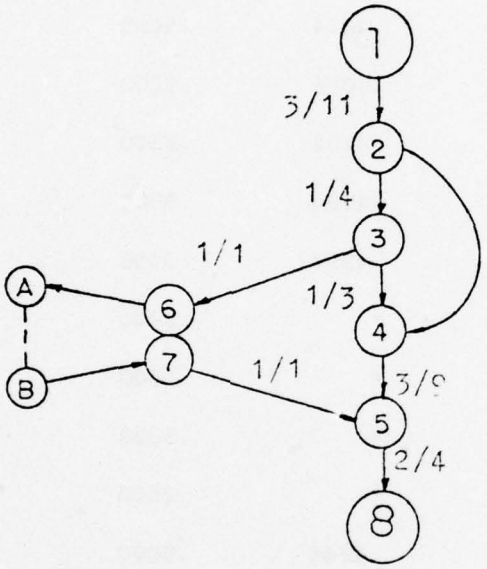
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	0
2	3	.4944	.5000	3
2	9	.5056	.5000	0
3	4	.4992	.2500	0
3	5	.4994	.5000	2
4	3	.4992	.2500	0
5	6	*	.5000	3
6	A	*	.5000	0
7	8	*	.5000	1
8	5	*	.2500	1
8	9	.4944	.5000	0
A	B	*	.5000	0
B	7	*	.5000	0
				10

$\sum p_{ij}s_{ij} = 4.7500$

E(1) = .0931

E(1)/u = .4750

Module: 2 Procedure No.: 23



Number of nodes:	10
Number of arcs:	11
Number of paths:	3
Number of source stmts.:	12
Average error found:	0.1592
Percentage errors found:	67.66

MODULE 2 Procedure 23

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.000	1.000	3
2	3	.4947	.5000	1
2	4	.5053	.5000	0
3	4	.2497	.2500	1
3	6	.2450	.2500	1
4	5	.7550	.7500	3
5	8	.7550	.7500	2
6	A	.2450	.2500	0
7	8	.2450	.2500	1
A	B	.2450	.2500	0
B	7	.2450	.2500	0
				12

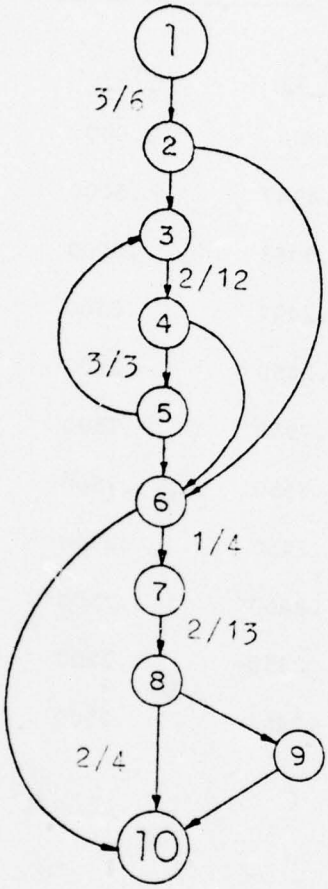
$\sum p_{ij}s_{ij} = 8.0000$

E(1) = .1569

E(1)/u = .6667

Module: 2

Procedure No.: 41



Number of nodes:	10
Number of arcs:	14
Number of paths:	12
Number of source stmts.:	13
Average error found:	0.1554
Percentage errors found:	60.96

Module 2 Procedure 41

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	3
2	3	.4955	.5000	0
2	6	.5045	.5000	0
3	4	*	.5000	2
4	5	*	.2500	3
4	6	.3303	.3333	0
5	3	*	.1250	0
5	6	.1652	.1667	0
6	7	.4993	.5000	1
6	10	.5007	.5000	0
7	8	.4993	.5000	2
8	9	.2474	.2500	0
8	10	.2519	.2500	2
9	10	.2474	.2500	0
				13

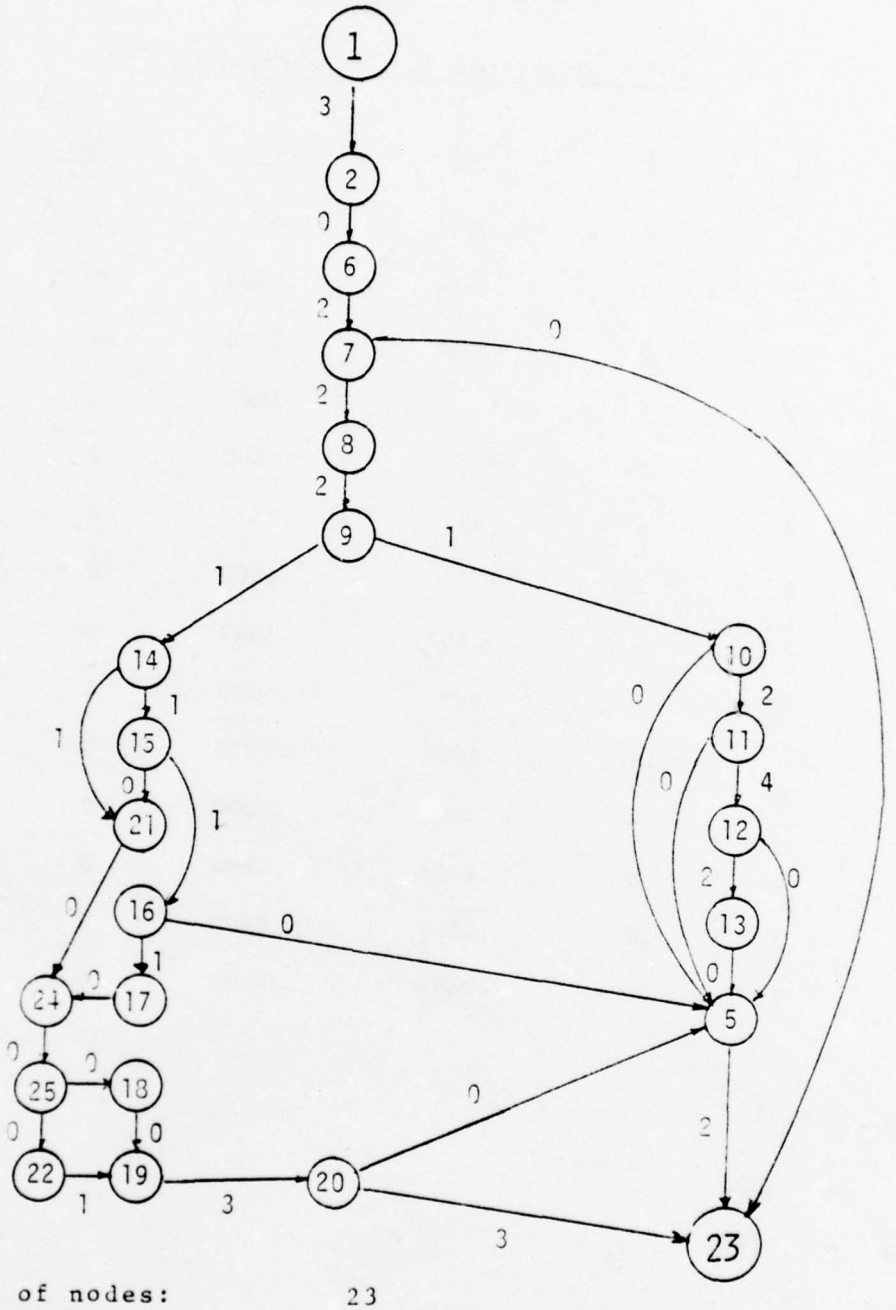
$\sum p_{ij}s_{ij} = 6.75$

E(1) = .1324

E(1)/u = .5192

Module: 2

Procedure No. 46



Number of nodes: 23

Number of arcs: 32

Number of paths: 18

Number of source stmts.: 32

Average error found: 0.1657

Percentage errors found: 26.41

Module 2 Procedure 46

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	3
2	6	1.0000	1.0000	0
5	23	.3908	.3906	2
6	7	1.0000	1.0000	2
7	8	.4955	.5000	2
7	23	.5045	.5000	0
8	9	.4955	.5000	2
9	10	.2485	.2500	1
9	14	.2470	.2500	1
10	5	.1220	.1250	0
10	11	.1265	.1250	2
11	5	.0642	.0625	0
11	12	.0623	.0625	4
12	5	.0333	.0313	0
12	13	.0290	.0313	2
13	5	.0290	.0313	0
14	15	.1228	.1250	1
14	21	.1242	.1250	1
15	16	.0607	.0625	1
15	21	.0621	.0625	0
16	5	.0330	.0313	0
16	17	.0277	.0313	1
17	24	.0277	.0313	0
18	19	.1051	.1094	0

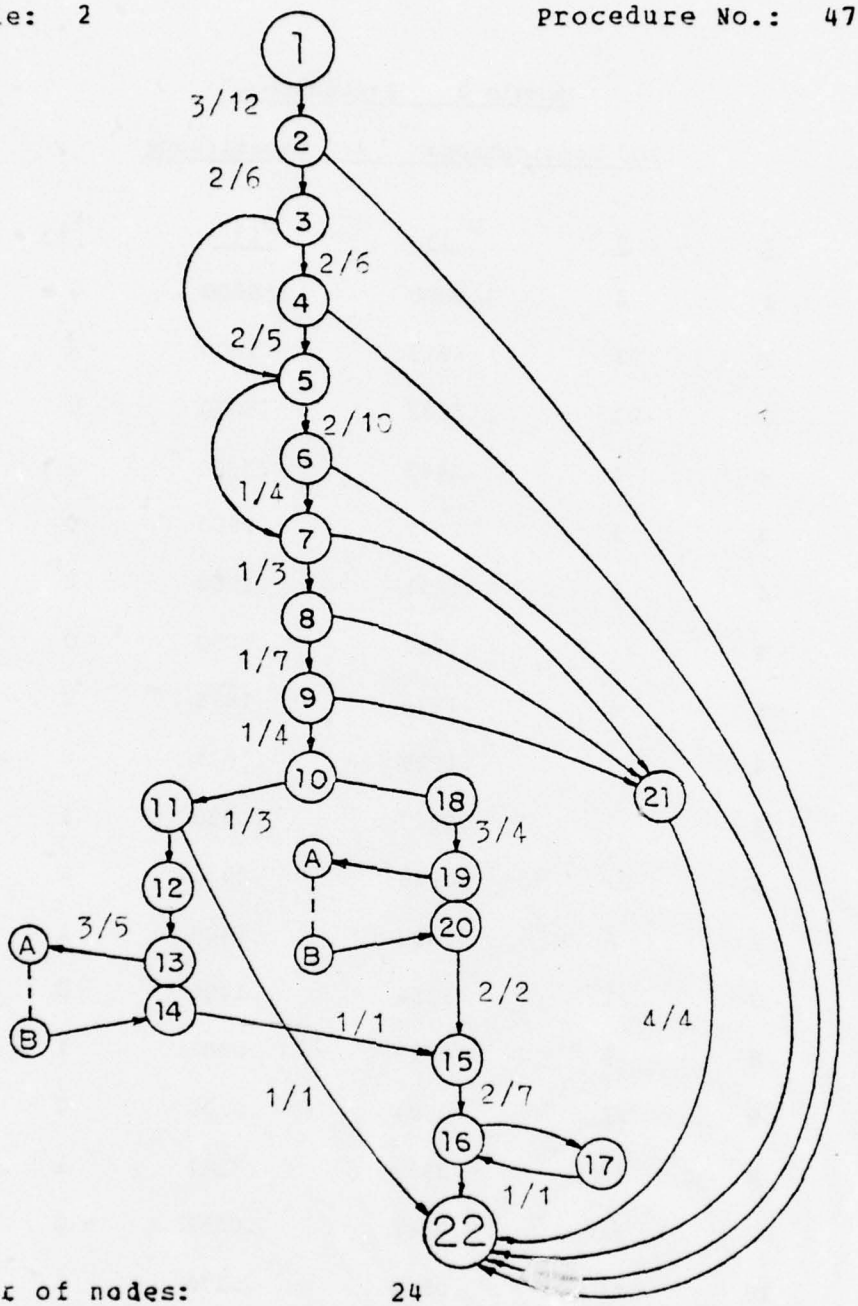
19	20	.2140	.2188	3
20	5	.1093	.1094	0
20	23	.1047	.1094	3
21	24	.1863	.1875	
22	19	.1089	.1094	1
24	25	.2140	.2188	0
25	18	.1051	.1094	0
25	22	.1089	.1094	0
				32

$\sum p_{ij}s_{ij} = 10.2815$

E(1) = .2016

E(1)/u = .3213

Module: 2 Procedure No.: 47



Number of nodes:	24
Number of arcs:	35
Number of paths:	36
Number of source stmts.:	33
Average error found:	0.1163
Percentage errors found:	17.97

Module 2 Procedure 47

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	3
2	3	.4915	.5000	2
2	22	.5085	.5000	0
3	4	.2443	.2500	2
3	5	.2472	.2500	0
4	5	.1231	.1250	2
4	22	.1212	.1250	0
5	6	.1874	.1875	2
5	7	.1829	.1875	0
6	7	.0928	.0938	1
6	22	.0946	.0938	0
7	8	.1403	.1406	1
7	21	.1354	.1406	0
8	9	.0719	.0703	1
8	21	.0684	.0703	0
9	10	.0352	.0352	1
9	21	.0367	.0352	0
10	11	.0184	.0176	1
10	18	.0168	.0176	0
11	12	.0089	.0088	0
11	22	.0095	.0088	1
12	13	.0089	.0088	3
13	A	.0089	.0088	0

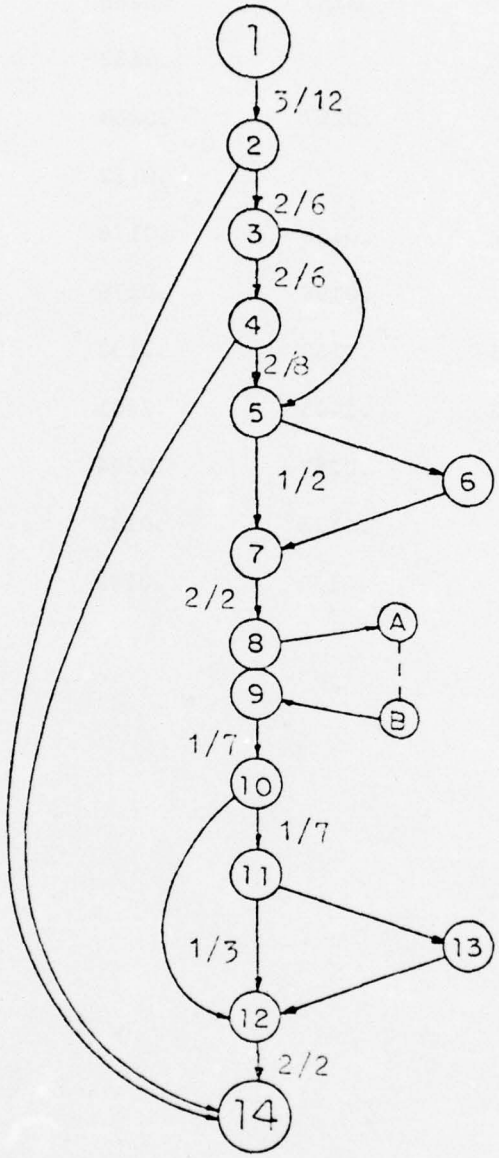
14	15	.0127	.0132	1
15	16	.0257	.0268	2
16	17	*	.0132	0
16	22	.0257	.0264	1
17	16	*	.0132	0
18	19	.0168	.0176	3
19	A	.0168	.0176	0
20	15	.0130	.0132	2
21	22	.2405	.2461	4
A	B	.0257	.0264	0
B	14	.0127	.0132	0
B	20	.0130	.0132	0
				33

$\sum p_{ij}s_{ij} = 6.7983$

E(1) = .1333

E(1)/u = .2060

Module: 2 Procedure No.: 48



Number of nodes:	16
Number of arcs:	21
Number of paths:	14
Number of source stmts.:	17
Average error found:	0.1580
Percentage errors found:	47.40

Module 2 Procedure 48

100 Replications 100 Repetitions

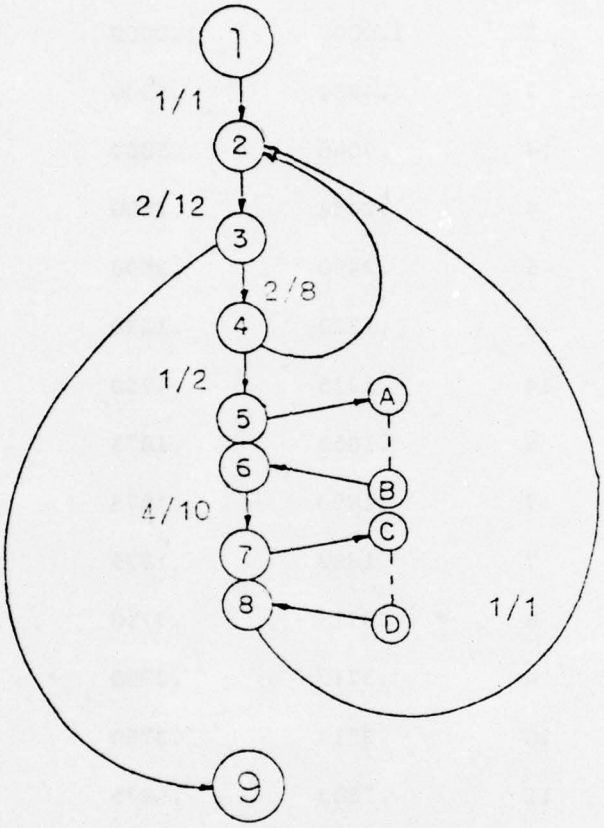
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	3
2	3	.4934	.5000	2
2	14	.5066	.5000	0
3	4	.2444	.2500	2
3	5	.2490	.2500	0
4	5	.1229	.1250	2
4	14	.1215	.1250	0
5	6	.1869	.1875	0
5	7	.1850	.1875	1
6	7	.1869	.1875	0
7	8	.3719	.3750	2
8	A	.3719	.3750	0
9	10	.3719	.3750	1
10	11	.1863	.1875	1
10	13	.1856	.1875	0
11	12	.0979	.0938	0
11	13	.0884	.0938	1
12	13	.0979	.0938	0
13	14	.3719	.3750	2
A	B	.3719	.3750	0
B	9	.3719	.3750	0
				17

$\sum p_{ij} s_{ij} = 7.9063$

E(1) = .1550

E(1)/U = .4651

Module: 2 Procedure No.: 79



Number of nodes:	13
Number of arcs:	14
Number of paths:	5
Number of source stmts.:	11
Average error found:	0.1379
Percentage errors found:	63.94

Module 2 Procedure 79

100 Replications 100 Repetitions

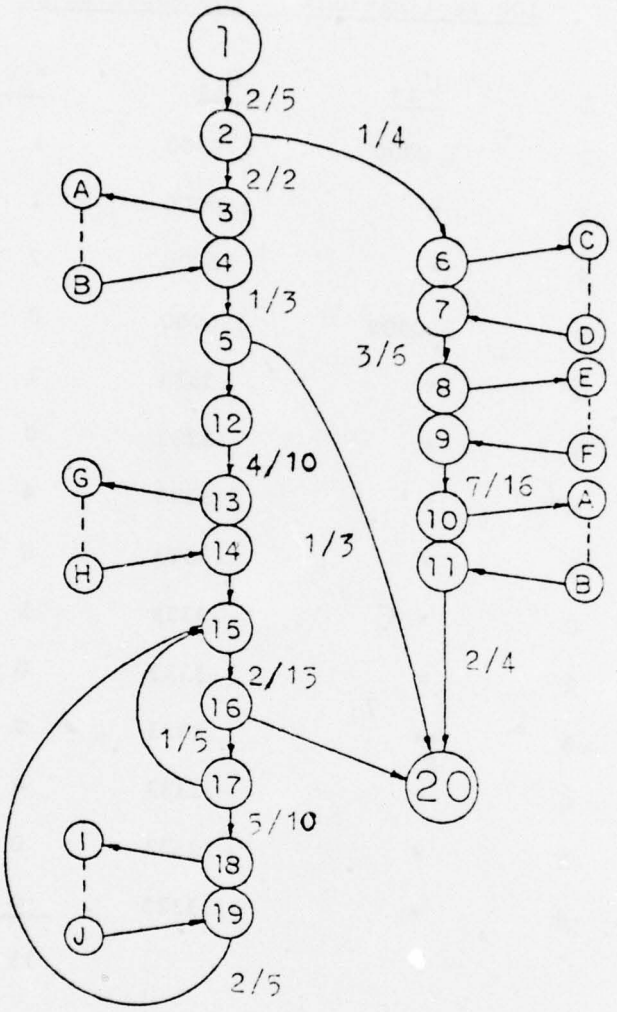
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	1
2	3	*	1.0000	2
3	4	*	.5000	2
3	9	1.0000	1.0000	0
4	2	*	.3333	1
4	5	*	.3333	0
5	A	*	.3333	4
6	7	*	.3333	0
7	C	*	.3333	1
8	2	*	.3333	0
A	B	*	.3333	0
B	6	*	.3333	0
C	D	*	.3333	0
D	8	*	.3333	0
				11

$\sum p_{ij}s_{ij} = 6.0000$

E(1) = .1176

E(1)/u = .5455

Module: 2 Procedure No.: 86



Number of nodes:	30
Number of arcs:	34
Number of paths:	6
Number of source stmts.:	33
Average error found:	0.2042
Percentage errors found:	31.55

Module 2 Procedure 86

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4962	.5000	2
2	6	.5038	.5000	1
3	A	.4962	.5000	0
4	5	.4996	.5000	1
5	12	.2524	.2500	0
5	20	.2472	.2500	1
6	C	.5038	.5000	0
7	8	.5038	.5000	3
8	E	.5038	.5000	0
9	10	.5038	.5000	7
10	A	.5038	.5000	0
11	20	.5004	.5000	2
12	13	.2524	.2500	4
13	G	.2524	.2500	0
14	15	.2524	.2500	0
15	16	*	.2500	2
16	17	*	.1250	1
16	20	.2524	.2500	0
17	15	*	.0833	0
17	18	*	.0833	5
18	I	*	.0833	0
19	15	*	.0833	2
A	B	1.0000	1.0000	0
B	4	.4996	.5000	0

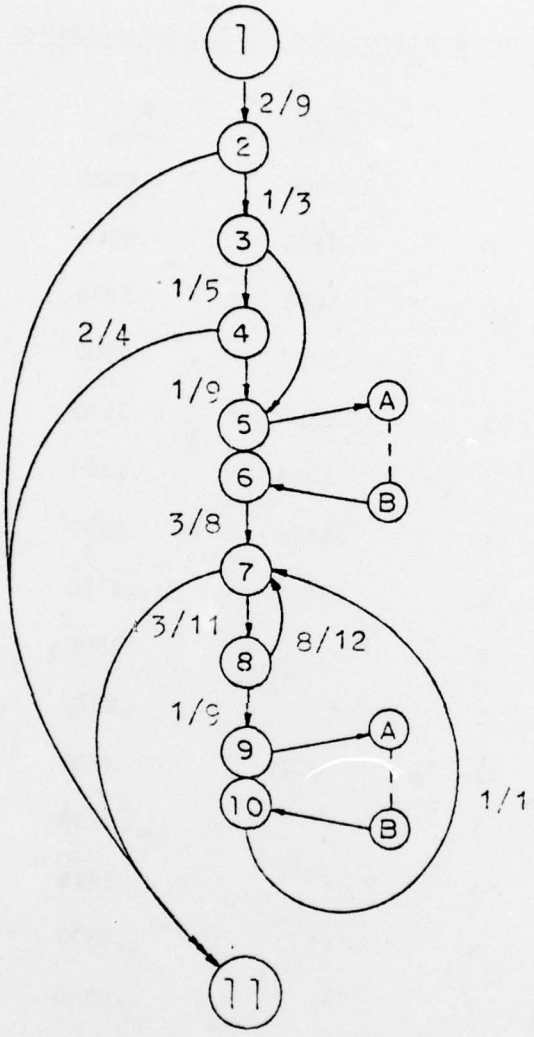
B	11	.5004	.5000	0
C	D	.5038	.5000	0
D	7	.5038	.5000	0
E	F	.5038	.5000	0
F	9	.5038	.5000	0
G	H	.2524	.2500	0
H	14	.2524	.2500	0
I	J	*	.0833	0
J	19	*	.0833	0
				33

$\sum p_{ij}s_{ij} = 12.4582$

E(1) = .2443

E(1)/u = .3775

Module: 2 Procedure No.: 90



Number of nodes:	13
Number of arcs:	18
Number of paths:	8
Number of source stmts.:	23
Average error found:	0.0958
Percentage errors found:	21.24

Module 2 Procedure 90

100 Replications 100 Repetitions

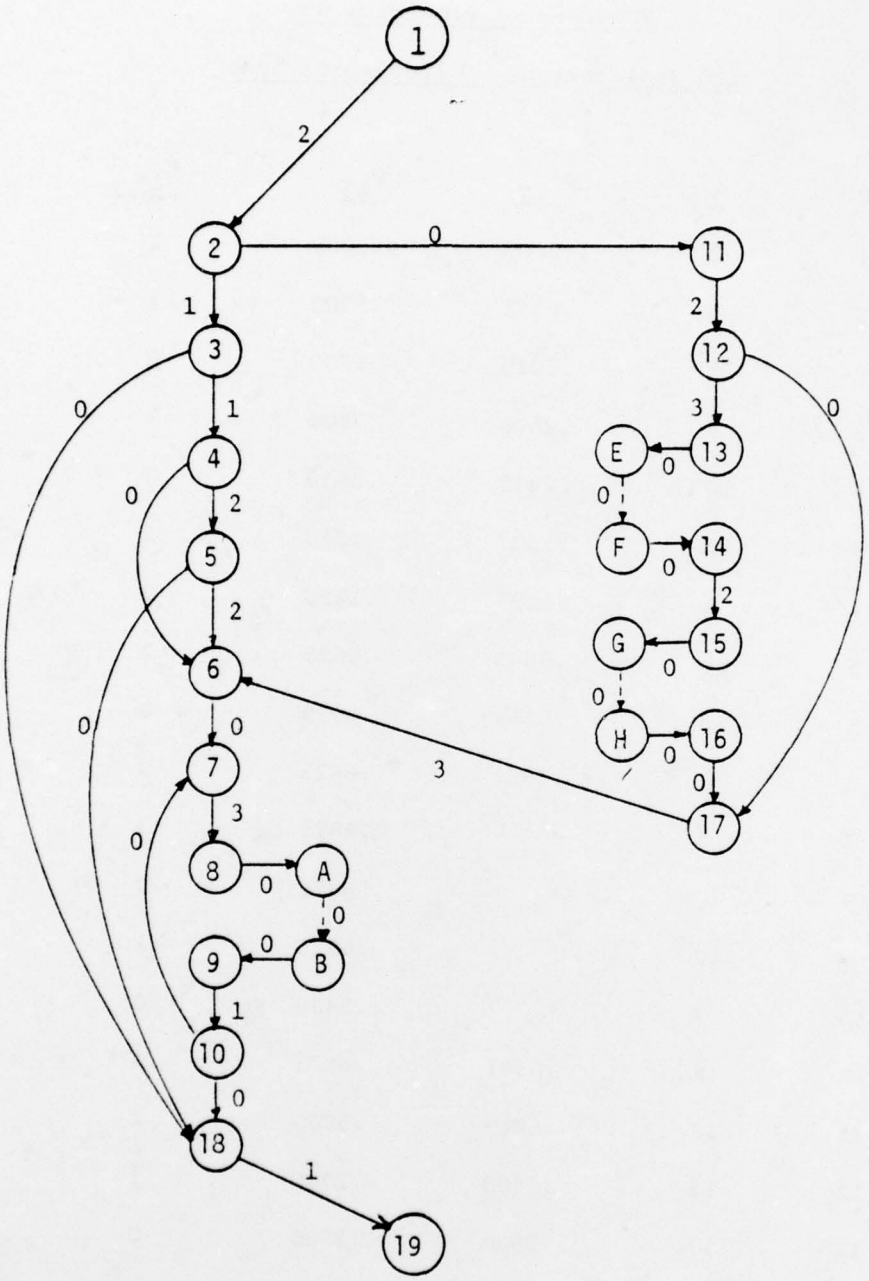
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4925	.5000	1
2	11	.5075	.5000	0
3	4	.2427	.2500	1
3	5	.2498	.2500	0
4	5	.1249	.1250	1
4	11	.1178	.1250	2
5	A	.3747	.3750	0
6	7	*	.2500	3
7	8	*	.1875	3
7	11	.3747	.3750	0
8	7	*	.0938	8
8	9	*	.1875	1
9	A	*	.1875	0
10	7	*	.2500	1
A	B	*	.3750	0
B	6	*	.2500	0
B	10	*	.2500	0
				23

$\sum p_{ij}s_{ij} = 5.625$

E(1) = .11029

E(1)/u = .2446

Module: 2 Procedure No.: 99



Number of nodes:	25
Number of arcs:	30
Number of paths:	10
Number of source stmts.:	23
Average error found:	0.1513
Percentage errors found:	33.55

MODULE 2 Procedure 99

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4981	.5000	1
2	11	.5019	.5000	0
3	4	.2506	.2500	1
3	18	.2475	.2500	0
4	5	.1307	.1250	2
4	6	.1199	.1250	0
5	6	.0643	.0625	2
5	18	.0664	.0625	0
6	7	.6861	.6875	0
7	8	*	.6875	3
8	A	*	.6875	0
9	10	*	.6875	1
10	7	*	.3438	0
10	18	.6861	.6875	0
11	12	.5019	.5000	2
12	13	.2483	.2500	3
12	17	.2536	.2500	0
13	E	.2483	.2500	0
14	15	.2483	.2500	2
15	G	.2483	.2500	0
16	17	.2483	.2500	0
17	6	.5019	.5000	3
18	19	1.0000	1.0000	1

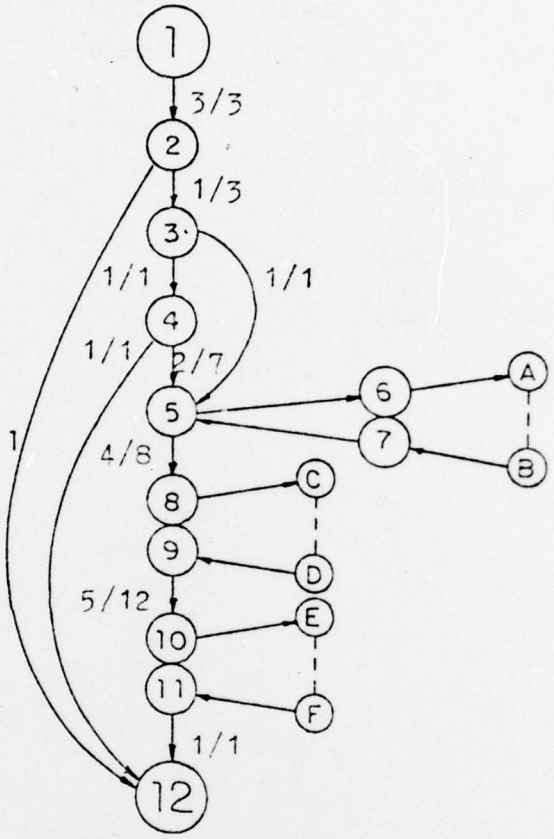
A	B	*	.6875	0
B	9	*	.6875	0
E	F	.2483	.2500	0
F	14	.2483	.2500	0
G	H	.2483	.2500	0
H	16	.2483	.2500	0
				23

$\sum p_{ij}s_{ij} = 10.625$

$E(1) = .20833$

$E(1)/U = .4620$

Module: 2 Procedure No.: 122



Number of nodes:	18
Number of arcs:	21
Number of paths:	6
Number of source stmts.:	20
Average error found:	0.1686
Percentage errors found:	42.99

Module 2 Procedure 122

100 Replications 100 Repetitions

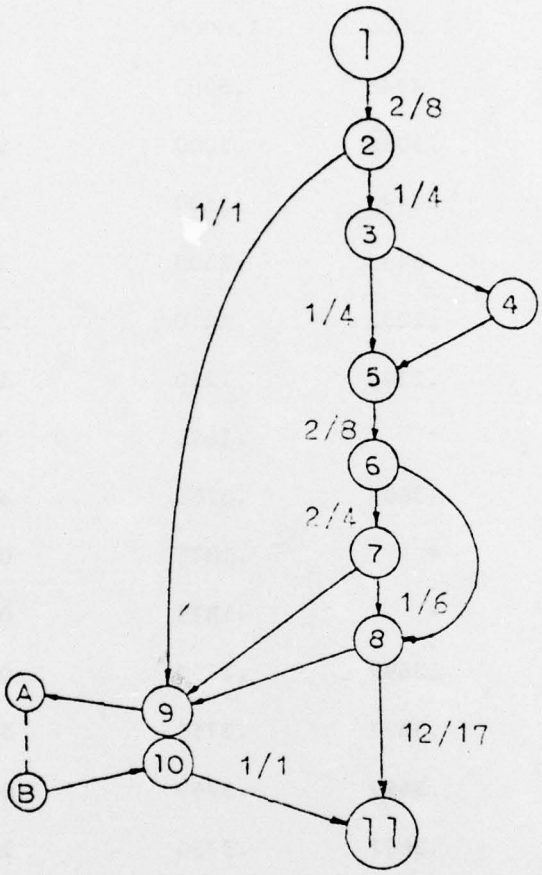
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	3
2	3	.4945	.5000	1
2	12	.5055	.5000	1
3	4	.2480	.2500	1
3	5	.2465	.2500	1
4	5	.1232	.1250	2
4	12	.1248	.1250	1
5	6	*	.1875	0
5	8	.3697	.3750	4
6	A	*	.1875	0
7	5	*	.1875	0
8	C	.3697	.3750	0
9	10	.3697	.3750	5
10	E	.3697	.3750	0
11	12	.3697	.3750	1
A	B	*	.1875	0
B	7	*	.1875	0
C	D	.3697	.3750	0
D	9	.3697	.3750	0
E	F	.3697	.3750	0
F	11	.3697	.3750	0
				20

$\sum p_{ij}s_{ij} = 8.625$

E(1) = .1691

E(1)/u = .4313

Module: 2 Procedure No.: 137



Number of nodes:	13
Number of arcs:	17
Number of paths:	11
Number of source stmts.:	23
Average error found:	0.1178
Percentage errors found:	26.12

MODULE 2 Procedure 137

100 Replications 100 Repetitions

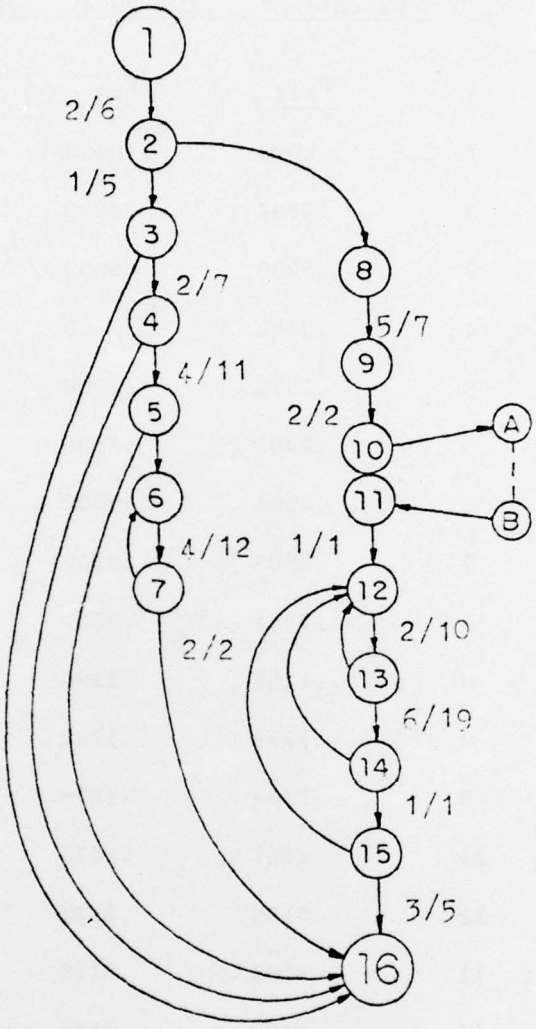
i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4994	.5000	1
2	9	.5006	.5000	1
3	4	.2482	.2500	0
3	5	.2512	.2500	1
4	5	.2482	.2500	0
5	6	.4994	.5000	2
6	7	.2503	.2500	2
6	8	.2491	.2500	0
7	8	.1269	.1250	1
7	9	.1234	.1250	0
8	9	.1863	.1875	0
8	11	.1897	.1875	12
9	12	.8103	.8125	0
10	11	.8103	.8125	1
12	13	.8103	.8125	0
13	10	.8103	.8125	0
				23

$\sum p_{ij}s_{ij} = 8.0625$

E(1) = .15809

E(1)/u = .3505

Module: 2 Procedure No.: 149



Number of nodes:	18
Number of arcs:	25
Number of paths:	9
Number of source stmts.:	35
Average error found:	0.2357
Percentage errors found:	34.34

Module 2 Procedure 149

100 Replications 100 Repetitions

i	j	p'_{ij}	p_{ij}	s_{ij}
1	2	1.0000	1.0000	2
2	3	.4895	.5000	1
2	8	.5105	.5000	0
3	4	.2410	.2500	2
3	16	.2485	.2500	0
4	5	.1213	.1250	4
4	16	.1197	.1250	0
5	6	.0604	.0625	0
5	16	.0609	.0625	0
6	7	*	.0625	4
7	6	*	.03125	0
7	16	.0604	.0625	2
8	9	.5105	.5000	5
9	10	.5105	.5000	2
10	A	.5105	.5000	0
11	12	.5105	.5000	1
12	13	*	.5000	2
13	12	*	.3750	0
13	14	*	.5000	6
14	12	*	.3333	0
14	15	*	.5000	1
15	12	*	.2500	0

15	16	.5105	.5000	3
A	B	.5105	.5000	0
B	11	.5105	.5000	0
				35

$\sum p_{ij}s_{ij} = 13.8750$

E(1) = .2721

E(1)/u = .3964

INITIAL DISTRIBUTION LIST

	Copies
Mr. Richard Pariseau Naval Air Development Center Warminster, Pennsylvania	4
Defense Documentation Center Cameron Station Arlington, Virginia 22314	2
Naval Postgraduate School	
Dean of Research	2
Knox Library	2
W. R. Church Computer Center Library	1
Prof. G. Bradley	1
CDR C. Gibfried	1
Prof. G. Howard	1
Prof. C. Jones	1
Prof. N. Schneidewind	12