TECHNICAL MEMORANDUM NO. 151

MATHNET AND RISCA
(NETWORK ANALYZER PROGRAMS),
A USERS' MANUAL

WILBERT J. BROOKS
WARD V. FOSTER
RICHARD T. MARUYAMA

NOVEMBER 1972

Approved for public release; distribution unlimited.

U.S. ARMY MATERIAL SYSTEMS ANALYSIS AGENCY
Aberdeen Proving Ground, Maryland
DISPOSITION

Destroy this report when no longer needed. Do not return it to the originator.

DISCLAIMER

The findings in this report are not to be construed as an official Department of the Army position.

WARNING

Information and data contained in this document are based on the input available at the time of preparation. The results may be subject to change and should not be construed as representing the AMC position unless so specified.
ABSTRACT

Two network analyzer programs, MATHNET and RISCA, which allow the analyst to simulate a general class of network representations are described and evaluated for the potential user.

Network concepts, program listings, and program flow charts are included for both programs in addition to detailed description of input preparation and output interpretation for a hypothetical example.
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION</td>
<td>7</td>
</tr>
<tr>
<td>1.1 Background</td>
<td>7</td>
</tr>
<tr>
<td>1.2 Organization of the Report</td>
<td>7</td>
</tr>
<tr>
<td>1.3 Network Concepts</td>
<td>8</td>
</tr>
<tr>
<td>1.4 Types of Network Representations</td>
<td>8</td>
</tr>
<tr>
<td>2. RISCA AND MATHNET CHARACTERISTICS AND CAPABILITIES</td>
<td>14</td>
</tr>
<tr>
<td>3. RISCA AND MATHNET PROGRAM INPUT</td>
<td>20</td>
</tr>
<tr>
<td>3.1 Introduction</td>
<td>20</td>
</tr>
<tr>
<td>3.2 Arc and Node Notation</td>
<td>21</td>
</tr>
<tr>
<td>3.3 Input Format</td>
<td>24</td>
</tr>
<tr>
<td>3.4 Example Problem</td>
<td>29</td>
</tr>
<tr>
<td>4. RISCA AND MATHNET PROGRAM OUTPUT</td>
<td>39</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>39</td>
</tr>
<tr>
<td>4.2 MATHNET Output</td>
<td>39</td>
</tr>
<tr>
<td>4.3 RISCA Output</td>
<td>41</td>
</tr>
<tr>
<td>4.4 MATHNET and RISCA Output Comparison</td>
<td>42</td>
</tr>
<tr>
<td>4.5 Use of the Output</td>
<td>43</td>
</tr>
<tr>
<td>5. SUMMARY</td>
<td>47</td>
</tr>
<tr>
<td>BIBLIOGRAPHY</td>
<td>49</td>
</tr>
<tr>
<td>APPENDIX I MATHNET PROGRAM LISTING</td>
<td>51</td>
</tr>
<tr>
<td>APPENDIX II RISCA PROGRAM LISTING</td>
<td>77</td>
</tr>
<tr>
<td>APPENDIX III FLOW CHARTS OF RISCA AND MATHNET SUBROUTINES</td>
<td>99</td>
</tr>
<tr>
<td>APPENDIX IV MATHNET OUTPUT FOR EXAMPLE PROBLEM</td>
<td>135</td>
</tr>
<tr>
<td>APPENDIX V RISCA OUTPUT FOR EXAMPLE PROBLEM</td>
<td>161</td>
</tr>
<tr>
<td>DISTRIBUTION LIST</td>
<td>183</td>
</tr>
</tbody>
</table>

Next page is blank.
1. INTRODUCTION

1.1 Background.

Although having much wider application, the primary use of network analysis has been in the planning and control of R&D projects. The use of network analysis techniques for this purpose had its origin with PERT (Program Evaluation and Review Technique) in 1958 in the Polaris Program. Since that time, PERT and many other network analysis techniques have gained wide acceptance both in the Department of Defense and in private industry.

In the last two years, two network analyzer programs, MATHNET* and RISCA**, have gained acceptance within AMC. Under PROMAP 70 the responsibility for instruction in risk analysis was assigned to the Army Logistics Management Center (ALMC) at Fort Lee, Virginia. In order to expedite the program, a contract was let to MATHEMATICA by the Army Research Office to develop this course of instruction. MATHNET, developed by MATHEMATICA as a teaching aid for this course was modified by ALMC. They call their version of the program RISCA.

The version of MATHNET currently being used at this agency is not identical to that developed by Mathematica. The original Mathematica program contained several logic inconsistencies with respect to the time and cost values generated in various types of nodes. These logic errors, corrected by analysts at Picatinny Arsenal, are not in the version of MATHNET discussed in this manual. The RISCA version discussed herein is also free of these logic inconsistencies.

Even though the two programs are accepted and the results used, there does not exist an adequately documented users' manual which compares both programs at this time. Recognizing the utility of network analysis techniques in the Materiel Acquisition Decision Making Process and the need for such a manual both within AMSAA and AMC, a project was initiated to prepare this users' manual.

1.2 Organization of the Report.

The remainder of the first section is devoted to defining the various characteristics of a network and the types of networks that can be modeled using RISCA and MATHNET. In Section 2, the various capabilities of the two network analyzer programs are described. In Section 3, a description of the construction of a network and the method for inputting data to the program is provided. The output of

* Mathematical Network Analyzer.

** Risk Information System and Cost Analysis.
both programs is described in Section 4. In the last section of the manual, the program capabilities are compared and contrasted and a recommendation is made concerning which program to use.

1.3 Network Concepts.

Before the two network analyzer programs are described and compared, the more basic concepts of a graph, a node, an arc, a network and a path must be defined.

Figure 1.1 is an example of a graph. The circles represent nodes, and the lines joining the nodes are called arcs. Hence, a graph is a collection of two or more nodes joined by arcs. Any arc can be characterized by the pair of nodes that it connects. For example (1,2) characterizes the arc connecting nodes 1 and 2 in Figure 1.1.

The only difference between a graph and a network is that the arcs have some type of flow in them (see Figure 1.2). One example of a system that can be represented by a network is a development test-program. The nodes* in a development test-program represent the initiation or completion of various tests, the arcs** represent the actual tests being conducted and the flow in the arcs is time and/or cost involved in testing.

Finally, a path is defined as a sequence of arcs connecting two nodes. For example, the following sequence of arcs form paths between nodes 1 and 4 in Figure 1.2:

PATH (M) (1,2), (2,4); PATH (O) (1,4)  
PATH (N) (1,3), (3,4)

1.4 Types of Network Representations.

Given the preceding concepts, it is now possible to describe the different types of network representations and network analysis techniques for analyzing them. The differences in the networks result from assumptions made concerning the events and the flows in the activities being modeled in the project. As mentioned previously, the arcs represent activities, the nodes represent events, and the flow in the arcs usually represents time and/or cost.

* Nodes generally refer to events.

** Arcs generally refer to activities or jobs.
Figure 1.1 Example of a Graph.

Figure 1.2 Example of a Network.
Three types of network models will be discussed. They are:

a. Deterministic event and activity time/cost network

b. Deterministic event and probabilistic activity time/cost network

c. Probabilistic event and activity time/cost network

It should be pointed out that this discussion will be very general and only the major attributes and assumptions about the networks being modeled will be discussed.

The differences in the types of network representations are most easily described by analyzing one example under the varying assumptions made concerning the events and flows in the activities. Changing the oil in a car is the example that will be used throughout this discussion.

The first type of possible network representation for describing the events and activities involved in changing the oil in a car is illustrated in Figure 1.3. Here there are five milestone events in changing the oil in a car. For this network representation, as well as all others, it is assumed that the events must be completed in a particular sequence in order to complete the project. Further, it is assumed that all events must be completed and the completion times are known with certainty (i.e., the events and completion times are assumed to be deterministic). For most programs, these last two assumptions are not thought to be very realistic. For rarely are the events and/or activity times known with certainty. Even though the first type of network representation is not realistic for R&D projects, it is realistic in the construction industry where tasks for a project are known with certainty. Further, these tasks are repetitive so the assumption of deterministic activity times is more realistic. The Critical Path Methodology (CPM) is the name given to the network analysis technique developed by DuPont in 1958 to handle deterministic event and activity time networks (Type I). This technique was initially used to find an efficient method for planning the construction of a new facility.

In the second network type, the assumption of deterministic activity times is replaced by the assumption of probabilistic activity times. This means that the activity times are not known with certainty (i.e., there exists some distribution of activity times). For instance, the activity time for BC in the changing the oil example could vary due to random interruptions such as having to pump gas for cars as they arrive at the station. For instance, the most likely time for completing this activity might be ten minutes and the best and worst times might be five and twenty minutes respectively. Therefore, if this activity time is assumed to be distributed as a triangular distribution, with a minimum, most likely and maximum...
EVENTS
A - CAR IS LEFT AT THE SERVICE STATION.
B - CAR IS ON THE LIFT.
C - OLD OIL IS DRAINED AND THE FILTER REMOVED.
D - NEW FILTER AND OIL IN THE CAR.
E - CAR OFF THE LIFT.

ACTIVITY TIMES
ARC
AB - 2 MINUTES
BC - 5 MINUTES
CD - 6 MINUTES
DE - 2 MINUTES

Figure 1.3. Changing the Oil in a Car.
Figure 1.4 Distribution of Activity Time BC.
EVENTS
A - CAR IS LEFT AT THE SERVICE STATION.
B - CAR IS ON THE LIFT.
C - OIL IS DRAINED.
D - FILTER IS REMOVED.
E - NEW FILTER IS IN THE CAR.
F' AND F - NEW OIL IS IN THE CAR.
G' AND G - CAR OFF THE LIFT.
H - FILTER IS NOT REMOVED BECAUSE IT CANNOT BE REPLACED.

Figure 1.5. Changing the Oil in a Car.
(Probabilistic Network)
activity times being 5, 10, and 20 minutes respectively, then Figure 1.4 represents the distribution of this activity time. The network analysis technique developed to analyze this type of network representation is PERT (Program Evaluation and Review Technique) which was developed in 1958 for planning and controlling the development of the Polaris missile. Even though PERT is more realistic than the CPM, it still is deficient for modeling many R&D projects since the events are assumed to be deterministic.

In the third network type, all events and activities are modeled probabilistically. This type is the most realistic for modeling R&D programs. To illustrate this network type, assume that it may not always be possible, for example, to replace the oil filter because the type of filter required may not be in stock. Assume that there is a .98 probability that it is in stock and a .02 probability that the filter is not in stock. The probabilistic nature of this network type is illustrated in Figure 1.5. In addition, when each of the activity times is modeled probabilistically, this network type is beyond the scope of PERT and CPM and due to the additional complexity introduced by probabilistic events, the development of RISCA and MATHNET was prompted.

It should be noted that both MATHNET and RISCA also allow one to analyze the first two network types.

2. RISCA AND MATHNET CHARACTERISTICS AND CAPABILITIES

RISCA and MATHNET are computer programs that allow one to analyze systems that can be represented by a general class of networks. Since the events and activity times and/or costs can be modeled probabilistically, a simulation process is utilized. The output consists of a frequency of occurrence distribution for each of all possible terminal events and corresponding time and/or cost distribution for each terminal event. In addition, the distribution of time and/or cost weighted over all possible terminal events is estimated.

Many of the benefits derived from analyzing a network result from the analysis and thinking that is required in the construction of the network. Consider, for example, the development of a tank where there are several alternative designs. Describing the sequence of events for alternate development programs for each design provides insight into the types of problems that one is likely to encounter in each program.

The oil changing example (Figure 1.5) from the introduction is utilized to demonstrate the characteristics and capabilities of MATHNET and RISCA.
Since in this example the event, "removing the oil filter," is the only uncertain event, the probabilistic event network can be described in terms of two deterministic-event sub-networks. One sub-network represents the events and activities involved in changing the oil and the oil filter and the other represents the events and activities involved in changing only the oil. In this simple example there is only one path in each sub-network; however, in more realistic problems there will almost certainly be several possible paths in a sub-network.

Monte Carlo procedures are used to determine which deterministic event sub-network will be followed in the probabilistic event network. Each of the sub-networks have a terminal event whose completion time is determined by Monte Carloing all the activity completion time distributions within the sub-network. All of the potential paths in the sub-network are then investigated using these sample activity times. If the sub-network chosen in the oil changing example involves removal of the oil filter, then the activity time distributions in this sub-network would be randomly sampled. These sample values would then be summed to estimate the sub-network completion time. In addition, the cost of all activities in the sub-network would be sampled and summed to estimate completion costs. However, in this example costs were not considered.

The preceding procedure is repeated many times, and the sampling distributions of terminal events and time and/or cost are constructed. It should be pointed out that in reality the deterministic event sub-network is chosen and the corresponding time and cost estimates for each activity on this network are accumulated as the network is simulated.

For this example, assume that 300 iterations have been run. Both RISCA and MATHNET would provide the frequency histograms of the percentage of times each terminal event was selected as shown in Figure 2.1 and the completion time distributions shown in Figures 2.2, 2.3, and 2.4. In addition, RISCA would provide a cumulative distribution of time for each completion time distribution in Figures 2.2, 2.3, and 2.4. Further discussion, interpretation and comparison of the two programs' output are deferred to the Output Section.

If cost is considered, there are two options available: (1) the cost can be estimated independent of time by running a separate simulation or (2) the cost can be estimated as a linear function of time in the same simulation, i.e., cost = (fixed cost)+(variable cost)x(time).

In addition to the insight derived in structuring and simulating the network, this type of analysis can provide a framework for evaluating and consolidating relevant information for decision making purposes. If one is trying to select an alternative system, this type
Figure 2.1 Possible Terminal Events.

* Percent of times the terminal event was selected.
Figure 2.2 Frequency Histogram of Completion Times for Terminal Event G.
Figure 2.3  Frequency Histogram of Completion Times for Terminal Event G'
Figure 2.4 Frequency Histogram for Project Completion Times Weighted Over Terminal Events G and G'.
of analysis can provide a framework for evaluating and consolidating relevant information for decision making purposes. If one is trying to select an alternative system, this type of analysis can provide a reasonable framework for comparing time and cost trade-offs in light of possible terminal events for each possible alternative. Consider, for example, the development of a tank where there are alternate designs. These alternate designs could be compared on the chances of program failure (i.e., having to cancel the development) and the chances of developing the tank for a fixed number of dollars in a fixed period of time. In addition, if the chances of failure are great with a preferred alternative, then this analysis might serve as the basis for restructuring the development program to allow for parallel development of two designs thereby increasing the chances of successful development.

In addition, this type of analysis could provide a framework for examining the progress in a program and provide inputs to determine whether or not to continue the program or to assess the impact of program changes. Once the network has been constructed, it is a relatively simple matter to modify the basic structure or update the data. Therefore, this technique could provide an up-to-date statement of the chances of meeting program time and cost objectives.

Using these programs is no substitute for analysis. Major effort still must be applied in trying to realistically model the system as a network. These programs are only tools. If a conscientious job of modeling the system has been accomplished, then the output of these programs is useful for both planning and decision making. The uses of the output in a decision making context will be discussed in more detail in the Output Section.

3. RISCA AND MATHNET PROGRAM INPUT

3.1 Introduction.

This section describes how the elements of a network are coded and inputted into RISCA and MATHNET.

Included is a description of the arc and node notation, the program input formats, an example problem and a comparison of the two input decks used in the example.

It is important to note that all of the arc and node notation and the input formats described herein are the same for both RISCA and MATHNET. MATHNET, for time sharing mode, contains several additional control cards. The nature of these cards and their use will be fully explained when the input decks for the example are compared.
3.2 Arc and Node Notation.

As explained in the introduction, the two classes of symbols used in describing a network are arcs and nodes. An arc is used to connect two nodes and represents an activity. A node is used to represent a decision or the initiation or termination of an activity.

Each arc inputted to the RISCA* network analyzer program is characterized by:

a. An arc name
b. The name of the initial node
c. The name of the terminal node
d. A distribution of completion times
e. A distribution of activity costs (as a linear function of time)
f. The probability of successful completion

The following time distribution types are available:

a. Normal
b. Triangular
c. Uniform
d. Constant

Arcs may exist in one of the following states:

a. Idle - the activity associated with the arc has not been initiated

b. Initiated - the activity associated with the arc has been initiated, but its outcome has not been determined

c. Completed - the activity associated with the arc has been successfully completed

d. Unsuccessful - the activity associated with the arc has been initiated but not successfully completed

* This also applies to MATHNET
e. Utilized - the node which terminates the arc has been satisfied.

The conditions which determine in which of the above states an arc will be are discussed following a description of the node types.

Nodes represent major events or decisions. Each node inputted to the program is characterized by:

a. a node name
b. an input rule
c. an output rule

In addition, time and/or cost values are also assigned to the nodes. However, these values vary (except for constant distributions) with each iteration. The time and/or cost values assigned to the nodes are those derived from the arcs entering them. In the cases where more than one arc enters a node, the exact values assigned to the node are dependent upon that node's input rule. These rules are described in detail in this section.

In order for an activity to be completed, it must satisfy the conditions of the node that it enters. For example, assume that an activity is "developing a model" and the terminating event is "completion of the model". If the model is completed, the arc is said to be successful and the node "satisfied".

The rules which determine whether or not the nodes of the network are satisfied are the input rules. They are:

a. And Input - All input arcs must be successfully completed. The time value assigned to the node is the maximum of all input arcs. The cost value is the sum of the costs generated by each input arc.

b. Or Input - At least one input arc must be successfully completed. The time value assigned is the minimum of all input arcs successfully completed. The cost value is the sum of the costs generated by each input arc based upon the smallest time value.

When an activity has satisfied the input rule of a node, the nature of its exit from the node is dependent upon the node output rules. There are two such rules:

a. All - all output arcs are initiated simultaneously.

b. Probabilistic - Associated with each output arc is a probability. The sum of the probabilities exiting one node must equal one.
One important point should be stressed at this time. The probabilities assigned by the analyst to each arc emanating from a node are done so on node data cards. This probability that an arc will be initiated is not to be confused with the probability that an arc will be successfully completed. This latter information is entered on an arc data card.

There are five special nodes which do not conform to the input and output rules outlined above. They are:

a. Initial Node - used to initiate the simulation. It possesses no input arcs

b. Terminal Node - used to terminate simulation. It possesses no output arcs

c. 1/1 (one-to-one bar) Node - This node possesses N input arcs and N+1 output arcs. Associated with each input arc is a unique output arc. The extra output arc is a default arc. The time value assigned to this node is the maximum of the successfully completed input arcs. The cost is the sum of the cost values for each of the input arcs which are completed successfully. This time and cost assignment is true for the remaining node types in this discussion. This node is satisfied in one of two ways:

   (1) by having one successful input arc (all input arcs must be initiated) which will result in one output arc being initiated

   (2) by initiation of the default arc. This is done when all of the input arcs have been initiated but none successfully completed.

The one-to-one bar node is used to simulate the time, cost and probability of completion for competing activities. For example, if two contractors submit the time, costs and probabilities associated with producing a certain prototype, this data can be applied to two arcs entering a 1/1 Node. By examining the corresponding output arcs and the default arc, the developer can gain valuable insight from success and failure probabilities, as well as the time and cost involved with each alternative.

d. 1/1 (one-to-one) Node - This node possesses N input arcs. Associated with each input arc is a unique output arc. The node will be satisfied by one complete input arc. One output arc will be initiated, namely the one associated with the input arc satisfying the node. 1/1 Nodes are used to preserve the time and cost factors for two or more activities experiencing a common event (the event being represented by the 1/1 Node). For example, suppose two contractors are developing blueprints for a house and the cost and time for each contractor differs. If the probability that both will
be successfully completed is 1.0, each of these activities (arcs) may enter a common node (blueprint completion). As the arcs exit the node, the time and cost of each activity is preserved.

Unlike the I/I node, there is no default arc to be initiated when activities are unsuccessful. Therefore all arcs entering the I/I node must have a probability of completion 1.0. A probability of less than 1.0 may cause a premature termination of the simulation resulting in no output.

e. Preferred Node - This node possesses N input arcs and N+1 output arcs. Associated with each input arc is a unique output arc. The extra output arc is a default arc. The preferred node differs from a I/I node in that the input/output arc pairs are ordered by the users preference. This node is satisfied in two ways:

(1) If all the input arcs are initiated and at least one arc has been successfully completed the output arc associated with the most preferred of the complete input arcs will be initiated.

(2) If all the input arcs are in the unsuccessful state the default output is initiated.

It is important to note that this node will never be satisfied unless all input arcs have been initiated and are in either the complete or unsuccessful state.

This node is used to simulate the time, cost and probability of completion for competing activities when the activities can be listed by order of preference.

A standard set of symbols is used to identify the various node types and input/output rules in a schematic network representation. A listing of these node types and the symbols used to represent them is found in Table 3.1.

3.3 Input Format.

The format used to enter the arc and node data into the RISCA network analyzer program will be discussed.

The Data deck consists of 3 blocks:

Block I - Header Card

Block II - ARC Cards

Block III - Node Cards
### Table 3.1 Symbolic Representation of Input/Output Rules and Special Nodes

<table>
<thead>
<tr>
<th>Input Rules</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>AND</td>
<td>AND</td>
</tr>
<tr>
<td>OR</td>
<td>OR</td>
</tr>
<tr>
<td>Initial</td>
<td>INIT</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Output Rules</th>
<th>Symbol</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>ALL</td>
</tr>
<tr>
<td>Prob</td>
<td>PROB</td>
</tr>
<tr>
<td>Terminal</td>
<td>TERM</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Special Nodes</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>One-to-One</td>
<td>1/1</td>
</tr>
<tr>
<td>One-to-One Bar</td>
<td>1/1</td>
</tr>
<tr>
<td>Preferred</td>
<td>PREF</td>
</tr>
</tbody>
</table>
A description of each follows:

Block I: The header card is one card used to describe the network being analyzed. All 80 columns may be used.

Block II: The information required for each arc appears on one card which is formatted as follows:

<table>
<thead>
<tr>
<th>Info</th>
<th>Columns</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Arc Name</td>
<td>1-4</td>
<td>A4</td>
</tr>
<tr>
<td>b. Input Node</td>
<td>5-8</td>
<td>A4</td>
</tr>
<tr>
<td>c. Output Node</td>
<td>9-12</td>
<td>A4</td>
</tr>
<tr>
<td>d. Distribution</td>
<td>13</td>
<td>I1</td>
</tr>
</tbody>
</table>

In column 13, only integers 1, 2, 3, and 4 may be entered.

These integers are used to indicate one of four possible distribution types:

<table>
<thead>
<tr>
<th>Integer</th>
<th>Distribution Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Normal</td>
</tr>
<tr>
<td>2</td>
<td>Triangular</td>
</tr>
<tr>
<td>3</td>
<td>Uniform</td>
</tr>
<tr>
<td>4</td>
<td>Constant</td>
</tr>
</tbody>
</table>

e. First time distribution argument 14-23 F10.0

f. Second time distribution argument 24-33 F10.0

g. Third time distribution argument 34-43 F10.0

When the distribution is normal, the first time distribution argument is the mean and the second time distribution argument is the standard deviation. The third time distribution argument is left blank.
When the distribution is triangular, the first time distribution is the most optimistic value, the second time distribution argument is the most likely value and the third time distribution argument is the most pessimistic value.

When the distribution is uniform, the first time distribution argument is the optimistic value, the second is the pessimistic value and the third is blank.

In a constant distribution, the first time distribution argument is the constant value and the second and third arguments are blank.

h. Fixed cost 44-53 F10.0
i. Variable Cost Coefficient 54-63 F10.0
j. Probability of successful completion 64-73 F10.0

Both the fixed cost and variable cost coefficients are components of a linear equation which describes the cost as a function of time. This equation is written:

\[ \text{Cost} = \text{fixed cost} \times \text{variable cost coefficient} \times t \]

where \( t \) is time in appropriate units.

The probability of successful completion refers to the probability that an activity (arc) will reach its desired end (node).

The arc cards which comprise Block II may be inputed in any order. However, it is desirable to enter the arcs in the order in which they appear in the network to avoid confusion.

The end of Block II is indicated by a card following the last arc card in Block II with "RETU" in card columns 1-4.

Block III: The information required for each node is formatted as follows:

<table>
<thead>
<tr>
<th>Info</th>
<th>Columns</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. Node name</td>
<td>1-4</td>
<td>A4</td>
</tr>
<tr>
<td>b. Input rule</td>
<td>5</td>
<td>II</td>
</tr>
</tbody>
</table>
There are 6 possible integers (input rules) which appear in column 5:

<table>
<thead>
<tr>
<th>Integer Value</th>
<th>Input Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>And</td>
</tr>
<tr>
<td>2</td>
<td>Or</td>
</tr>
<tr>
<td>4</td>
<td>Initial</td>
</tr>
<tr>
<td>5</td>
<td>1/1</td>
</tr>
<tr>
<td>6</td>
<td>1/1</td>
</tr>
<tr>
<td>7</td>
<td>Preferred</td>
</tr>
</tbody>
</table>

c. Output rule 6 II

There are 6 possible integer values which can appear in column 6. Listed below is each integer and the indicated output rule:

<table>
<thead>
<tr>
<th>Integer Value</th>
<th>Output Rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>All</td>
</tr>
<tr>
<td>2</td>
<td>Prob</td>
</tr>
<tr>
<td>4</td>
<td>Terminal</td>
</tr>
<tr>
<td>5</td>
<td>1/1</td>
</tr>
<tr>
<td>6</td>
<td>1/1 Bar</td>
</tr>
<tr>
<td>7</td>
<td>Preferred</td>
</tr>
</tbody>
</table>

When the node card has an input/output rule 1/1, 1/1 or PREF, a second card with the following information is required:

Info

a. No. of Output Arcs 1-2 I2

b. Input Arc Names 3-6, 11-14, A4 19-22, etc.

c. Output Arc Names 7-10, 15-18, A4 23-26, etc.
Each input arc name and its corresponding output arc name will appear as pairs on this card. The first input arc name will appear first in columns 3-6 followed by the output arc name it initiates (columns 7-10). The second input and output arc pairs will appear in columns 11-14 and 15-18 respectively, and so on until all arc pairs are listed.

Since the default output arc has no corresponding input arc name (T7T and PREF), "ZZZZ" must be entered as the input arc name. The above information card must immediately follow the node card it describes.

When the node card has an output rule Prob (2), a second card with the following information is required:

<table>
<thead>
<tr>
<th>Info</th>
<th>Columns</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>a. No. of Output Arcs</td>
<td>1-2</td>
<td>I2</td>
</tr>
<tr>
<td>b. Output Arc Names</td>
<td>3-6, 13-16, 23-26, etc.</td>
<td>A4</td>
</tr>
<tr>
<td>c. Probabilities associated with</td>
<td>7-12, 17-22, 27-32, etc.</td>
<td>F6.3</td>
</tr>
</tbody>
</table>

On this card, the name of each output arc and the probability of its exiting appear in pairs similar to the input and output arc pairs of the previous card. For example, the name of the first output arc will appear in columns 3-6 followed by the probability that it will exit in columns 7-12 and so on until all arc name and probability pairs have been listed. This card must immediately follow the node card it describes.

The node cards in Block III may be inputted in any order. However, be certain that in the case of I/I, T/T, Preferred and Probability nodes that each node card is succeeded by its appropriate information card. Following the last node card in Block III is a card with "RETU" in card column 1-4 marking the end of the block.

In the next section a hypothetical problem will be described, structured and inputs prepared as a network for RISCA.

3.4 Example Problem.

The use of the program will be illustrated via an example problem containing a variety of arc and node types.
A worker is confronted with a new tardiness policy established by his employer. It has been decided that an employee who is late for work more than 10 percent of the next 500 working days will be docked accordingly. The amount that the worker will be docked will be the cumulative time lost due to lateness.

This announcement has prompted the worker to closely examine the various routes and hazards facing him each morning in order to evaluate his chances of being docked.

His analysis of the paths* for driving to work revealed the following critical areas:

a. A fork in the road about 12 minutes from home. It has been his experience that the shortest route is only possible 90 percent of the time due to hazardous road conditions. 10 percent of the time he must travel a considerably longer route.

b. Fuel Problems. The worker is assured of not running out of gas if he takes the shorter route. However, if he takes the longer, alternate route, there is an 80 percent chance that he will have to make a 5 minute fuel stop.

c. Rough Road. Because of the extreme punishment to his tires along a stretch of road beyond the service station, the worker feels that he has a 5 percent chance of getting a flat tire that would take 15 minutes to repair.

d. Rider Stop. Beyond the stretch of bumpy road is the home of the worker's friend who rides to work with him every day. Historical data reveals that:

(1) If the shortest route is taken, and no fuel stops or tire troubles are experienced, his rider will be waiting outside for a ride 100 percent of the time.

(2) If the alternate route is taken, and no fuel stops or tire troubles are experienced, his rider will be waiting outside 80 percent of the time.

(3) If he takes the alternate route, stops for gas and has no tire trouble, his rider will be waiting for him 60 percent of the time.

*Recall that a path is a sequence of activities connecting two events. In this instance the events are departing for work and arriving at work.
If he has a flat tire, regardless of previous routes or stops, his rider will only be waiting outside 40 percent of the time.

In each case when the rider is not outside waiting, he must stop and check to see whether he is inside or has obtained another rider. This is a 5 minute stop.

Figure 3.1 illustrates the alternative routes and hazards.

In order to determine which node and arc type should be used to simulate the route in a network, each critical area should be examined.

a. Fork in the road. This is clearly a probabilistic node. The probability that the arc representing the shorter route will emanate is .90 and the probability that the arc representing the longer route will emanate is .10.

b. Gas Station. A T/T node will be used to simulate this event although a probabilistic node could also be used. The arc entering this node represents the longer route path and a probability of completion of .8 will be assigned to it. In the event that the arc is not completed, the default arc representing a gas stop will be initiated.

c. Rough Road. Three T/T nodes will be used to represent this hazard. Three nodes are used instead of one in order to account for this event in each of the three deterministic subnetworks developed thus far. Which T/T node is used in each iteration is dependent upon which of the following alternatives precedes it:

   (1) Short route,

   (2) Long route, no gas stop,

   (3) Long route, gas stop.

An arc, representing each of these alternatives, will be assigned a probability of completion of .95. In each case, failure to complete the node (get through the bumpy road without a flat) will initiate a default arc. All three default arcs will enter the fix flat node which will have an OR input rule.

d. Rider Stop. This event will be represented by three T/T nodes and an AND/ALL node. The AND/ALL node represents picking up the rider when the short route is taken and there is no tire trouble. Recall that in this case, the rider is waiting outside 100 percent of the time.
Figure 3.1 Graphical Representation of the Driving to Work Problem.
In the other three instances, the probability that the rider will be waiting is less than 1.0. These probabilities are

<table>
<thead>
<tr>
<th>Event</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1) Long route, no gas stop, no tire trouble</td>
<td>.8</td>
</tr>
<tr>
<td>(2) Long route, gas stop, no tire trouble</td>
<td>.6</td>
</tr>
<tr>
<td>(3) Tire trouble</td>
<td>.4</td>
</tr>
</tbody>
</table>

Separate I/I nodes will be used in each of these instances with all three default arcs entering an OR/ALL node representing a stop to check on the rider.

e. Arrive at Work. Five AND/TERM nodes will be used to represent the worker’s arrival. Their time and probability values will represent the time and probability associated with 5 separate alternatives which are:

(1) Short route, no stops,
(2) Long route, no stops,
(3) Long route, gas stop,
(4) Stop to fix flat,
(5) Stop for rider.

Figure 3.2 illustrates the network representation of "Driving to Work" using the RISCA and MATHNET arc and node construction symbols.

Table 3.2 lists each of the time distribution arguments, the events to which they correspond, the distribution arguments, the events to which they correspond, the distribution type involved and the arc names assigned to each of these time consuming activities.

Now that the network has been graphically represented, the arcs and nodes have been labeled and the time distribution argument has been determined, the input deck can be prepared.

As explained earlier, there are some differences in MATHNET and RISCA with respect to input deck content.

The contents of the RISCA deck are the three blocks outlined previously. That is, the first card is BLOCK 1 consisting of one title card describing the network to be analyzed. This is followed by
Figure 3.2 Network Representation of the Driving to Work Problem.
<table>
<thead>
<tr>
<th>Activity</th>
<th>Distribution Type</th>
<th>Arc Name</th>
<th>Time Distribution</th>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td>From Home to Junction</td>
<td>Triangular</td>
<td>Arc1</td>
<td>10.0</td>
<td>12.0</td>
</tr>
<tr>
<td>Junction to Gas Station</td>
<td>Triangular</td>
<td>Arc2</td>
<td>5.0</td>
<td>6.0</td>
</tr>
<tr>
<td>(short route)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Junction to Gas Station</td>
<td>Triangular</td>
<td>Arc3</td>
<td>11.0</td>
<td>13.0</td>
</tr>
<tr>
<td>(long route)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gas Station Stop</td>
<td>Constant</td>
<td>Arc8</td>
<td>5.0</td>
<td></td>
</tr>
<tr>
<td>Gas Station to Bumpy Road</td>
<td>Arc6</td>
<td>3.0</td>
<td>4.0</td>
<td>5.0</td>
</tr>
<tr>
<td>to Bumpy Road</td>
<td>Arc7</td>
<td>3.0</td>
<td>4.0</td>
<td>5.0</td>
</tr>
<tr>
<td>Road</td>
<td>Triangular</td>
<td>Arc9</td>
<td>3.0</td>
<td>4.0</td>
</tr>
<tr>
<td>Stop to Change Flat Tire</td>
<td>Constant</td>
<td>AR15</td>
<td>15.0</td>
<td></td>
</tr>
<tr>
<td>Bumpy Road To Rider's House</td>
<td>AR10</td>
<td>4.0</td>
<td>5.0</td>
<td>6.0</td>
</tr>
<tr>
<td>AR12</td>
<td>4.0</td>
<td>5.0</td>
<td>6.0</td>
<td></td>
</tr>
<tr>
<td>AR14</td>
<td>4.0</td>
<td>5.0</td>
<td>6.0</td>
<td></td>
</tr>
<tr>
<td>AR16</td>
<td>4.0</td>
<td>5.0</td>
<td>6.0</td>
<td></td>
</tr>
<tr>
<td>Stop to Check Rider</td>
<td>Constant</td>
<td>AR23</td>
<td>5.0</td>
<td></td>
</tr>
<tr>
<td>Rider's House To Work</td>
<td>AR17</td>
<td>8.0</td>
<td>9.0</td>
<td>10.0</td>
</tr>
<tr>
<td>AR18</td>
<td>8.0</td>
<td>9.0</td>
<td>10.0</td>
<td></td>
</tr>
<tr>
<td>AR20</td>
<td>8.0</td>
<td>9.0</td>
<td>10.0</td>
<td></td>
</tr>
<tr>
<td>AR22</td>
<td>8.0</td>
<td>9.0</td>
<td>10.0</td>
<td></td>
</tr>
<tr>
<td>AR24</td>
<td>8.0</td>
<td>9.0</td>
<td>10.0</td>
<td></td>
</tr>
</tbody>
</table>
BLOCK II, the arc cards, the last card being a "RETU" and indicating termination of the block. Then comes BLOCK III, the node cards, the last card again being a "RETU" card indicating the end of the block.

The version of RISCA described in this manual is designed only to run in a batch mode while MATHNET is designed to run in either a batch or time shared mode. It should be emphasized that adapting RISCA for a time sharing environment is not a difficult task. However, it would probably not be possible to run these versions of MATHNET and RISCA in any batch processing mode without first adapting the program to the particular system. Similarly, it would probably not be possible to run this version of MATHNET in any time sharing system without adapting it to the system. Because the MATHNET program can be run in either mode, the input deck is designed to give operators working in time-share the capability of introducing either each section of the input deck with a card describing the section or a card describing an operation to be performed. When the program reads the introduction card, it is directed to that portion of the program which performs the indicated operation.

The first card in the MATHNET input deck is a card with either a 0 or a 1 punched in column I (format II). The 0 card indicates that the problem will be run in a time share environment, and a 1 indicates batch mode.

For the purposes of comparing RISCA and MATHNET inputs in this manual, only the input decks for a batch mode environment will be utilized (See Figure 3.3).

The information cards mentioned above are cards with an integer punched in column I (format II). The following is a list of each integer which may be used, and the information card it indicates:

<table>
<thead>
<tr>
<th>Integer</th>
<th>Indicates</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Node cards follow</td>
</tr>
<tr>
<td>2</td>
<td>Arc cards follow</td>
</tr>
<tr>
<td>3</td>
<td>Net iteration number is set</td>
</tr>
<tr>
<td>4</td>
<td>Net is scanned</td>
</tr>
<tr>
<td>5</td>
<td>Net is run</td>
</tr>
<tr>
<td>6</td>
<td>Identification card follows</td>
</tr>
<tr>
<td>9</td>
<td>Run is to be completed (Ends Session)</td>
</tr>
</tbody>
</table>

36
Note that the number of iterations is controlled by the operator in MATHNET. In RISCA, the number of iterations is fixed at 500. It is conceivable that only 500 iterations could be restrictive in a network having a large number of terminal nodes. However, as was mentioned above, the program can be easily expanded to allow for more iterations. The card following the information card with a 3 in column 1 is a card with an integer value in columns 1-5 (format 15) specifying the number of iterations (from 1 to 1000).

The information cards with integers 4, 5 and 9 punched in column 1 do not introduce input cards, but they indicate to the program that one of three operations are to be performed. These operations are:

a. Scan the net - A scan of the net means that all arc and node characteristics are printed out in a tabular format (see Section 4).

b. Run the net - The net simulation is done.

c. Run is Completed - All output from the simulation is printed out. (See Section 4).

The arc, node and identification cards are formatted in the same manner as RISCA. However, the order in which they are inputted is not as restrictive as RISCA. The data cards must be entered before the operation cards, but the order in which the data cards are inputted is not important. The operation cards, however, must be entered as follows:

a. Scan the net

b. Run the net

c. Complete the run.

Figure 3.3 provides a comparison of the RISCA and MATHNET input decks used in the example problem. This figure illustrates the fact that with the exception of the MATHNET control cards discussed above, the two decks are identical.

Since the RISCA input deck has already been discussed in detail, and the fact that its MATHNET counterparts are identical has been established, only an explanation of the MATHNET control cards in the example problem input deck remains to be covered.

The first control card indicates that the problem will be run in a batch mode environment (1 in column 1). The second card indicates that the problem identification card follows (6 in column 1). The third control card (3 in column 1) indicates that the number of iterations follows. The next card sets the number of iterations at 500. The following card indicates that the arc cards follow (2 in
column I). The next control card appears at the end of the arc card block and indicates that the node cards follow (1 in column I).

The last 3 cards in the MATHNET input deck control the running of the simulation. The first card initiates the scanning of the net (4 in column I). The second card indicates the simulation is to be run (5 in column I). The last card indicates that the run is to be completed and the results printed out (9 in column I).

It is important to note that the blanks which appear in the listing of the RISCA input deck in figure 3.3 were included for ease of comparison only and are not required.

It was pointed out above that there are certain restrictions as to the number of iterations permissible in these versions of MATHNET and RISCA. In addition to these iteration restrictions, there are certain other inherent restrictions which deserve mention. For a complete listing, see Table 3.3. These limits may be increased (through minor program modifications) to the upper bound of the computer memory core.

4. RISCA AND MATHNET PROGRAM OUTPUT

4.1 Introduction.

In this section, the output from MATHNET and RISCA for the going-to-work example will be described and compared. In addition, the potential uses of the output for decision making purposes will be discussed both in and out of the context of this example.

4.2 MATHNET Output.

The output of any MATHNET run consists of a detailed input listing, frequency histograms for completion time and cost for each of the terminal nodes, frequency histograms of completion times and cost weighted over all terminal nodes, and frequency histograms of the percent of time each terminal node was selected in the simulation. A complete output listing for MATHNET for the example is provided in Appendix IV.

The first four pages of output consists of a printout of the dialogue between the user and the program, and a detailed description of the network representation being simulated. (See the Input Section). This dialogue is really only applicable to the terminal user. It provides him with a detailed set of instructions for inputting his network representation. This example was run in a batch mode, therefore the first data card inputted into the program had to have a 1 in the first column of the first card. On the two pages following the dialogue there is a detailed listing of the arc and node input data. This is particularly
### TABLE 3.3 Restrictions to MATHNET and RISCA

<table>
<thead>
<tr>
<th></th>
<th>MATHNET</th>
<th>RISCA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maximum number of nodes</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Maximum number of arcs</td>
<td>500</td>
<td>100</td>
</tr>
<tr>
<td>Maximum number of arcs into, or out of a single node</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Maximum number of initial nodes</td>
<td>10</td>
<td>10</td>
</tr>
<tr>
<td>Maximum number of terminal nodes</td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td>Maximum number of iterations</td>
<td>1000</td>
<td>500</td>
</tr>
</tbody>
</table>
useful information for checking the network. The format of the arc and node data will not be discussed since this is the same format that was discussed in the Input section.

Next, for each of the terminal nodes (ND 16, 17, ND 18, ND 19, ND 20) a frequency histogram of the completion time is provided. In this example, costs were not considered, but if they were then there would be a frequency histogram of cost for each terminal node. In addition, each terminal node represents the event "arriving at work" however, in each instance the path (i.e. sequence of events) is different. For example, the first histogram is for the completion times for ND 16. ND 16 is the short route to work with no time delay associated with hazards, stopping for gas, or picking up a rider. The horizontal axis is the frequency of occurrence and the vertical axis is time in minutes.** Looking at this graph, the probability of arriving at work within 35.6 and 35.9 minutes given this route is taken is .087. The interpretation of all the remaining frequency histograms is the same with two exceptions. One is that if the terminal cost frequency histograms were given then vertical axis would be cost. The other is that the final pair of frequency histograms for completion times and cost (where considered) is weighted over all possible terminal nodes (routes).

The final graph provided is the frequency plot of the percentage of times each terminal node was selected in the simulation. The horizontal axis again represents frequency of occurrence and the vertical axis lists the possible terminal nodes. Looking at the graph in Appendix IV, the probability of arriving at work having taken route ND 17 is .046, or ND 17 was chosen in 4.6 percent of the simulations.

4.3 RISCA Output.

The output of any RISCA run consists of a detailed input listing, frequency and cumulative frequency histograms for completion times and cost for each of the terminal nodes, frequency and cumulative frequency histograms of completion times and cost weighted over all terminal nodes, and a frequency histogram of the percent of times each terminal node was selected in the simulation. A complete listing of RISCA for this example is provided in Appendix V.

---

* On each time and cost frequency histogram the following statistics are provided: mean, variance, median, and mode.

** The time and cost units are selected by the analyst and depend upon the problem.
The first page of output consists of a detailed listing of the arc and node input data. This is particularly useful information for checking the network. Once again, the format of the arc and node data will not be discussed since this is the same format that was discussed in the Input Section.

Next, for each of the terminal nodes (ND 16, ND 17, ND 18, ND 19, ND 20) a frequency and cumulative frequency histogram* of the completion times and cost is provided. It is emphasized again that costs were not considered. In addition, each terminal node represents the event arriving at work, but in each instance the path (i.e., sequence of events) is different. Since the interpretation of the frequency histograms is the same as for MATHNET, only the cumulative frequency histograms will be discussed. For example, the first histogram is for the completion times for ND 16. It is followed by the cumulative frequency histograms for completion times for ND 16. The vertical axis is time** to completion, and the horizontal axis is the probability that the true time is less than or equal to the time on the vertical axis. For instance, the probability of arriving at work in less than 35.647 minutes given that this path was taken is .401. The interpretation of all of the remaining frequency and cumulative frequency histograms is the same with two exceptions. One is that for cost frequency histograms, the vertical axis is cost**. The other is that the final pair of frequency and cumulative frequency histograms for completion times and cost is weighted over all possible terminal nodes (routes).

The final graph provided is the frequency plot of the percent of time each terminal node was selected in the simulation. The interpretation of this histogram is identical to the MATHNET interpretation of the same histogram.

4.4 MATHNET and RISCA Output Comparison.

There are only two differences in the output of these programs. One is that RISCA provides cumulative frequency histograms for time and cost, for terminal nodes, and MATHNET does not. Clearly, the cumulative frequency histogram is required information in most applications as for example, in estimating the probability of meeting program time and cost goals. Consequently, there is an advantage in using this version of RISCA over this version of MATHNET since it eliminates the need for manually generating cumulative frequency histograms.

* On each time and cost histogram and following statistics are provided: mean, variance, and standard deviation.

** The time and cost units are selected by the analyst and depend upon the problem.
The other difference is in the output statistics computed for each time and cost distribution. In MATHNET, the mean, variance, median and mode are computed while in RISCA the mean, variance and standard deviation are computed. On first glance it may appear that one is given more information with the MATHNET statistics, but this is not the case. Since, if one is given the frequency and cumulative frequency histogram for any terminal node, determining the mode and median is a minor operation.

4.5 Use of the Output.

While there are more ways to use the output of a network analysis than will be discussed in this section, the two decision problems that are discussed probably represent the more popular applications. Before describing these problems and how to use the output for decision making purposes several general comments should be reiterated.

First, the greatest benefit to be derived in using a network analyzer program comes from the effort that is put into modeling the project or system as a network. Using this type of tool forces one to examine all possible events and the interaction of these events in the program. Further, representing a system as a network allows one to handle all of the relevant decision information in a systematic and composite fashion and to evaluate the impact of interactions that would be otherwise impossible.

Next, no matter what the decision is, there will always be a need for an overall framework for consolidating all the information. Network analyzer programs such as MATHNET and RISCA provide such a framework.

Finally, once the network has been structured, future modifications to the network are a relatively simple matter. In many instances, this type of analysis should not be a one time effort. The network representation and time and cost estimates should be modified on a periodic basis because as time passes, more information is gained and the initial network may no longer realistically represent the system.

In the remainder of this section, the use of network analysis for two general decision problems and the going to work example will be discussed.

The first decision problem either is one where the program* status must be evaluated periodically to determine whether or not to

---

* Program refers to a development program for a system.
continue or one where the impact of a program change must be evaluated. The discussion that follows will only be in terms of the decision to continue for brevity purposes, although network analysis could be used to help in a program change decision. A good example of a program would be a weapon system development. Clearly, the decision to continue depends on the chances of successfully developing the system within time and budget constraints.

A network representation for a development program would probably consist of two sets of outcomes. One set would constitute successful development, and the other set would constitute failure. In this case, the chances of successful development equals the sum of the percent of times each successful terminal was selected in the simulation. For any successful outcome the probability of developing the system within time and cost constraints may then be evaluated. This information is taken directly from the RISCA output (i.e. frequency histogram of the percent of times the terminal nodes were selected and the appropriate terminal node cumulative time and cost histograms). In addition, without a great deal of effort, the frequency and cumulative frequency histograms for time and cost weighted over all successful outcomes are estimated. This is done by either modifying the network representation so that all successful terminal events feed into one terminal event (successful development), or it may be reconstructed from the detailed frequency histograms of the successful outcomes.

This type of information should give the decision maker a foundation upon which to decide whether to continue with the program in light of the risks. Of course this doesn't relieve the burden of decision making, but it should enable the decision maker to make a more informed decision. Further, as the program continues, the network could be updated periodically to evaluate the program status for any future program decisions.

The other decision problem is one in which a choice between alternative systems for meeting a particular set of requirements* must be made. For this type of problem, a network could be structured for each alternative.** Each of these networks could then be simulated and the chances of successfully developing each alternative system and the corresponding development costs and completion times could be estimated. Given this information, the analyst has several relative measures for comparing the alternatives, and the decision maker should

* This assumes that "meeting a particular set of requirements..." generates equal and required effectiveness.

** It is assumed that a system is to be developed.
be able to systematically consider time and cost trade-offs in light of the risks.

Even though these two hypothetical decision problems are over-simplified, they are representative of many of the decisions that must be made within AMC. In both of these decision problems there is a great need for an overall framework for the evaluation. All too frequently this framework seems to be lacking. Network analysis provides an overall framework for synthesizing a large portion of the decision information in a systematic fashion that takes into account the known uncertainties. It is not meant to imply that the network analysis is the answer to all of the decision makers problems, but reasonable applications are a step in the right direction.

Returning now to the going to work example: How can this analysis help the worker in achieving his objective? First, it is assumed that the objective is not to be docked for tardiness. Since in all instances the worker arrives at work, the distribution of interest is the arrival time distribution weighted over all possible outcomes. Assuming the workman continues to leave his house at 0705 every morning and work starts at 0745, the probability of his being late is .157 or 1.0 minus the probability of arriving at work in less than or equal to 40 minutes (.843) (See the cumulative frequency histogram for arrival times weighted over all outcomes in Appendix V.)

If the probability of being late on any given day that the worker departs at 0705 is .157, the risk of being late over 10 percent of the next 500 working days is the probability of being late 51 or more times. Let $\ell$ be the random variable representing the number of late arrivals and $p$ the probability of being late on a given day, then

$$P[51 \leq \ell \leq 500] = \sum_{\ell=51}^{500} \binom{500}{\ell} p^{\ell} (1-p)^{500-\ell}$$

is the risk of being late in excess of 10 percent of the time.

If the worker leaves at 0705, the above calculation results in a risk of .99($p=.157$).

Using this method it is possible to derive a risk-of-being-docked profile for the worker as a function of departure time (See Figure 4.1).

What should the worker do? The answer is not clear cut. It depends upon the worker's assessment of the value of additional rest versus the potential financial impact of being docked. However, given the risk profile, the worker is in a position to consider the trade-offs. For example, if the worker cannot possibly afford to be docked, he can select a departure time from the risk profile where the risk is approximately zero (i.e., leave at 0654 or earlier).
Thus, it is seen in this simple example that, through the modeling of a network using RISCA or MATHNET, the analyst gains significant insight into the risks inherent in a system composed of various alternatives. This insight into the risks provides the information for making decisions concerning which alternative to choose, or decisions involving the various cost and/or time parameters (in this example departure time is the deciding factor).

Of course, much more complex systems containing a myriad of activities, events and alternatives may be modeled using these modeling tools. The scope and complexity of each network is only constrained by the number of arcs and nodes that can be inputted.

5. SUMMARY

MATHNET and RISCA are programs for simulating a class of networks where both the events and activities can be modeled probabilistically. These programs should provide the analyst with a tool for evaluating the status of existing programs, for evaluating the impact of proposed program changes on the total program, for deciding whether to continue the program, and for deciding between alternative systems. These decisions may not be mutually exclusive.

Since RISCA is a modification of MATHNET, there are very few differences in the input and output of the programs, and there are no differences in the method of simulating the networks. As seen in the Output Section, the additional features of this version of RISCA's output to this version of MATHNET's output are not really significant.

Throughout this report R&D applications have been emphasized, however, these techniques obviously have broader applications.
BIBLIOGRAPHY

A Course of Instruction in Risk Analysis, ALM-3315-H, United States Army Logistics Management Center, Fort Lee, Virginia, UNCLASSIFIED.

APPENDIX I

GENERAL FLOW CHART AND LISTING OF MATHNET

Next page is blank.
This appendix includes a general processing flow chart for MATHNET, but it does not include a description or flow chart of the individual subroutines. A detailed description and flow chart of the individual subroutines are provided in Appendix III. In addition to the flow chart, a complete program listing is provided. This version of MATHNET is designed to be run in either a batch or time sharing mode.

MATHNET is written in FORTRAN IV. Even though FORTRAN IV is considered to be standard language, adapting the program for a particular computer will probably require minor program modifications. These modifications generally result from peculiarities of the given system.

There is one definite modification that the user must make before MATHNET can be run on his computer. To run the simulation it is necessary to generate uniform random numbers. Within the subroutine RANDU these random numbers are generated by calling the library subroutine peculiar to the particular machine. In this listing the library subroutine RANSET is called to generate the uniform random numbers. Therefore, the user must either call the appropriate uniform random number generator for his machine or the analyst must write his own uniform random number generating routine within the RANDU subroutine. The latter was done by ALMC in their modification of MATHNET. The interested reader is referred to Appendix II where the ALMC Program is listed.
START

IXXX = 65539
CALL RANDU(IXXX, IXYY, RVAL)
IXX = IXYY

WRITE INSTRUCTIONS FOR RUNNING NET (MATHNET)

READ INDV BATCH RUN...

READ MODTYP

IF INDV = 0

WRITE EXPLANATION OF BRANCHES

GO TO

MODTYP 1
CALL NODIN

MODTYP 2
CALL ARcin

MODTYP 3
CALL REPSET

MODTYP 4
CALL SCAN

MODTYP 5
CALL RUNSYS

MODTYP 6
CALL IDIN

MODTYP 9
CALL EXIT

STOP

Flow Chart of "MATHNET"
PROGRAM MATHW ( INPUT, TAPE5=INPUT, OUTPUT, TAPE6=OUTPUT, 
PUNCH, TAPE7=PUNCH, TAPE1, TAPE2, TAPE3, TAPE4)

******************************************************************************
MATHNET WAS DEVELOPED BY MATHEMATICA INC. FOR THE ARMY
RESEARCH OFFICE UNDER CONTRACT NUMBER DAHC 04 70 CO025.
ALTHOUGH THIS PROGRAM HAS BEEN EXTENSIVELY TESTED WITH A LARGE
NUMBER OF NETWORKS, EXHAUSTIVE TESTING OF ALL POSSIBLE CLASSES
OF NETWORKS IS COMPUTATIONALLY IMPractical. MATHEMATICA THEREFORE
MAKES NO GUARANTEES CONCERNING THE ACCURACY OF THE OUTPUT GENERATED.
BY A RUN OF THE MATHNET PROGRAM
IT WOULD BE APPRECIATED IF ANY NETWORKS WHICH PRODUCE ERRONEOUS
OUTPUT, OR ANY OTHER SYSTEM ERRORS, WOULD BE REPORTED TOO.
MR. STEPHEN ROBINSON
MATHEMATICA
ONE PALMER SQUARE
PRINCETON, NEW JERSEY 08540

SUGGESTIONS CONCERNING ADDITIONAL OUTPUT FEATURES, ARC
CHARACTERISTICS, OR NODE CAPABILITIES, ARE ALSO WELCOMED.
******************************************************************************

MATHNET - VERSION 1 MODIFICATION LEVEL 0 - AUGUST 1, 1970
******************************************************************************
THIS IS THE MAIN ROUTINE
COMMON/RANC/Ixx
COMMON/INDEV/ INDE

INITIALIZE RANDU
Ixx=65339
CALL RANDU(Ixxx, Ixyy, Rval)
Ixx=Ixyy
WRITE(6,20)

FIRST I WILL READ IN A RECORD FROM DATA SET 5 TO DETERMINE
IF WE ARE IN BATCH MODE OR NOT

20 FORMAT(IX, 39HIF YOU ARE RUNNING THIS FROM A TERMINAL)
WRITE(6,21)
21 FORMAT(IX, 45HPLEASE ENTER A 1, IF RUNNING BATCH YOU SHOULD)
WRITE(6,22)
22 FORMAT(IX, 28HHAVE ENTERED A 0)
WRITE(6,23)
23 FORMAT(IX, 12HFORMAT IS 11)
READ(5,24) INDO
24 FORMAT(I1)

NOW TEST MODE, IF BATCH SKIP EXPLANATIONS FROM NOW ON
IF(IDV,EQ,0) GO TO 7
WRITE(6,30)
30 FORMAT(IX, 53HYOU ARE NOW IN MONITOR MODE, FROM THE FOLLOWING LIST)
WRITE(6,31)
31 FORMAT(IX, 35HSELECT THE MODE YOU WISH TO GO INTO)
WRITE(6,32)
32 FORMAT(1X, 16H1) ENTER NODE(S)
    WRITE(6,33) MAT00590
33 FORMAT(1X, 15H2) ENTER ARC(S)
    WRITE(6,34) MAT00600
34 FORMAT(1X, 25H3) SET ITERATION NUMBER)
    WRITE(6,35) MAT00610
35 FORMAT(1X, 24H4) SCAN THE NET SO FAR)
    WRITE(6,36) MAT00620
36 FORMAT(1X, 12H5) RUN NET)
    WRITE(6,37) MAT00630
37 FORMAT(1X, 25H6) ENTER RUN IDENTIFIER)
    WRITE(6,38) MAT00640
38 FORMAT(1X, 16H9) END SESSION)
    GO TO 7 MAT00650
77 WRITE(6,50) MAT00660
    WRITE(6,78) MAT00670
78 FORMAT(1X, 33H YOU HAVE RETURNED TO MONITOR MODE)
    WRITE(6,79) MAT00680
79 FORMAT(1X, 55H SELECT THE MODE YOU WISH TO GO INTO AS INDICATED BELOW)
    GO TO 7 MAT00690
40 FORMAT(1H *)
1 FORMAT(1I)
   IF(MODTP.EQ.1) CALL NODIN
   IF(MODTP.EQ.2) CALL ARCIN
   IF(MODTP.EQ.3) CALL REPSET
   IF(MODTP.EQ.4) CALL SCAN
   IF(MODTP.EQ.5) CALL RUNSYS
   IF(MODTP.EQ.6) CALL IDIN
   IF(MODTP.EQ.9) CALL EXIT
   GO TO 7 END

SUBROUTINE ARCIN
   .
   THIS ROUTINE WILL SERVE TO READ IN ARCS AND ASSOCIATED DATA
   COMMON/ INDEV/ INDV
   COMMON/ PARA/NODE, NARC
   COMMON/ ARC2/ TIME(500), ISTAT(500), PROB(500)
   COMMON/ ARC1/ ARC(500), INODE(500), ONODE(500), ITIME(500),
   CTARG(500), TARG2(500), TARG3(500), COSTC(500), COSTV(500)
   COMMON/ NODI/ NODE(100), IARC(100, 10), OARC(100, 10), PPJA(100, 10),
   CIAARC(100), OARC(100), IYPE(100), OYPE(100), MINDU(100),
   CTIM(100)
   REAL NODE
   INTEGER INODE
   INTEGER OARC1, OARC
   DATA RETC/4HRETU/
   IF(INDV.EQ.1) GO TO 50
   WRITE(6,60) MAT00920
   60 FORMAT(1X, 29H YOU ARE NOW IN ENTER ARC MODE)
   WRITE(6,61) MAT00930
   61 FORMAT(1H *)
WRITE(6,70)  
FORMAT(1X, 48HENTER ARC NAME, INPUT NODE NAME, OUTPUT NODE NAME,/)  
C 1X, 57HTIME DISTRIBUTION TYPE, TIME DISTRIBUTION ARGUMENTS 1, 2, 3, MAT01161  
C /1X, 71HCONSTANT COST COEFFICIENT, COEFFICIENT OF TIME TERM IN MAT01162  
C COST TERM, //1X, 40HPROBABILITY OF SUCCESSFUL ARC COMPLETION MAT01163  
C)  
WRITE(6,71)  
FORMAT(1X, 23HFORMAT IS 3A4, I1, 6F10.0)  
WRITE(6,61)  
WRITE(6,72)  
WRITE(6,61)  
WRITE(6,72)  
FORMAT(1X, 38HTO RETURN TO MONITOR MODE ENTER RETURN)  
READ(5,1) ANAME, AINODE, AOUTDE, IDIST, P1, P2, P3, C1, C2, D  
1 FORMAT(3A4, I1, 5F10.0, F9.0)  
IF(ANAME.EQ.RETC) RETURN  
NARC=NARC+1  
ARC(NARC)=ANAME  
ITIMET(NARC)=IDIST  
TARG1(NARC)=P1  
TARG2(NARC)=P2  
TARG3(NARC)=P3  
COSTC(NARC)=C1  
COSTV(NARC)=C2
SUBROUTINE NODIN
THIS ROUTINE WILL READ IN NODE NAMES AND DATA

COMMON/INDEV/INDV
COMMON/NODI/NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),
CIARC(100),JARC(100),ITYPE(100),OETYPE(100),MNIND(100),
CTIME1(100)
REAL NODE
DIMENSION DDUM(10)
COMMON/PARA/NNODE,NARC
COMMON/TERNV/NODN(30),NODI,TIMEZ(1000),COSTZ(1000),NODEZ(1000),
CNCOUNT(30)
COMMON/ARC2/TIME(500),ISTAT(500),PROB(500)
COMMON/ARC1/ARC(500),INODE(500),ONODE(500),ITIMET(500),TARG1(500),
CTARG2(500),TARG3(500),COSTG(500),COSTV(500)
DIMENSION ZNAM(10),ZPR0B(10),IZNAM(10),YI(10),YJ(10)
INTEGER OARC,OARCI
INTEGER OETYPE
COMMON/TERN/INODI(10),INODT
DATA CZZ/4HZZZZ/
DATA RETC/4HRETC/
TEST IF BATCH, IF SO SKIP GOBBELDYGOOK

IF(INDV.EQ.0) GO TO 20

TERMINAL MODE PRINT INSTRUCTIONS
WRITE(6,1)
1 FORMAT(1X, 26HYOU ARE IN ENTER NODE MODE)
WRITE(6,50)
50 FORMAT(1X, 38HYENTER NODE NAME, INPUT RULE, OUTPUT RULE)
WRITE(6,51)
51 FORMAT(1X, 18HFORMAT IS A4, I1, I1)
WRITE(6,52)
52 FORMAT(1X, 37HINPUT AND OUTPUT RULES ARE AS FOLLOWS)
WRITE(6,53)
53 FORMAT(10X, 11HRULE NUMBER,10X, 10HINPUT RULE,5X, 11HOUTPUT RULE)
WRITE(6,540)
540 FORMAT(9X,13(1H.)), 8X, 12(1H.)), 13X,13(1H.))
WRITE(6,54)
54 FORMAT(15X,1H1,18X, 3HAND,20X, 8HALL FIRE)
WRITE(6,55)
55 FORMAT(15X,1H2,18X, 2HOR,21X, 10HPROB. FIRE)
WRITE(6,56)
56 FORMAT(15X,1H4,18X, 7HINITIAL,16X, 8HTERMINAL)
WRITE(6,57)
57 FORMAT(15X,1H5,18X, 3H1/1,20X, 3H1/1)
WRITE(6,58)
58 FORMAT(15X,1H6,18X, 7H1/1 BAR,16X, 7H1/1 BAR)
WRITE(6,59)
59 FORMAT(15X,1H7,18X, 9HPREFERRED,14X, 9HPREFERRED)
20 WRITE(6,30)
30 FORMAT(2H *)
READ(5,40) ADUM, IDUM, IODUM
40 FORMAT(A4,I1,I1)
41 IF(ADUM.EQ.RETC) RETURN
DO 100 I=1,NNODE
NDUM=I
42 IF(NODE(I).EQ.ADUM) GO TO 101
100 CONTINUE
NDUM=NNODE+1
101 ITYPE(NDUM)=IDUM
222 ITYPE=4 MEANS INITIAL NODE
OTYPE(NDUM)=IODUM
43 IF(ITYPE(NDUM).EQ.4) GO TO 90
44 IF(ITYPE(NDUM).EQ.6) GO TO 400
45 ITYPE=6 MEANS WE HAVE A 1/1 NODE WITH A NEGATIVE INPUT
IF(ITYPE(NDUM).EQ.5) GO TO 60
46 ITYPE= 5 MEANS A 1/1 NODE WHICH MEANS WE MUST READ IN ORDERINGS
IF(OTYPE(NDUM).EQ.2) GO TO 70
47 OTYPE=2 MEANS PROBABILISTIC FIRINGS.. WE MUST READ IN PROBABILITIES
48 IF(ITYPE(NDUM).EQ.4) GO TO 80
49 IF(ITYPE(NDUM).EQ.7) GO TO 600
GO TO 2 
60 CONTINUE
C INSERT READS HERE FOR 1/1 NODES
WRITE(6,300)
300 FORMAT(1X, 29HYOU HAVE INDICATED A 1/1 NODE)
WRITE(6,301)
301 FORMAT(1X, 52HINPUT NUMBER OF ARCS INPUT ARC NAME, OUTPUT ARC NAMEMAT02520
59
WRITE(6,302)
FORMAT(ix, 23HF0RMAT IS I2,10(A4,A4 ) )
WRITE(6,30)
READ(5,103) MM,(YI(I),YO(I),I=1,MM)
203 FORMAT(12,i(1,A4,A4))
204 FORMAT(12.i(A4,F6.3))
DO 305 L=1,NARC
DO 306 K=1,MM
IF(IARC(L),EQ,YI(K)) IARC(NDUM,K)=L
IF(IARC(L),EQ,YO(K)) OARC(NDUM,K)=L
306 CONTINUE
305 CONTINUE
GO TO 2
70 CONTINUE
C INSERT READ FOR PROBABILITIES
IF(INDV,EQ.0) GO TO 220
WRITE(6,200)
200 FORMAT(ix, 49HYOU HAVE INDICATED A NODE WITH STOCHASTIC OUTPUTS)
WRITE(6,201)
201 FORMAT(ix, 53HINPUT NUMBER OF OUTPUT ARCS NAME OF OUTPUT ARC,PROB.
C )
WRITE(6,202)
202 FORMAT(ix, 25HF0RMAT IS I2,1G(A4,F6.3) )
WRITE(6,30)
220 READ(5,203) NN,(ZNAM(I),ZPRcB(I),I=1,NN)
IF(NN.NE.OARC(NDUM)) CALL TERM(99)
DO 205 I=1,NN
DO 206 J=1,NN
JJ=J
LM=OARC(NDUM,I)
IF(ZNAM(J).EQ.ARC(LM)) GO TO 2G7
206 CONTINUE
CALL TERM(100)
207 PPOA(NDUM,I)=ZPRcB(JJ)
205 CONTINUE
GO TO 2
80 NODI=NODI+1
NODN(NODI)=NDUM
GO TO 2
90 INODT=INODT+1
INODI(INODT)=NDUM
GO TO 222
400 IF(INDV,EQ.0) GO TO 420
WRITE(6,401)
401 FORMAT(ix, 33HYOU HAVE INDICATED A 1/1 BAR NODE)
WRITE(6,402)
402 FORMAT(ix, 51HINPUT NUMBER OF ARCS NAME OF INPUT ARC,NAME OF OUTPUT ARC
C NAME OF PROB.)
WRITE(6,403)
403 FORMAT(ix, 47HINPUT A NAME OF ZZZZ FOR THE NO INPUT CONDITION)
WRITE(6,302)
WRITE(6,30)
420 READ(5,303) MM,(YI(I),YO(I),I=1,MM)
DO 405 L=1,NARC
DO 406 K=1,MM
IF(IARC(L),EQ,YI(K)) IARC(NDUM,K)=L
IF(IARC(L),EQ,YO(K)) OARC(NDUM,K)=L
IF(YI(K),EQ.ZZ) IARC(NDUM,K)=500
406 CONTINUE
405 CONTINUE
IARC(NDUM)=IARC(NDUM)+1
GO TO 20

600 IF (INDV.EQ.0) GO TO 666
WRITE(6,601)
601 FORMAT (1X, 35H YOU HAVE INDICATED A PREFERRED NODE)
WRITE(6,302)
WRITE(6,602)

602 FORMAT (1X, 43H I/O ARC PAIRS SHOULD BE IN PREFERENCE ORDER)
WRITE(6,403)
WRITE(6,30)

666 READ (5,303) MM, (YI(I), YO(I), I=1,MM)
DO 605 L=1,NARC
DO 606 K=1,MM
IF (ARC(L).EQ.YI(K)) IARC(NDUM,K)=L
IF (ARC(L).EQ.YO(K)) OARC(NDUM,K)=L
IF (YI(K).EQ.CZZ) IARC(NDUM,K)=500
606 CONTINUE
605 CONTINUE
IARC(NDUM)=IARC(NDUM)+1
GO TO 20
END

SUBROUTINE IDIN
COMMON/IDD/ RUNID(20)
WRITE(6,1)
1 FORMAT (1X, 47H ENTER A RUN IDENTIFIER OF 80 CHARACTERS OR LESS)
READ (5,2) RUNID
2 FORMAT (20A4)
RETURN
END

SUBROUTINE REPSET
COMMON/ITERA/ ITER
COMMON/INDEV/ INDV
C TEST IF IN BATCH MODE, IF SO SKIP GOBBELDY GOOK
IF (INDV.EQ.0) GO TO 4
WRITE(6,1)
1 FORMAT (1X, 40H YOU CAN NOW SET THE NUMBER OF ITERATIONS)
WRITE(6,2)
2 FORMAT (1X, 42H ENTER A 5 POSITION INTEGER, RIGHT ADJUSTED)
WRITE(6,3)
3 FORMAT (1X,1H*)
4 READ(5,5) ITER
5 FORMAT (15)
RETURN
END

SUBROUTINE SCAN
C THIS ROUTINE WILL PRINT OUT THE NET TO DATE
COMMON/RANC/ IXX
COMMON /ARC2/ TIME(500), ISTAT(500), PROB(500)
COMMON/ARC1/ARC(500), INODE(500), ONODE(500), ITIMET(500), TARG1(500), MAT03120
MAT03130
MAT03140
MAT03150
MAT03160
MAT03170
MAT03180
MAT03190
MAT03200
MAT03210
MAT03220
MAT03230
MAT03240
MAT03250
MAT03260
MAT03270
MAT03280
MAT03290
MAT03300
MAT03310
MAT03320
MAT03330
MAT03340
MAT03350
MAT03360
MAT03370
MAT03380
MAT03390
MAT03400
MAT03410
MAT03420
MAT03430
MAT03440
MAT03450
MAT03460
MAT03470
MAT03480
MAT03490
MAT03500
MAT03510
MAT03520
MAT03530
MAT03540
MAT03550
MAT03560
MAT03570
MAT03580
MAT03590
SUBROUTINE RUNSYS

THIS ROUTINE CONTROLS THE RUNNING OF A NET

COMMON/IRZZ/IREPIT
COMMON/KIND/KIND
COMMON/ARC1/ARC(500), INODE(500), ONODE(500), ITIMET(500), TARG1(500), COSTC(500)
CTARG2(500), TARG3(500), COSTC(500), COSTV(500)
COMMON/ITERA/ITER
COMMON/ARC2/TIME(500), ISTAT(500), PROB(500)
COMMON/RUNER/TERMS
COMMON/HINT/SMTIM, INSM
COMMON/TERIN/NODN(30), NODI, TIMEZ(1000), COSTZ(1000), NODEZ(1000),
CNCOUNT(30)
COMMON/NOD1/ NODE(100), IARC(100, 10), OARC1(100, 10), PPJA(100, 10),
CIARC1(100), OARC(100), ITYPE(100), OTYPE(100), MNIND(100),
CTIME(100)
COMMON/para/NODE, NARC
COMMON/TERNI/ INODI(10), INODT
INTEGER OARC, OARC1, OTYPE

C SET NUMBER OF ITERATIONS SO FAR TO 0
IREPIT=0
C
SMTIM=999999.0

ABOVE INITIALIZES SMALLEST TERMINAL FINISH TIME TO A LARGE NUMBER

THE FOLLOWING CODE SETS THE INDICATIVE OF WHETHER OR NOT A TERMINAL NODE HAS BEEN FILLED TO INDICATE NO.

ITERMS=0

NOW FIRE INITIAL NODES

DO 10 I=1,INODT
LM=INODI(I)
TIMEN(LM)=0.
IF(CTYPE(LM).EQ.1)
   IF(OTYPt(LM).EQ.2) CALL ALLFIR(LM)
   IF(OTYPt(LM).EQ.1) CALL PROFIR(LM)
10 CONTINUE

CALL ARCCHK
CALL NODCHK
IF(KIND.EQ.0) GO TO 60
IF(ITERMS.EQ.0) GO TO 1
CALL ENDIT(KEY)
IF(KEY.EQ.0) GO TO 10

IREPIT=IREPIT+1
IF(IREPIT.EQ.ITER) GO TO 61
IF(IREPIT.EQ.ITER) DO 40 I=1,NARC
TIME(I)=0.
ISTATU(I)=0
CONTINUE
GO TO 20

COME HERE IF NO NODES HAVE FIRED

IF(ITERMS.EQ.0) CALL TERNAL(3030)

DO ABOVE IF NO TERMINAL NODES HAVE BEEN FILLED

GO TO 61

DO ABOVE IF THERE IS A TERMINAL NODE-ENDING ITERATION

SUBROUTINE NODCHK

SEE WHAT NODES ARE READY TO FIRE, FIRE THOSE THAT ARE READY

COMMON/PARA/NNODE,NARC
COMMON/NODI/ NODE(100),IARC(100,10),OARC(100,10),PPQA(100,10),
CIARC(100),OARCI(100),ITYPE(100),O TYPE(100),MNIND(100),
CTIMEN(100)
COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)

INTEGER OTYPE,OARCI,OARC
COMMON /KKIND/KIND
REAL NODE
KIND=0
DO 1 I=1,NNODE
   IL=I
   J=0
   IF(ITYPE(I).NE.1) GO TO 40
SUBROUTINE ANDFST(I,J)
   CALL ANDFST(IL,J)
   GO TO 100

 40   IF(IY(I),NE.2) GOTO 41
   CALL GATST(II,J)
   GO TO 100

 41   CONTINUE
   IF(IY(I),EQ.5) GOTO 52
   IF(IY(I),EQ.6) GOTO 53
   IF(IY(I),EQ.7) GOTO 54

 100   IF(J,NE.0) GOTO 1
   KIND=KIND+1

   CALL ANDFST(IL,J)
   GO TO 100

   CONTINUE
   IF(IY(I),NE.1) GOTO 50
   CALL ALLFIR(I)
   GO TO 1

 50   IF(IY(I),NE.2) GOTO 51
   CALL PRFIR(I)
   GO TO 1

 51   IF(IY(I),NE.4) GOTO 52
   CALL ITALL(I)
   GO TO 51

 52   I=IARCI(I)
   DO 22 J=1,IRK
   LM=IARC(I,J)
   IF(ISTAT(LM),EQ.0) GOTO 22
   ISTAT(LM)=4

 22   CONTINUE
   IF(IY(I),EQ.1) GOTO 50
   CALL ALLFIR(I)
   GO TO 1

 53   IF(IY(I),NE.6) GOTO 54
   CALL ONEONE(I,J)
   KIND=KIND+J
   GO TO 1

 54   IF(IY(I),NE.7) GOTO 55
   CALL PREFER(I,J)
   KIND=KIND+J

 55   CONTINUE
   RETURN
   END

C

CALL ANDFST(II,J)
GO TO 100

40   IF(IY(I),NE.2) GOTO 41
CALL GATST(II,J)
GO TO 100

41   CONTINUE
IF(IY(I),EQ.5) GOTO 52
IF(IY(I),EQ.6) GOTO 53
IF(IY(I),EQ.7) GOTO 54

100   IF(J,NE.0) GOTO 1
KIND=KIND+1

CALL ANDFST(II,J)
GO TO 100

CONTINUE
IF(IY(I),NE.1) GOTO 50
CALL ALLFIR(I)
GO TO 1

50   IF(IY(I),NE.2) GOTO 51
CALL PRFIR(I)
GO TO 1

51   IF(IY(I),NE.4) GOTO 52
CALL ITALL(I)
GO TO 51

52   I=IARCI(I)
DO 22 J=1,IRK
LM=IARC(I,J)
IF(ISTAT(LM),EQ.0) GOTO 22
ISTAT(LM)=4

22   CONTINUE
IF(IY(I),EQ.1) GOTO 50
CALL ALLFIR(I)
GO TO 1

53   IF(IY(I),NE.6) GOTO 54
CALL ONEONE(I,J)
KIND=KIND+J
GO TO 1

54   IF(IY(I),NE.7) GOTO 55
CALL PREFER(I,J)
KIND=KIND+J

CONTINUE
RETURN
END

SUBROUTINE ANDTST(I,J)
   THIS ROUTINE TESTS AND NODES
   COMMON/NODI/ NODE(100),IARC(100),OARC(100),PP0A(100),POCA(100),
   CIARC(100),OARCI(100),ITYPE(100),OTYPE(100),MNIND(100),
   CTIMEN(I)
   REAL NODE
   COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)
   II=IARC(I)
   TIM=0.
   DO 1 K=1,II
   KK=IARC(/,K)
   IF(ISTAT(KK),NE.2) GOTO 2
   IF(TIME(KK),GT.TIM) TIM=TIME(KK)
1 CONTINUE
TIMEN(I) = TIM
J = 1
RETURN
2 J = 0
RETURN
END

SUBROUTINE PROFIR(I)
FIRE NODE I USING STOCHASTIC CONSIDERATIONS
COMMON / ARC2/ TIME(500), ISTAT(500), PROB(500)
COMMON / RANC/ IXX
INTEGER OARC, OARCI, OTYPE

IXXX = IXX
CALL RANDU(IXXX, IXY, RVAL)
IXX = IXY
AHIGH = 0.
II = OARC(I)
DO 1 K = 1, II
KK = OARCI(I, K)
ALOW = AHIGH
AHIGH = ALOW * PPOA(I, K)
IF((RVAL.GE. ALOW). AND. (RVAL.LE. AHIGH)) GO TO 2
CONTINUE
CALL TERM(I)
1 ISTAT(KK) = 1
2 ISTAT(KK) = 1

SUBROUTINE DTST(T, J)
THIS ROUTINE TEST OR NODES
COMMON / NODE/ NODE(100), IARC(100, 10), OARC(100, 10), PPOA(100, 10),
CIARC(100), OARCI(100), ITYPE(100), OTYPE(100), MNIND(100),
CTIMEN(100)
REAL NODE
COMMON / ARC2/ TIME(500), ISTAT(500), PROB(500)
COMMON / RANC/ IX X
INTEGER OARC, OARCI, OTYPE
SUBROUTINE ALLFIR(I)
FIRE ALL OUTPUT ARCS
COMMON /ARC2/ TIME(500), ISTAT(500), PROB(500)
COMMON /NODE/ NODE(100), IARC(100,10), OARC(100,10), PPOA(100,10),
CIARC(100), ARC(100), ITYPE(100), OTYPE(100), MNIND(100),
CTIMEN(100)
REAL NODE
INTEGER OARC, IARC
II=OARC(I)
DO 1 K=1,II
KK=OARC(I,K)
ISTAT(KK)=1
1 CONTINUE
RETURN
END

SUBROUTINE ONEONE(ILK, J)
COMMON / PARA / NNODE, NARC
COMMON / NODE / NODE(100), IARC(100,10), OARC(100,10), PPOA(100,10),
CIARC(100), ARC(100), ITYPE(100), OTYPE(100), MNIND(100),
CTIMEN(100)
COMMON / ARC1 / ARC(500), NNODE(500), ONODE(500), ITIMET(500), TARG1(500),
CTARG2(500), TARG3(500), COSTC(500), COSTV(500)
COMMON / ARC2/ TIME(500), ISTAT(500), PROB(500)
INTEGER OTYPE, OARC, IARC
REAL NODE
J=0
II=IARC(ILK)
TIM=1000000.
DO 1 K=1,II
KK=IARC(ILK,K)
IF(ISTAT(KK).NE.2) GO TO 1
IF(TIME(KK).GE.TIM) GO TO 1
TIM=TIME(KK)
J=K
1 CONTINUE
IF(J.EQ.O) RETURN

ZERO OUT ALL INPUTS, FIRE J TH OUTPUT ARC
DO 2 K=1,II
LM=IARC(ILK,K)
IF(ISTAT(LM).EQ.0) GO TO 2
ISTAT(LM)=4
2 CONTINUE
LM=OARC(ILK,J)
ISTAT(LM)=1
TIMEN(ILK)=TIM
RETURN
END
SUBROUTINE ONEHAR(ILK, J)
COMMON/ PARA/ NNODE, NARC
COMMON/NODE/ NODE(100), IARC(100, 10), OARC(100, 10), PPOA(100, 10),
CMARC(100), DARC(100), DTYPE(100), OTYPE(100), MIND(100),
CTIMEN(100)
COMMON/ ARC/ ARC(500), NODE(500), NODE(500), TIMENG(500), TARG1(500),
CTARG2(500), TARG3(500), COSTC(500), COSTV(500)
COMMON/ ARC2/ TIME(500), ISTAT(500), PROB(500)
INTEGER OTYPE, OARC, IARC
REAL NODEN, OARC
J=0
II=IARC(ILK)
TIM=1000.0
DO 1 K=1,II
KK=IARC(ILK, K)
IF(KK.EQ.500) IBAR=K
IF(KK.EQ.500) GO TO 1
IF(ISTAT(KK).NE.2) GO TO 1
IF(TIME(KK).LT.TIM) GO TO 1
TIM=TIME(KK)
J=K
1 CONTINUE
IF(J.EQ.0) GO TO 5
C ZERO OUT ALL INPUTS, FIRE J TH OUTPUT ARC
DO 2 K=1,II
LM=IARC(ILK, K)
IF(ISTAT(LM).EQ.0) GO TO 2
ISTAT(LM)=4
2 CONTINUE
LM=OARC(ILK, J)
ISTAT(LM)=1
TIMEN(ILK)=TIM
RETURN
5 TIM=0.
DO 6 K=1,II
KK=IARC(ILK, K)
IF(KK.EQ.500) GO TO 6
IF(ISTAT(KK).NE.3) GO TO 7
IF(TIME(KK).LE.TIM) GO TO 6
TIM=TIME(KK)
6 CONTINUE
LM=OARC(ILK, IBAR)
ISTAT(LM)=1
J=1
DO 12 K=1,II
LM=IARC(ILK, K)
ISTAT(LM)=4
12 CONTINUE
TIMEN(ILK)=TIM
7 CONTINUE
RETURN
END

SUBROUTINE PREFER(ILK, J)
COMMON/ PARA/ NNODE, NARC
COMMON/NODE/ NODE(100), IARC(100, 10), OARC(100, 10), PPOA(100, 10),
C IARC(100), DARC(100), DTYPE(100), OTYPE(100), MIND(100),
CTIMENG(100)
COMMON/ ARC/ ARC(500), NODE(500), NODE(500), TIMENG(500), TARG1(500),
CTARG2(500), TARG3(500), COSTC(500), COSTV(500)
COMMON/ ARC2/ TIME(500), ISTAT(500), PROB(500)
INTEGER OTYPE, OARC, IARC
REAL NODEN, OARC
J=0
II=IARC(ILK)
TIM=1000.0
DO 1 K=1,II
KK=IARC(ILK, K)
IF(KK.EQ.500) IBAR=K
IF(KK.EQ.500) GO TO 1
IF(ISTAT(KK).NE.2) GO TO 1
IF(TIME(KK).LT.TIM) GO TO 1
TIM=TIME(KK)
J=K
1 CONTINUE
IF(J.EQ.0) GO TO 5
C ZERO OUT ALL INPUTS, FIRE J TH OUTPUT ARC
DO 2 K=1,II
LM=IARC(ILK, K)
IF(ISTAT(LM).EQ.0) GO TO 2
ISTAT(LM)=4
2 CONTINUE
LM=OARC(ILK, J)
ISTAT(LM)=1
TIMEN(ILK)=TIM
RETURN
5 TIM=0.
DO 6 K=1,II
KK=IARC(ILK, K)
IF(KK.EQ.500) GO TO 6
IF(ISTAT(KK).NE.3) GO TO 7
IF(TIME(KK).LE.TIM) GO TO 6
TIM=TIME(KK)
6 CONTINUE
LM=OARC(ILK, IBAR)
ISTAT(LM)=1
J=1
DO 12 K=1,II
LM=IARC(ILK, K)
ISTAT(LM)=4
12 CONTINUE
TIMEN(ILK)=TIM
7 CONTINUE
RETURN
END
SUBROUTINE ARCCHK
C      IF INITIATED CHECK PROBABILITY OF COMPLETION
C      IF COMPLETED CHECK CALCULATE TIME AND COST
C
COMMON/RANC/ IXX
COMMON /ARC1/ TIME(500), ISTAT(500), PROB(500)
COMMON /ARC/ TIME(500), ISTAT(500), PROB(500)
COMMON /ARC2/ TIME(500), ISTAT(500), PROB(500)
COMMON /ARC/ TIME(500), ISTAT(500), PROB(500)
COMMON /ARC1/ARC(500), INODE(500), ONODE(500), ITIME(T500), TARG1(500)
COMMON /ARC1/ARC(500), INODE(500), ONODE(500), ITIME(T500), TARG1(500)
COMMON /ARC1/ARC(500), INODE(500), ONODE(500), ITIME(T500), TARG1(500)
COMMON /ARC1/ARC(500), INODE(500), ONODE(500), ITIME(T500), TARG1(500)
REAL NODE

J=0
II=IARCI(ILK)
TIM=0.
DO 1 K=1,II
KK=IARC(ILK,K)
IF(KK.EQ.50) GO TO 1
IF(ISTAT(KK).EQ.0) GO TO 30
IF(ISTAT(KK).EQ.1) GO TO 30
IF(ISTAT(KK).EQ.4) GO TO 30
IF(TIME(KK).GT.TIM) TIM=TIME(KK)
1 CONTINUE
J=1
C IF WE GET HERE THE NODE WILL BE FIRED
C FIRE FIRST ARC PAIR WITH 2 STATUS, IF THERE IS ONE
III=II-1
DO 2 K=1,III
KK=IARC(ILK,K)
KKK=K
IF(ISTAT(KK).EQ.2) GO TO 4
2 CONTINUE
C IF WE GET HERE FIRE BAR ARC
LM=OARC(ILK,II)
GO TO 5
C FIRE THE KKK ARC
LM=OARC(ILK,KKK)
TIME(LM)=1
DO 40 K=1,II
LM=IARC(ILK,K)
ISTAT(LM)=4
40 CONTINUE
30 CONTINUE
TIME(IIK)=TIM
RETURN
END
DO 1 I=1,NARC
IF(ISTAT(I).NE.1) GO TO 1
IXXX=IXX
A=TARG1(I)
B=TARG2(I)
C=TARG3(I)

IF(ITIMET(I).EQ.1) CALL GAUSS(IXXX,B,A,TVAL)
IF(ITIMET(I).EQ.2) CALL TRIANG(IXXX,A,B,C,TVAL)
IF(ITIMET(I).EQ.3) CALL UNIFRM(IXXX,A,B,C,TVAL)
IXX=IXXX

LM=INOOE(I)
TIME(LM)=TVAL+TIMFN(LM)
IFPROB(I),J.EQ.1) GO TO 2

ISTAT(I)=2
CONTINUE
RETURN
END

SUBROUTINE ITALL(II)

C THIS ROUTINE WILL HANDLE A TERMINAL NODE BEING FILLED
C IT WILL SEE IF THE TIME IS SMALLER THAN ANY OTHER TERMINAL
C NODE TIME AND IF SO SWAP TIME AND COST INDICATORS
C
COMMON/MINT,SMTIM,INSM
COMMON/RUNER/ ITERMS
COMMON/NODI/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),
CIARC(100,10),ARC(100),ITYPE(100),OTYPE(100),MNIND(100),
CTIMEN(100)
ITERMS=1

C ABOVE INDICATES A TERMINAL NODE HAS BEEN FILLED
IF(TIMEN(II).GE.SMTIM) RETURN
SMTIM=TIMEN(II)
INSM=II
RETURN
END

SUBROUTINE ENEDIT(KEY)

C THIS ROUTINE CHECKS TO SEE IF THERE ARE COMPLETED ARCS WITH TIMES
C SMALLER THAN THE SMALLEST TERMINAL NODE
C IF SO SET KEY=0, IF NOT SET KEY=1 AND TERMINATE THIS RUN
COMMON /ARC2/ TIME(500),ISTAT(500),PROB(500)
COMMON/MINT/SMTIM,INSM
COMMON/PARA/NNODE,NARC
DO 1 I=1,NARC
IF(ISTAT(I).EQ.0) GO TO 1
IF(ISTAT(I).EQ.3) GO TO 1
1 CONTINUE
RETURN
END
IF (STAT(I).EQ.0) GO TO 1
IF (TIME(I).GT.SMTIM) GO TO 4
CONTINUE
KEY = 1
RETURN
2
KEY = 0
RETURN
END

SUBROUTINE PTERM
COMMON/ Para/ NNODE, NARC
COMMON/ IRRT/ IREPIT
COMMON/ MNOD/ SMTIM, INSM
COMMON/ NODI/ NODE(100), IARC(100,10), OARC(100,10), PPOA(100,10),
CIARC(100), ARC(100), ITYPE(100), OTYPE(100), MNIND(100),
CTIMEN(100)
COMMON/ TERN/ NODNI(30), NODI, TIME(1000), COSTZ(1000), NODE(100)
CNCOUNT(10)
COMMON/ ARC1/ ARC(500), INODE(500), ONODE(500), ITIME(500), TARG1(500),
CTARG2(500), TARG3(500), COSTC(500), COSTV(500)
COMMON / ARC2/ TIME(500), ISTAT(500), PROB(500)
COMMON/ TERN/ INODI(10), NODI 
DO 1 I = l, NODI
JJ = I
IF (NODN(I).EQ.INSM) GO TO 2
1
CONTINUE
CALL TERM(69)
2
NCOUNT(JJ) = NCOUNT(JJ) + 1
LM = NCOUNT(JJ)
IRR = IREPIT + 1
TIMEZ(IRR) = SMTIM
NODEZ(IRR) = INSM
COSTZ(IRR) = 0.
DO 3 I = 1, NARC
LM = NODE(I)
IF (STAT(I).EQ.0) GO TO 3
IF (TIME(I).GT.SMTIM) GO TO 4
COSTZ(IRR) = COSTZ(IRR) + COSTC(I) + COSTV(I) * (TIME(I) - TIMEN(LM))
GO TO 3
4
IF (TIME(LM).GT.SMTIM) GO TO 3
COSTZ(IRR) = COSTZ(IRR) + COSTC(I) + COSTV(I) * (SMTIM - TIMEN(LM))
3
CONTINUE
RETURN
END

SUBROUTINE TERM(I)
C
THIS ROUTINE WILL ACT AS AN ERROR TERMINATOR
COMMON/ARC1/ ARC(500), INODE(500), ONODE(500), ITIME(500),
CTARG1(500), TARG2(500), TARG3(500), COSTC(500), COSTV(500)
COMMON/ PARA/ NNODE, NARC
COMMON / ARC2/ TIME(500), ISTAT(500), PROB(500)
WRITE(6,1) I
1
FORMAT(1X, 31HEXECUTION TERMINATED FOR REASON,I5)
IF(I .EQ. 3030) GO TO 2
CALL EXIT
2 WRITE(6,3)
3 FORMAT(25H ARCS ACTIVATED THUS FAR )
   DO 4 J = 1,NARC
   IF(ISTAT(J) .NE. 0) WRITE(6,5) ARC(J)
4 FORMAT(1X,A4)
   CONTINUE
   CALL EXIT
   END

SUBROUTINE RANDU(Ix, IY, YFL)
   IF(Ix .NE. 65539) GO TO 10
   CALL RANSET(Y)
10 CONTINUE
   YFL = RANF(Y)
   YFL = ABS(YFL)
   IY = 1
   RETURN
END

SUBROUTINE GAUSS(Ix, S, AM, V)
   A = 0.0
   DO 50 I = 1, 12
      CALL RANDU(Ix, IY, Y)
      IY = IY
      A = A + Y
   50 V = (A - 6.0) * S + AM
   RETURN
END

SUBROUTINE TRIANG(IXT, A, B, C, X)
   THIS ROUTINE WILL CALCULATE RANDUM TRIANGULARLY DISTRIBUTED
   VARIABLES
   IF(C .EQ. A) GO TO 1
   IF(B .EQ. A) AM = 0.
   IF(B .EQ. A) GO TO 2
   AM = (B - A) / (C - A)
2 CONTINUE
   CALL RANDU(IXT, IXY, VAL)
   IXT = IXY
   IF(VAL .LE. AM) XI = SQRT(AM * VAL)
   IF(VAL .GT. AM) XI = 1.0 - SQRT(1.0 - AM - VAL + AM VAL)
   X = A + XI * (C - A)
   RETURN
1 X = A
   RETURN
END

SUBROUTINE SGRAPH
FIRST GENERATE INDIVIDUAL GRAPHS BY NODE

DO 1 J=1,NDI
LM=NOID(J)
LL=NCOUNT(J)
DO 3 I=1,2
LLK=0
DO 2 K=1,ITER
IF(NODEZ(K).NE.LM) GO TO 2
LLK=LLK+1
IF(IZ.EQ.1) ARRZ(LLK)=TIMEZ(K)
IF(IZ.EQ.2) ARRZ(LLK)=COSTZ(K)
2 CONTINUE
IF(LLK.NE.LL) WRITE(6,109) LLK,LL
109 FORMAT(1X,25HVALUES OF LLK AND LL ARE ,215)
IF(LLK.NE.LL) CALL TERM(444)
IF(LLK.EQ.0) GO TO 3
CALL GRAPH(ARRR,LL)
WRITE(6,10)
WRITE(6,10)
WRITE(6,10)
WRITE(6,25) RUNID
25 FORMAT(20X,25A4)
10 FORMAT(1H )
11 FORMAT(20X, 44HGRAPH OF COMPLETION TIMES FOR TERMINAL NODE ,A4)
12 FORMAT(20X, 44HGRAPH OF COMPLETION COSTS FOR TERMINAL NODE ,A4)
3 CONTINUE
1 CONTINUE
CALL GRAPH(TIMEZ,ITER)
WRITE(6,10)
WRITE(6,10)
WRITE(6,10)
WRITE(6,65)
65 FORMAT(20X, 39HGRAPH OF COMPLETION TIMES FOR ALL NODES)
WRITE(6,10)
WRITE(6,25) RUNID
CALL GRAPH(COSTZ,ITER)
WRITE(6,10)
WRITE(6,10)
WRITE(6,66)
66 FORMAT(20X,39HGRAPH OF COMPLETION COSTS FOR ALL NODES)
WRITE(6,10)
WRITE(6,25) RUNID
WRITE(6,111)
111 FORMAT(8H)
DO 200 JJ=1,NODI
NCC= NCT(JJ)
ROGE= FLOAT(NCC)/1000.0/ FLOAT(ITER)
ROG= ROGE- FLOAT(IFIX(ROGE)/1)
IF(ROG.LT.5) GO TO 80
ROGE=ROGE+1.0
80 CONTINUE
200 NBLIP(JJ)=RGE
DO 201 JJ=1,NODI
JZ= NBLIP(JJ)/10
JY= (NBLIP(JJ)-JZ*100)/10
JX= (NBLIP(JJ)-JZ*100-JY*10)
IF(JZ.EQ.0) JZ=10
IF(JY.EQ.0) JY=10
IF(JX.EQ.0) JX=10
KK=NODN(JJ)
LL= NBLIP(JJ)
IF(LL.EQ.0) GO TO 201
LL = LL/10
IF(LL.EQ.0) LL=1
WRITE(6,202) (DOT(K),K=1,LL),TT
WRITE(6,203)NODE(KK), (DOT(K),K=1,LL),TT,DDT,ANN(JZ),ANN(JY),ANN(JX)
1
WRITE(6,204)
WRITE(6,204)
202 FORMAT(11X,1HI,120A1)
203 FORMAT(6X,A4,1X,1HI,120A1)
204 FORMAT(11X,1HI)
201 CONTINUE
WRITE(6,13)
13 FORMAT(12X,10(10H--------1))
WRITE(6,14) (AVAL(I),I=1,10)
14 FORMAT(13X,10(7X,F3.1))
WRITE(6,10)
WRITE(6,10)
WRITE(6,15)
15 FORMAT(20X,27HGRAPH OF NODE PROBABILITIES)
WRITE(6,10)
WRITE(6,111)
RETURN
END

SUBROUTINE GRAPH(Arr,LIM)
DIMENSION ANN(10)
DIMENSION ARR(LIM)
DIMENSION CAT(50)
DIMENSION ICAT(50),NUM(60),DOT(120),AVAL(10)
DATA DOT/120*1HI/
DATA AVAL/1.2,3,4,5,6,7,8,9,1.0/
DATA TERM/1HI/
DATA BLANK/1H /
DATA ANN/1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1H0/
DATA DOT/4H.... /
HIG=0.
SMALL=10000.000
SSUM=0.
DO 1 I=1,LIM
SSUM=SSUM+ARR(I)
IF(ARR(I).GT.BIG) BIG=ARR(I)
IF(ARR(I).LT.SMALL) SMALL=ARR(I)
1 CONTINUE
RANGE=BIG-SMALL
IF(RANGE.EQ.0) GO TO 900
AINT=RANGE/25.
DO 4 K=1,50
4 ICAT(K)=0
DO 2 J=1,LIM
AHIGH=SMALL
DO 3 K=1,25
KK=K
ALOW=AHIGH
AHIGH=ALOW+AINT
IF(K.EQ.25) AHIGH=BIG
IF((ARR(J).LE.ALOW).AND.(ARR(J).LE.AHIGH)) GO TO 77
3 CONTINUE
77 ICAT(KK)=ICAT(KK)+1
2 CONTINUE
SUM=0.
DO 90 I=1,25
90 SUM=SUM+ICAT(I)
SCAT=0.
DO 91 I=1,25
91 CAT(I)=FLOAT(ICAT(I))/SUM
IF(CAT(I).GT.SCAT) SCAT=CAT(I)
IF(SCAT.EQ.CAT(I)) ISCAT=I
91 CONTINUE
AINS=.02
DO 7 I=1,25
7 LM=25-I+1
NUM(LM)=0
AHIGH=0.
DO 8 J=1,50
8 JJ=J
ALOW=AHIGH
AHIGH=ALOW+AINS
IF((CAT(I).GT.ALOW).AND.(CAT(I).LE.AHIGH)) GO TO 20
8 CONTINUE
GO TO 7
20 NUM(LM)=JJ*2-1
7 CONTINUE
VAL=AHIGH
WRITE(6,101)
WRITE(6,822) VAL
822 FORMAT(IX,F9.1,IX,1H1)
DO 10 I=1,125
LM=25-I+1
JZ=CAT(LM)*10.
JY=CAT(LM)*100.-JZ+1
JX=CAT(LM)*1000.-JZ*100.-JY*10.
IF(JX.EQ.0) JX=10
10 CONTINUE
IF(JY.EQ.0) JY=10

IF(JZ.EQ.0) JZ=10

VAL=VAL-AINT

101 FORMAT(1HI)

NUMB=NUMB(I)

IF(NUMB.GT.110) NUMB=110

IF(NUMB.EQ.0) GO TO 15

NUMB=NUMB-1

WRITE(6,12) (DOT(K),K=1,NUMB),TERM

12 FORMAT(1X,10X,1HI,120A1)

WRITE(6,11)VAL,(DOT(K),K=1,NUMB),TERM,BLANK,ODT,ANN(JZ),ANN(JY)

GO TO 10

11 FORMAT(1X,F9.1,1X,1HI,120A1)

GO TO 12

15 WRITE(6,80)

WRITE(6,81) VAL

80 FORMAT(11X,1HI)

81 FORMAT(1X,F9.1,1X,1HI)

CONTINUE

WRITE(6,13)

13 FORMAT(12X,10(1H---------------I))

WRITE(6,14) (AVAL(I),I=1,10)

14 FORMAT(13X,10(7X,F3.1))

AMean=SSum/LIM

SSQ=0.

DO 800 I=1,LIM

SSQ=SSQ+(AMean-ARR(I))**2

SDUM=0.

DO 801 I=1,25

SDUM=SDUM+CAT(I)

IF(SDUM.LT.0.5) GO TO 801

AMED=SMALL+(I-1)*AINT+AINT*(SDUM-CAT(I))/SDUM

GO TO 802

801 CONTINUE

802 CONTINUE

AMODE=SMALL+AINT*(ISCAT-1)+AINT/2.0

AVAR=SSQ/LIM

WRITE(6,810)

810 FORMAT(1H )

WRITE(6,811) AMEAN,AVAR,AMED,AMODE

811 FORMAT(13H THE MEAN IS ,F10.2, 18H THE VARIANCE IS ,F10.2,
C 16H THE MEDIAN IS ,F10.2, 14H THE MODE IS ,F10.2)

RETURN

900 WRITE(6,101)

902 FORMAT(1X,49H ALL VALUES IN THE ARRAY ARE IDENTICAL , AND ARE =,
C ,F12.4)

RETURN

END
DATA KDUM/11*0/
DATA NNODE/500*,NARC/0/
DATA DUM/500*0.0/
DATA IDUM/500*0.0/
DATA DUMA/500*0.0/
DATA BDUM/500*0.0/
DATA JDUM/1500*0.0/
DATA CDUM/2500*0.0/
DATA KJUM/2100*0.0/
DATA ZDUM/1000*0.0/
DATA LDUM/1000*0.0/
DATA YDUM/1000*0.0/
DATA LLDUM/31*0/
DATA ABBA/2000*0.0/
DATA KKAL/1030*0/
DATA RUNID/20*4H /
END

SUBROUTINE DUMP1
COMMON/PARA/NNODE,NARC
COMMON/ARC2/TIME(500),ISTAT(500),PROB(500)
WRITE(6,1)
1 FORMAT(1X, 21HC0MING THROUGH NOGCIF)
WRITE(6,2) (ISTAT(K),K=1,NARC)
2 FORMAT(1015)
RETURN
END

SUBROUTINE DUMP2
COMMON/PARA/NNODE,NARC
COMMON/ARC2/TIME(500),ISTAT(500),PROB(500)
WRITE(6,1)
1 FORMAT(1X, 21HC0MING THROUGH ARCCK)
WRITE(6,2) (ISTAT(K),K=1,NARC)
2 FORMAT(1015)
RETURN
END

SUBROUTINE UNIFRM(IXT,A,B,C,X)
C
C THIS SUBROUTINE WILL CALCULATE UNIFORM DISTRIBUTED
VARIABLES
PA ADDITION TO PROGRAM AT 8 DEC 70
CODE NUMBER IS 3
C
IF( C .EQ. A ) GO TO 1
IF( B .NE. 0.0 ) CALL TERM( 1001)
CALL RANDU(I,IXY,VAL)
IXT = IXY
X = A + VAL * (C - A )
RETURN
1 X = A
RETURN
END
APPENDIX II

GENERAL FLOW CHART AND LISTING OF RISCA
This appendix includes a general processing flow chart for RISCA, but it does not include a description or flow chart of the individual subroutines. A detailed description and flow chart of the individual subroutines are provided in Appendix III. In addition to the flow chart, a complete program listing is provided. This version of RISCA is designed to run in a batch mode only, although the interested user could certainly adapt this program for a time sharing mode.

RISCA is written in FORTRAN IV. Even though FORTRAN IV is considered to be a standard language, adapting the program for any computer will probably require minor program modifications. These modifications generally result from peculiarities of the given system.

Unlike MATHNET, the RANDU subroutine in RISCA contains its own uniform random number generating routine. Therefore, no modification in this area is required.
Flow Chart of "RISCA"
COMMON/RAiC/ IXX

PROGRAM RISCA

THIS IS THE MAIN ROUTINE

IXXX=0 CALL RANDU(Ixx,IXYY,RVAL)
IXX=IXYY CALL REPID CALL ARCIN CALL NODIN CALL SCAF CALL RUNSYS CALL EXIT STOP END

SUBROUTINE RANDU(I,J,RAN)
IF(I.EQ.0) I=1111111

SUBROUTINE REPID
READS IN NAME OF NET AND SETS ITERATIONS

COMMON/ITERA/ITER
COMMON/1DO/ RUNID(20)
READ(5,2) RUNID
2 FORMAT(25A4)
ITER=500
RETURN END

SUBROUTINE ARCIN
THIS ROUTINE WILL SERVE TO READ IN ARCS AND ASSOCIATED DATA
COMMON/ARC1/ARC(100),INODE(100),ONODE(100),ITIMET(100),TARG1(100),TARG2(100),TARG3(100),COSTC(100),COSTV(100)
COMMON/PAPA/NNODE,NARC
COMMON /ARC2/ TIME(100),ISTAT(100),PROB(100)
COMMON/NODI/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),IIARC(100),OARC1(100),ITYPE(100),OTYPE(100),MNIND(100),TIME(100)
2TIMEN(100)
REAL NODE
INTEGER IARC,OARC
INTEGER INODE
DATA RETC/4*RETCU/ I 50 READ(5,1) ANAME,AINODE,ADNODE,IDIST,P1,P2,P3,C1,C2,D 1 FORMAT(3A4,T1,3F10.0,3F10.0)
IF(ANAME.EQ.RETC) RETURN NARC=NARC+1 ARC(NARC)=ANAME ITIMET(NARC)=IDIST TARG1(NARC)=P1 TARG2(NARC)=P2 TARG3(NARC)=P3
SUBROUTINE NODIN

C THIS ROUTINE WILL READ IN NODE NAMES AND DATA

COMMON/NODI/ NODE(100), IARC(100,10), OARC(100,10), PPOA(100,10),
 IIARC(100), OARC(100), IYPE(100), OTYPE(100), MNIND(100),
2TIME(100)

REAL NODE

DIMENSION DDUH(10)

COMMON/PARA/ INODE, NARC

COMMON/TERRN/ NODN(30), NODI, TIMEZ(500), COSTZ(500), NODEZ(500),
 INCOUNT(30)

COMMON /ARC2/ TIME(100), ISTAT(100), PROB(100)

COMMON/ARC1/ ARC(100), INODE(100), ONODE(100), TIMET(100), TARG1(100),
 1TARG2(100), TARG3(100), COSTC(100), COSTV(100)

DIMENSION ZNAM(10), ZPROB(10), IZNAM(10), Y1(10), Y0(10)

INTEGER OARC, OARCI

COMMON/TERNI/ INODI(10), INODT

INTEGER OTYPE

DATA CZZ/4HZZZZ/

DATA RETC/4HRETC/
C 20 READ(5,40) ADUM,1:NM,1:NDUM
  40 FORMAT(A4,11,11)
   IF (ADUM.EQ.'+',RETC) RETURN
   DO 100 I=1,NNODE
   NDUM=1
   IF (NODE(1),.EQ.,ADUM) GO TO 101
  100 CONTINUE
   NNODE=NNODE+1
   NDUM=NNODE
  101 ITYPE(NDUM)=1:IDUM
   OTYPE(NDUM)=1:IDUM
   IF (ITYPE(NDUM),.EQ.,4) GO TO 90
   C ITYPE=4 MEANS INITIAL NODE
   IF (ITYPE(NDUM),.EQ.,6) GO TO 400
   C ITYPE=6 MEANS WE HAVE A 1/1 NODE WITH A NEGATIVE INPUT
   IF (ITYPE(NDUM),.EQ.,5) GO TO 60
   C ITYPE=5 MEANS A 1/1 NODE WHICH MEANS WE MUST READ IN ORDERINGS
   30 IF (ITYPE(NDUM),.EQ.,2) GO TO 70
   C ITYPE=2 MEANS PROBABILISTIC FIRINGS...WE MUST READ IN PROBABILITIES
   IF (ITYPE(NDUM),.EQ.,4) GO TO 80
   IF (ITYPE(NDUM),.EQ.,7) GO TO 400
   GO TO 20
  60 CONTINUE
   C INSERT READS HERE FOR 1/1 NODES
   READ(5,303) HM,(YI(I),YO(I),I=1,MM)
  303 FORMAT(I2,10(A4,A4))
  203 FORMAT(I2,10(A4,F4.3))
   DO 305 L=1,NARC
   DO 306 K=1,MM
   IF (ARC(L),.EQ.,YI(K)) IAARC(NDUM,K)=L
   IF (ARC(L),.EQ.,YO(K)) OAARC(NDUM,K)=L
  306 CONTINUE
  305 CONTINUE
   GO TO 20
  70 CONTINUE
  70 C INSERT READ FOR PROBABILITIES
  220 READ(5,203) NN,(ZHAM(I),ZPROB(I),I=1,NN)
   IF (NN,NE.,OARC(NDUM)) CALL TERM(1)
   DO 205 I=1,NN
   DO 206 J=1,NN
   JJ=J
   LM=OARC(NDUM,1)
   IF (ZHAM(J),.EQ.,ARC(LM)) GO TO 207
  206 CONTINUE
   CALL TERM(2)
  207 ZPOA(NDUM,1)=ZPROB(JJ)
  205 CONTINUE
   GO TO 20
  80 INODI=INODI+1
   NDUM(INODI)=NDUM
   GO TO 20
  90 INODT=INODT+1
   INODT(INODT)=NDUM
   GO TO 30
 400 CONTINUE
 420 READ(5,303) HM,(YI(I),YO(I),I=1,MM)
   DD 405 L=1,NARC
C

COMMON/ITERS/ITER
   COMMON/ITD/RUNIND(20)
   COMMON/ARC/IXX
   COMMON/ARCF/ARC(100),INODF(100),OINU(100),ITIME(100),TARG1(100),
   TARG2(100),TARG3(100),COSTC(100),COSTV(100)
   COMMON/PAPA/INODE,NARC
   COMMON/J/INODE(100),ARC(100,10),OARC(100,10),PPUA(100,10),
   I1ARC(10),OARC(100),ITYPE(100),OTYPE(100),MNIN(100),
2TIME(100),
   COMMON/TERM/INODE(30),NODE,(TIMEZ(500),COSTZ(500),NODEZ(500),
1TIMEZ(30)
   REAL NODE,
   INTEGER IODE,CTYPE,ARCI
   DIMENSION TYPE(4),INRUL(7),OUTRUL(7),RLI(2)
   DATA TYPE/'NORM', 'TRI ', 'UNIF', 'CON ' /
   DATA INRUL/'AND ', 'OR ' , 'INIF ', '1-1 ', '1-B ', 'PREF '/
   DATA OUTRUL/'ALL ', 'PROP ', 'TERM ', '1-1 ', '1-B ', 'PREF '/
   DATA RL1/ 'NO ', 'YES '/

   WRITE(6,4) RUNID
4 FORMAT(1H1,25X,25A4)
   WRITE(6,7) ITER
7 FORMAT(50X,15,2X,'ITERATIONS')
   WRITE(6,50)
   WRITE(6,1)
1 FORMAT(110,1X,'ARC INP NODE OUT NODE TIME DIST ARG1',
1 ARC2 ARG3 COST P OF ',
2 'COM')
   WRITE(6,50)
   DO 2 1=1,NARC
   LM=INODE(1)
   LT=NODE(LM)
   WRITE(6,3) ARC(1),NODE(LM),NODE(LN),TYPE(1),TARG1(LT)
3 FORMAT(2X,'TARG2(LT),TARG3(LT),COSTC(LT),COSTV(LT),PROB(LT)
   12 FORMAT(2X,4X,5X,8X,4X,9X,4X,3f3X,10.2,4X,10.2,3H + ,F10.2,
   2 CONTINUE

C

COMMON WRITE OUT ARCS AND ASSOCIATED DATA
   WRITE(6,50)
   WRITE(6,50)
   50 FORMAT(1H )
   WRITE(6,51)
   51 FORMAT(1X,' NODE NO. OF INPUT ARCS NO. OF OUTPUT ARCS',MAT07530

C THIS ROUTINE CONTROLS THE RUNNING OF A NET

C COMMON/ITR7/IREPIT
COMMON/KIND/KIND
COMMON/ARC1/APC1(100),INODE(100),NODE(100),TIMET(100),TARG1(100),
TARG2(100),TARG3(100),COSTC(100),COSTV(100)
COMMON/ITRHA/ITER
COMMON/AFCC2/ TIME(100),ISTAT(100),PROB(100)
COMMON/RUIHER/ ITERMS
COMMON/KINT/SHTIM, INSH
COMMON/TERNN/ NODE(30), NODE, TIMEZ(500), COSTZ(500), NODEZ(500),
INCOUNT(30)
COMMON/NUMD/ NODE(100), IARC(100, 10), OARC(100, 10), PPOSEA(100, 10),
IARC1(100), OARC1(100), ITYPE(100), OTYPE(100), MIND(100),
2TIMFN(100)
COMMON/PAPA/INNODE, NARC
COMMON/TERNI/ NUMD(10), INGDT
INTEGER IARC, OARC, ITYPE

C SET NUMBER OF ITERATIONS SO FAR TO 0
IREPIT=0

20 SHTIM=999999.0

C ABOVE Initializes SMALLEST TERMINAL FINISH TIME TO A LARGE NUMBER
C THE FOLLOWING CODE SETS THE INDICATOR OF WHETHER OR NOT
C A TERMINAL NODE HAS BEEN FILLED TO INDICATE NO,

C ITERMS=0

C NOW FIRE INITIAL NODES
DO 10 I=1,INODT
LM=INODI(1)
TIMEN(LM)=0.
IF(OTYPE(LM), EQ, 1) CALL ALLFINR(LM)
IF(OTYPE(LM), EQ, 2) CALL PROFIR(LM)
CONTINUE
1 CALL ARCCCH
CALL NODCCH
IF(KIND, EQ, 0) GO TO 60
IF(ITERMS, EQ, 0) GO TO 1
CALL ENDIT(KEY)
IF(KEY, EQ, 0) GO TO 1

61 IREPIT=IREPIT+1
IF(IREPIT, EQ, ITER) CALL SGRAPH
IF(IREPIT, EQ, ITER) RETURN
DO 40 I=1,NARC
TIME(1)=0.
ISTAT(1)=0.
CONTINUE
TO 50 I=1,NNODES
TIME(IN)=0.
CONTINUE
TO 20
C CONFIRM IF NO NODES HAVE FIRED
IF (ITEMS.EQ.0) CALL TERM(*)
GO TO 61
C CONFIRM IF NO TERMINAL NODES HAVE BEEN FILLED
GO TO 61
C CONFIRM IF THERE IS A TERMINAL NODE- END ITERATION
END
SUBROUTINE ALLFIRE(I)
C FIRE ALL OUTPUT ARCS
COMMON /ARC2/ TIME(100),ISTAT(100),PHOR(100)
COMMON/NODE/ NCDE(100),IARC(100,10),OARC(100,10),PPOA(100,10),
           IARC(100),OARC(100),ITYPE(100),OTYPE(100),MNIND(100),
2TIME(100)
REAL NODE
INTEGER OARC,OARC,I
DO 1 K=1,11
K=OARC(I,K)
ISTAT(KK)=1
CONTINUE
RETURN
END
SUBROUTINE PROFIRE(I)
C FIRE NODE I USING STOCHASTIC CONSIDERATIONS
COMMON /ARC2/ TIME(100),ISTAT(100),PHOR(100)
COMMON/NODE/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),
           IARC(100),OARC(100),ITYPE(100),OTYPE(100),MNIND(100),
2TIME(100)
REAL NODE
COMMON/RAUC/ IXX
INTEGER OARC,OARC,OARC,ITYPE
IXX=IXX
CALL RAND(iXX,iXY,RVAL)
IXX =IXY
AHIGH=0.
II=OARC(I)
DO 1 K=1,11
K=OARC(I,K)
ALOW=AHIGH
AHIGH=ALOW+PPOA(I,K)
IF (RVAL.GE.ALOW).AND.(RVAL.LE.AHIGH) GO TO 2
CONTINUE
1 CALL TERM(3)
2 ISTAT(KK)=1
RETURN
END
SUBROUTINE ARCCHK
C IF INITIATEP CHECK PROBABILITY OF COMPLETION
C IF COMPLETE CALCULATE TIME AND COST
C
COMMON/RANC/ IXX
COMMON/ARC2/ TIME(100),I STAT(100),PROR(100)
COMMON/ARC1/APC(100),INODE(100),ONODE(100),ITIMET(100),TARGI(100),
ITARG2(100),TARG3(100),COSTC(100),COSTV(100)
COMMON/ PARA/ INODE, KARC
COMMON/I NODE/ NODE(100),I ARC(100,10),O ARC(100,10),PO U (100,10),
IIARC(100),OARC(100),I TYPE(100),O TYPE(100),MN IN(100),
2 TIMEN(100)

REAL N0DE

DO 1 I=1, NARC
IXXX=IXX
A=TARGI(1)
B=TARG2(1)
C=TARG3(1)
IF(ITIMET(I),EQ.1) CALL GAUSS(IXXX,B,A,TVAL)
IF(ITIMET(I),EQ.2) CALL TRIANG(IXXX,A,B,C,TVAL)
IF(ITIMET(I),EQ.3) CALL UNIF(IXXX,A,B,TVAL)
IF(ITIMET(I),EQ.4) TVAL=A
IXX=IXXX
LH=INODE(I)
TIME(I)=TVAL+TIMEN(LM)
IF( (I STAT(I),NE.1) ) GO TO 1
IF( ( PRO(1),EQ.1,) ) GO TO 2
IXXX=IXX
CALL RANDU(IXXX,IXXY,RVAL)
IXX=IXXY
IF( (RVAL.LE.PROP(1)) ) GO TO 2
I STAT(I)=3
GO TO 1
2 I STAT(I)=2
1 CONTINUE
RETURN
END

SUBROUTINE GAUSS(IX,S,AM,V)
C
CALCULATES NORMAL DISTRIBUTED VARIABLES
A=0.0
DO 50 I=1,12
CALL RANDU(I X,IY,Y)
IX =IY
50 A = A +Y
V=(A-6.0)*S+AM
RETURN
END

SUBROUTINE TRI ANG(I XT,A, B,C,X)
C
THIS ROUTINE WILL CALCULATE RANDOM TRIANGULARLY DISTRIBUTED
C
VARIABLES
C
IF((C-A),EQ.0.) CALL TERM(9)
AM=(R-A)/(C-A)
CALL RANDU(I X T,I XY,VAL)
IXT=IXY
IF( (VAL.LE. AM) ) XI=SQRT(AM*VAL)
IF( ( VAL > GT . AM) ) XI=1.- SQRT(1.-AM+VAL+AM*VAL)
X=A+XI*(C-A)
RETURN
END
SUBROUTINE UNIF(IX, IY, Y)
CALCULATES UNIFORMLY DISTRIBUTED VARIABLES

CALL RANDU(IY, Y)
I = IY
VAL = OPT * Y * (PFS - OPT)
RETURN
END

SUBROUTINE NONCHK
SEE WHAT NODES ARE READY TO FIRE, FIRE THOSE THAT ARE READY

COMMON /PARA/ NODE, NARC
COMMON /NOD1/ NODE(100), IARC(100, 10), OARC(100, 10), PPOA(100, 10),
IARC(100), OARC(100), IYPE(100), OTYPE(100), MIND(100),
2TIMES(100)
COMMON /APC2/ TIME(100), ISTAT(100), PROB(100)
INTEGER IYPE, OARC1, OARC
COMMON /KIND/KIND
REAL NODE

KIND = 0
DO 1 I = 1, IY, 1
J = 1
IF (IYPE(I).EQ.1) CALL ANPTST(IL, J)
IF (IYPE(I).EQ.2) CALL ORTST(IL, J)
IF (IYPE(I).EQ.5) GO TO 30
IF (IYPE(I).EQ.6) GO TO 31
IF (IYPE(I).EQ.7) GO TO 22
IF (J.EQ.0) GO TO 32
KIND = KIND + 1
1 CONTINUE
RETURN
END

SUBROUTINE ANPTST(I, J)
COMMON /NOD2/ NODE(100), IARC(100, 10), OARC(100, 10), PPOA(100, 10),
IARC(100), OARC(100), IYPE(100), OTYPE(100), MIND(100),
2TIMES(100)

22 CONTINUE
IF (OCTYPE(I).EQ.1) CALL ALLFIR(I)
IF (OCTYPE(I).EQ.2) CALL PROFIR(I)
IF (OCTYPE(I).EQ.4) CALL ITALL(I)
30 IF (OCTYPE(I).EQ.5) CALL ONECONF(I, IL, J)
31 IF (OCTYPE(I).EQ.6) CALL ONEBAR(I, IL, J)
32 IF (OCTYPE(I).EQ.7) CALL PREFEK(IL, J)
IF (OCTYPE(I).EQ.5) KIND = KIND + 1
IF (OCTYPE(I).EQ.6) KIND = KIND + 1
IF (OCTYPE(I).EQ.7) KIND = KIND + 1
1 CONTINUE
RETURN
END

SUBROUTINE ORTST(I, J)
COMMON /NODT/ NODE(100), IARC(100, 10), OARC(100, 10), PPOA(100, 10),
IARC(100), OARC(100), IYPE(100), OTYPE(100), MIND(100),
2TIMES(100)

CONTINUE
REAL NODE
COMMON /ARC2/ TIME(100), ISTAT(100), PROB(100)

II=IARCI(I)
TIM=0.
GO TO 1,II
KK=IARCI(I,K)
IF (ISTAT(KK).NE.2) GO TO 2
IF (TIME(KK).GT.TIM) TIM=TIME(KK)
1 CONTINUE
TIME(I)=TIM
J=1
RETURN
J=0
RETURN
END

SUBROUTINE ORTST(I,J)

COMMON /ARCI/ ARC(100),INODE(100),ITYPE(100), OARC(100), OTYPE(100), MINT(100), OARC(100), OARC1(100), ITYPER(100), OARC(100), MINT(100), ITARG1(100), ITARG2(100), TARG3(100), COSTC(100), COSTV(100)

II=IARCI(I)
TIM=1000000.
J=0
DO 1 K=1,II
KK=IARCI(I,K)
IF (ISTAT(KK).NE.2) GO TO 1
1 CONTINUE
IF(J.EQ.0) RETURN
TIME(I)=TIM
RETURN
END

SUBROUTINE ONEONE(ILK,J)

COMMON /NODE, NARC
COMMON /NODE/ NODE(100), IARC(100,10), OARC(100,10), PPOA(100,10), IARCI(100), OARC(100), ITYPE(100), OTYPE(100), MINT(100)
2TIME(100)
COMMON /ARC/ ARC(100), NODE(100), MINT(100), ITARG1(100), ITARG2(100), TARG3(100), COSTC(100), COSTV(100)

INTEGR OTYPE, OARC
REAL NODE

J=0
II=IARCI(ILK)
TIM=1000000.
DO 1 K=1,11
  KK=IARC(ILK,K)
  IF(ISTAT(KK),NE.2) GO TO 1
  IF(TIME(KK),GE,TIM) GO TO 1
  TIME=TIME(KK)
  J=K
  CONTINUE
  IF(J.EQ.0) RETURN
  C ZERO OUT ALL INPUTS, FIRE JTH OUTPUT ARC
  DO 2 K=1,11
    LM=IARC(ILK,K)
    IF(ISTAT(LM),NE.0) GO TO 2
    ISTAT(LM)=4
  CONTINUE
  LH=OARC(ILK,J)
  ISTAT(JM)=1
  TIM=TIME(JM)
  RETURN
END

SUBROUTINE OPEAR(ILK,J)

TESTS ONE-ONE BAR NOVES

COMMON/ PARA/NODE,NARC
COMMON/ HDMI NODE (100), IARC (100,10), OARC (100,10), PP0A (100,10),
1 IARC (100), OARC (100), IYPE (100), OYPE (100), MNIND (100),
2 TIME (100)
COMMON/ ARC1/ARC (100), THODE (100), NODE (100), ITIMET (100), TARG1 (100),
1 TARG2 (100), TARG3 (100), COST (100), COSTV (100)
COMMON /ARC2/ TIME (100), ISTAT (100), PKO (100)
INTEGER OYPE, OARC, IARC
REAL NCD

J=0
1=IARC(ILK)
TIM=1060000.
DO 1 K=1,11
  KK=IARC(ILK,K)
  IF(KK,EQ.100) IEAK=K
  IF(KK,EQ.100) GO TO 1
  IF(ISTAT(KK),NE.2) GO TO 1
  IF(TIME(KK),GE,TIM) GO TO 1
  TIME=TIME(KK)
  J=K
  CONTINUE
  IF(J.EQ.0) GO TO 5
  C ZERO OUT ALL INPUTS, FIRE JTH OUTPUT ARC
  DO 2 K=1,11
    LM=IARC(ILK,K)
    IF(ISTAT(LM),NE.0) GO TO 2
    ISTAT(LM)=4
  CONTINUE
  LH=OARC(ILK,J)
  ISTAT(JM)=1
  TIMEN(ILK)=TIM
  RETURN
5 TIM=0.
DO 6 K=1,11
  KK=IARC(ILK,K)
IF(KK, EQ, 100) GO TO 6
IF(ISTAT(KK), NE, 3) GO TO 7
IF(TIME(KK), LE, TIME) GO TO 6
TIME = TIME(KK)
CONTINUE
LM = OARC(ILK, IARF)
ISTAT(LM) = 1
J = 1
DO 12 K = 1, II
LM = IARC(ILK, K)
ISTAT(LM) = 4
CONTINUE
TIME = TIME(ILK) = TIME
CONTINUE
RETURN
END

SUBROUTINE PREFER(ILK, J)

COMMON /PAPA/ NNODE, NARC
COMMON /NODE/ NODE(100), IARC(100, 10), OARC(100, 10), PP(OA(100, 10),
1, IARCI(100), DARC(100, 10), OTYPE(100), OTYPE(100), MNINP(100),
2, TIME(100)
COMMON /ARC1/ ARC1(100), INODE(100), ONODE(100), ITIME(100), TARG1(100),
1, TARG2(100), TARG3(100), COSTC(100), COSTV(100)
COMMON /ARC2/ TIME(100), ISTATE(100), PROB(100)
INTEGER OTYPE, OARCI, OARC
REAL NODE
J = 0
II = IARCI(ILK)
DO 1 K = 1, II
KK = IARC(ILK, K)
IF(KK, EQ, 100) GO TO 1
IF(ISTAT(KK), EQ, 0) GO TO 30
IF(ISTAT(KK), EQ, 1) GO TO 30
IF(ISTAT(KK), EQ, 4) GO TO 30
CONTINUE
J = 1
C IF WE GET HERE THE NODE WLL BE FIRED
C FIRE FIRST ARC PAIR WITH 2 STATUS, IF THERE IS ONE
TIME = 0
III = II - 1
DO 2 K = 1, III
KK = IARC(ILK, K)
KKK = K
IF(TIME(KK), GT, TIME) TIME = TIME(KK)
IF(ISTAT(KK), EQ, 2) GO TO 4
CONTINUE
C IF WE GET HERE FIRE BAR ARC
LM = OARC(ILK, II)
GO TO 5
C FIRE THE KKK ARC
LM = OARC(ILK, KKK)
ISTAT(LM) = 1
DO 40 K = 1, II
LM = IARC(ILK, K)
ISTAT(LM) = 4
CONTINUE
COMMON/MINT/SMTIM, INSII
COMMON/RULEP/ TERMS
COMMON/NOD1/NODE(100), IARC(100, 10), OARC(100, 10), PPOA(100, 10),
IARC1(100), OARC1(100), ITYPE(100), OTYPE(100), MINN1(100),
2TIMEN(100)

TERMS=1

ABOVE INDICATES A TERMINAL NODE HAS BEEN FILLED
IF(TIMEN(11) .GE. SMTIM) RETURN
SMTIM = TIMEN(11)
INS1 = 11
RETURN
END

SUBROUTINE END1(K)

THIS ROUTINE CHECKS TO SEE IF THERE ARE COMPLETED ARCS WITH TIMES
SMALLER THAN THE SMALLEST TERMINAL NODE
IF SO SET KEY=0, IF NOT SET KEY=1 AND TERMINATE THIS RUN
COMMON /ARC2/ TIME(100), ISTAT(100), PROB(100)
COMMON/MINT/ SMTIM, INSII
COMMON/PARA/NODE, NARC
DO 1 I=1, IARC
IF(ISTAT(I), EQ, 0) GO TO 1
IF(ISTAT(I), EQ, 3) GO TO 1
IF(ISTAT(I), EQ, 4) GO TO 1
IF(TIME(I), LT, SMT1M) GO TO 2
1 CONTINUE
KEY = 1
RETURN
2 KEY = 0
RETURN
END

SUBROUTINE PTERM
DETERMINES COST OF ITERATION
COMMON/PARA/NODE, NARC
COMMON/IFRZ/IREPIT
COMMON/MINT/ SMTIM, INSII
COMMON/NOD1/NODE(100), IARC(100, 10), OARC(100, 10), PPOA(100, 10),
IARC1(100), OARC1(100), ITYPE(100), OTYPE(100), MINN1(100),
2TIMEN(100)
COMMON/TERMII/ NODN(30), NOD1, TIMEZ(500), COSTZ(500), NODEZ(500),
INCOUT(30)
COMMON/ARCI/ARC1(100), INODE(100), ONODE(100), ITIMET(100), TARG1(100),
1TARG2(100), TARG3(100), COSTC(100), COSTV(100)
COMMON /ARC2/ TIME(100), ISTAT(100), PROB(100)
COMMON /IDN/ RUNID(20)
COMMON/TERNI/ INOD1(10), INODT

DO 1 I=1, NOD1
JJ=1
IF(NODI(J).EQ.1).THEN GO TO 2
1 CONTINUE
CALL TERM(69)
2 NCOUNT(JJ)=NCOUNT(JJ)+1
LM=NCOUNT(JJ)
IH=IREPIT+1
TIMEZ(IH)=SHTIM
NODEZ(IH)=INSH
COSTZ(IH)=0.
DO 1 J=1,NARC
LM=NODE(J)
IF(ISTAT(J).EQ.0) GO TO 3
IF(TIME(J).GT.SHTIM) GO TO 4
COSTZ(IH)=COSTZ(IH)+COSTC(I)+COSTV(I)*(TIME(J)-TIMEN(LM))
GO TO 3
4 IF(TIME(LM).GT.SHTIM) GO TO 3
COSTZ(IH)=COSTZ(IH)+COSTC(I)+COSTV(I)*(SHTIM-TIMEN(LM))
3 CONTINUE
RETURN
END

SUBROUTINE SGRAPH
DIMENSION DOT(120)
COMMON/TERNN/ NODN(30), NODI, TIMEZ(500), COSTZ(500), NODEZ(500),
INCOUNT(30)
REAL NODE
COMMON/NODE/ NODN(100), IRAK(100,10), OARC(100,10), PPOA(100,10),
1 IARCI(100), OARC(100,10), ITYPE(100), OTYPE(100), MNINH(100),
2 TIMEN(100)
COMMON/ITEKA/ ITER
DIMENSION ARRRA(500)
DIMENSION ANN(10)
DIMENSION NBLIP(10)
DIMENSION AVAL(10)
COMMON/IID/ RUNIT(20)
DATA TT/1H1/
DATA D0T/120*1H/
DATA ANN/4H1111,4H2222,4H3333,4H4444,4H5555,4H6666,4H7777,4H8888,
14H9999,4H0000/
DATA D0T/4H..../
DATA AVAL/1,2,3,4,5,6,7,8,9,10/;
C FIRST GENERATE INDIVUAL GRAPHS BY NDF
DO 1 J=1,NODI
LM=NODN(J)
LL=NCOUNT(J)
DO 2 I=1,2
LLK=0
DO 2 K=1,ITER
IF(NDFZ(K).GE.LM) GO TO 2
LLK=LLK+1
IF(IZ.EQ.1) ARRRA(LLK)=TIMEZ(K)
IF(IZ.EQ.2) ARRRA(LLK)=COSTZ(K)
2 CONTINUE
IF(LLK.GE.LL) WRITE(6,109) LLK,LL
109 FORMAT(IX,'VALUES OF LLK AND LL ARE ',2I5)
IF(LLK.GE.LL) CALL TERM(6)
CALL GRAPHR(ARRRA,LL)
WRITE(6,60)
IF (IZ.FG.,1) WRITE(6,61) NODE(LM)
IF (IZ.FG.,2) WRITE(6,62) NODE(LM)
WRITE(6,60)
WRITE(6,25) RUNID

60 FORMAT(1H1)
61 FORMAT(20X,'GRAPH OF COMPLETION TIMES FOR TERMINAL NODE ',A4)
62 FORMAT(20X,'GRAPH OF COMPLETION COSTS FOR TERMINAL. NODE ',A4)
3 CONTINUE

1 CONTINUE
CALL GRAPH(TIMEZ,ITER)
WRITE(6,10)

10 FORMAT(1H1)
WRITE(6,10)
WRITE(6,65)

65 FORMAT(20X,'GRAPH OF COMPLETION TIMES FOR ALL NODES')
WRITE(6,10)
WRITE(6,25) RUNID
CALL GRAPH(COSTZ,ITER)
WRITE(6,10)
WRITE(6,10)
WRITE(6,65)

66 FORMAT(20X,'GRAPH OF COMPLETION COSTS FOR ALL NODES')
WRITE(6,10)
WRITE(6,25) RUNID
25 FORMAT(20X,25A4)
WRITE(6,111)

111 FORMAT(1H1)
DO 200 J,J,J,N0N1
NCC=NCOUNT(JJ)

200 NBLIP(JJ)=FLOAT(NCC)*100.*FLOAT(ITER)
DO 201 J,J,J,J,N0N1
JZ=NBLIP(JJ)/10
JY=NBLIP(JJ)-JZ*10
IF(JZ.FG.,0) JZ=10
IF(JY.FG.,0) JY=10
KK=NODN(JJ)
LL=NBLIP(JJ)
IF(LL.FG.,0) GO TO 201
WRITE(6,202) (DOT(K),K=1,LL),TT
WRITE(6,203) NODE(KK), DOT(K),K=1,LL),TT,DOT,ANN(JZ),ANN(JY)
WRITE(6,204)
WRITE(6,204)

202 FORMAT(11X,1HI,120A1)
203 FORMAT(6X,A4,1X,1HI,120A1)
204 FORMAT(11X,1HI)
201 CONTINUE
WRITE(6,13)

13 FORMAT(12X,10(10H--------1))
WRITE(6,14) (AVAL(I),I=1,10)
14 FORMAT(13X,10(7X,F3.1))
WRITE(6,10)
WRITE(6,10)
WRITE(6,15)

15 FORMAT(20X,'GRAPH OF NODE PROBABILITIES')
WRITE(6,10)
WRITE(6,25) RUNID
WRITE(6,111)
SUBROUTINE GRAPH(ARR,LIM)

DIMENSION ANN(10)
DIMENSION ARR(LIM)
COMMON/NODE/ NODE(100),IARC(100,10),OARC(100,10),PPOA(100,10),
IARC(100),OARC(100),ITYPE(100),OTYPE(100),MNIND(100),
COMMON/INDU/RUNID(20)

DIMENSION CAT(50)
DIMENSION ICAT(50),NUM(60),DOT(120),AV1(10)
DATA INT/4H00000/ DATA ANH/4H1111,4H2222,4H3333,4H4444,4H5555,4H6666,4H7777,4H8888,
14H9999,4H00000/ DATA BLANK/1H/ DATA TERM/1H/
DATA AVAL/.1,.2,.3,.4,.5,.6,.7,.8,.9,1.10/ DATA DOT/120*1H=/
BIG=0.
SMALL=10000000.
AVE=0.0
SD=0.0

IF (LIM.EQ.0) GO TO 900
DO 1 I=1,LIM
AVE = AVE+ARR(I)
1 CONTINUE

IF (ARR(I).GT.BIG) BIG=ARR(I)
IF (ARR(I).LT.SMALL) SMALL=ARR(I)
CUTINUE

RANGE=BIG-SMALL
IF (RANGE.EQ.0.) GO TO 900
AVE = AVE/FLOAT(LIM)
AINT=RANGE/25.
DO 4 K=1,50
ICAT(K)=0
DO 2 J=1,LIM
SD=SD+((AVE-ARR(J))*2)
2 CONTINUE

AHIGH=SMALL
ALOW=AHIGH
AHIGH=ALOW+AINT
IF ((ARR(J).GE.ALOW).AND.(ARR(J).LE.AHIGH)) ICAT(K)=ICAT(K)+1
3 CONTINUE

2 CONTINUE
IF (LIM.GT.1) VAP=SD/(FLOAT(LIM)-1.0)
IF (LIM.LE.1) VAP=0
SD=SQRT(VAP)
SUM=0.

DO 90 I=1,26
SUM=SUM+FLOAT(ICAT(I))
INDEX=1
90 CONTINUE

DO 92 I=1,26
CAT(I)=FLOAT(ICAT(I))/SUM
92 CONTINUE
GO TO 93

INDEX=2
DO 94 I=2,26
CAT(I)=CAT(I)+CAT(I-1)
94 CONTINUE

DO 7 I=1,26
LM=26-I+1
NUM(LM)=0
AHIGH=0
GO TO 2U
J=1,51
J3=3
ALQW=ALQW+1
AHIGH=ALQW+AINS
IF((CAT(I),GT,ALQW),AND,(CAT(I),LE,AHIGH)) GO TO 20
8 CONTINUE
GO TO 7
20 NUM(LM)=J,J+2-1
7 CONTINUE
VAL=HIGH+AINIT
WRITE(6,101)
DO 10 I=1,26
LM=26-I+1
JZ=CAT(I,J)*10,
JY=CAT(I,J)*100.-FLOAT(J7)*10.,
JX=CAT(I,J)*1000.-FLOAT(J7)*100.-FLOAT(J7)*100.,
IF(JY,.EQ.0) JY=10
IF(JZ,.EQ.0) JZ=10
IF(JX,.EQ.0) JX=10
VAL=VAL-AINIT
101 FORMAT(1H1)
NUM=NUM(I)
IF(NUM,.GT.110) NUM=110
IF(NUM,.EQ.0) GO TO 15
NUM=NUM-1
WRITE(6,12) (DOT(K),K=1,NUM),TERM
12 FORMAT(1X,12X,1HI,120A1)
WRITE(6,11) VAL,(DOT(K),K=1,NUM),TERM, BLANK, DOT, ANN(JZ), ANN(JY)
1,ANN(JX)
11 FORMAT(1X,F11.3,1X,1HI,120A1)
GO TO 10
15 WRITE(6,80)
WRITE(6,81) VAL
80 FORMAT(13X,1HI)
81 FORMAT(1X,F11.3,1X,1HI)
GO CONTINUE
WRITE(6,13)
13 FORMAT(12X,10(10H---------))
WRITE(6,14) (AVAL(I),I=1,10)
14 FORMAT(13X,10(7X,F3.1))
WRITE(6,70) AVE, VAR, SD
70 FORMAT(16X,'MEAN =',F11.3,'VARIANCE =',F11.3,'STANDARD DEVIATION =',F11.3)
IF(INDEX,.EQ.1) GO TO 94
RETURN
900 WRITE(6,101)
901 IF(LIM,N,GT.0) WRITE(6,900) BIG
902 FORMAT(1X,'ALL VALUES IN THE ARRAY ARE IDENTICAL , AND ARE = ',I1,F12.4)
903 FORMAT(1X,'THIS NODE WAS NEVER A TERMINAL NODE')
RETURN
END
SUBROUTINE TERM(I)
THIS SUBROUTINE PRINTS ERROR MESSAGES.

COMMON/ARC1/ARC(100),INODE(100),ONODE(100),ITIMET(100),TARG1(100),
ITARG2(100),TARG3(100),COSTC(100),COSTV(100)
COMMON/ACOC/TIME(100),ISTAT(100),PROR(100)
COMMON/Para/NNODE,NARC
GO TO (1,2,3,4,5,6,24,25,27),I
1 WRITE(6,7)
7 FORMAT(1X,'***ERROR***PROBABILISTIC OUTPUT NODE - WRONG NUMBER OF
1 ARCS STATED***NODIN')
   CALL EXIT
2 WRITE(6,8)
8 FORMAT(1X,'***ERROR***PROBABILISTIC OUTPUT NODE - WRONG ARC LISTED
1 ***NODIN')
   CALL EXIT
3 WRITE(6,9)
9 FORMAT(1X,'***ERROR***PROBABILISTIC OUTPUT NODE - NO ARC INITIATED)
   CALL EXIT
4 WRITE(6,10)
10 FORMAT(1X,'***ERROR***NO INPUT RULES WERE SATISFIED***RUNSYS')
WRITE(6,13) IREPIT
13 FORMAT(1X,'***ERROR***ITERATION ',13)
WRITE(6,14)
14 FORMAT(1X,'***STATUS OF ALL ARCS FOLLOWS***/,)
DC 15 I=1,NARC
15 WRITE(6,17) ARC(I),ISTAT(I)
17 FORMAT(1X,'***ERROR***CAN NOT FIND TERMINAL NODES FOR SCAN')
   CALL EXIT
18 FORMAT(1X,'WHEREO 0 NOT INITIATED',/19X,'1 INITIATED',/19X,
1'2 COMPLETED SUCCESSFULLY',/19X,'3 FAILED',/19X,
2'4 COST + TIME VALUES ALREADY CONSIDERED')
   CALL EXIT
5 WRITE(6,11)
11 FORMAT(1X,'***ERROR***COULD NOT DETERMINE TERMINAL NODE***PTERM')
   CALL EXIT
6 WRITE(6,12)
12 FORMAT(1X,'***ERROR***LLK AND LL MUST BE EQUAL***GRAPH')
   CALL EXIT
24 WRITE(6,23)
23 FORMAT(1X,'***ERROR***NO TERMINAL NODES ARE PUNCHED TO PRINT')
   CALL EXIT
25 WRITE(6,26)
26 FORMAT(1X,'***ERROR***CAN NOT FIND TERMINAL NODES FOR SCAN')
   CALL EXIT
27 WRITE(6,28)
28 FORMAT(1X,'***ERROR***CHECK ALL ARC CARDS',/11X,'AT LEAST ONE SHOW
1 IS FIRST AND THIRD ARGUMENT EQUAL WHILE TIME DISTRIBUTION TYPE IS
2TRIANGULAR',/11X,'CHANGE TO CONSTANT')
   CALL EXIT
STOP
END

BLOCK DATA
COMMON/IDD/ RUNID(20)
COMMON/Para/ NNODE,NARC
COMMON/ARC2/ DUM(100),IDUM(100),DUMA(100)
COMMON/TERRN/ LLDUM(31),ARDUM(1000),KLDUM(510)
COMMON/ARC1/ Rdum(100), Jdum(300), Cdu(m, 500)
COMMON/LDUM/ Kdum(2,100), Zdum(1, 000), Ldum(500), Ydum(100)
COMMON/TIME/ Kjedum(11)
DATA Lui16/25*4H/
DATA Uti16/0/NARC/
DATA Bum/100*0.0/, Ldum/100*0/, Duma/100*0.0/
DATA Lldum/31*0/, Rdum/100*0.0/, Kldum/500*0/.
DATA Bldum/100*0.0/, Jdum/306*0/, Cdu(m, 500*0.0/
DATA Kdum/2100*0/, Zdum/100*0.0/, Ldum/500*0/, Ydum/100*0.0/
DATA Kjedum/11*0/
END
* LIST(STOP)
APPENDIX III

SUBROUTINES THAT MAKE UP MATHNET AND RISCA
This appendix includes a detailed flow chart for each of the 25 subroutines in RISCA**. Table I contains a listing of the 25 subroutines in the sequence of their appearance in the appendix. In addition to the name and sequence number, a brief description of the subroutine function is given in Table I.

TABLE I. SUBROUTINE FUNCTION

<table>
<thead>
<tr>
<th>Name</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Subroutine IDIN</td>
<td>Reads in run identifier card name of network.</td>
</tr>
<tr>
<td>2. Subroutine ARCIN</td>
<td>Reads in the Arc cards. Stores and orders Arc data into corresponding arrays.</td>
</tr>
<tr>
<td>4. Subroutine SCAN</td>
<td>Summarizes the input data and prints out an input listing.</td>
</tr>
<tr>
<td>5. Subroutine RUNSYS</td>
<td>Runs the simulation and prints the results.</td>
</tr>
<tr>
<td>6. Subroutine REPSET</td>
<td>Sets the number of iteration.</td>
</tr>
<tr>
<td>7. Subroutine ALLFIR(I)</td>
<td>Fires* nodes using &quot;AND&quot; output arcs. Fires all output arcs from fix node.</td>
</tr>
<tr>
<td>8. Subroutine PROFIR(I)</td>
<td>Randomly selects an arc that exits from a probability node.</td>
</tr>
</tbody>
</table>

* Initiates
**Most of these subroutines are the same as those used in MATHNET.
<table>
<thead>
<tr>
<th>Name</th>
<th>Functions</th>
</tr>
</thead>
<tbody>
<tr>
<td>9. Subroutine ARCCHK</td>
<td>Checks all of the arcs to see if they have been initiated. Checks the probability of successful completion of initiated arcs. Calculates the time for completion.</td>
</tr>
<tr>
<td></td>
<td>1 - normal distribution (sub GAUSS)</td>
</tr>
<tr>
<td></td>
<td>2 - triangular distribution (sub TRIANG)</td>
</tr>
<tr>
<td></td>
<td>3 - uniform distribution (sub UNIF)</td>
</tr>
<tr>
<td>10. Subroutine GAUSS</td>
<td>Calculates normal random variables.</td>
</tr>
<tr>
<td>11. Subroutine RANDU</td>
<td>Generates uniform random numbers.</td>
</tr>
<tr>
<td>12. Subroutine TRIANG</td>
<td>Calculates the triangularly distributed random variables.</td>
</tr>
<tr>
<td>13. Subroutine UNIF</td>
<td>Calculates the uniform random variables.</td>
</tr>
<tr>
<td>14. Subroutine NODCHK</td>
<td>Executes the output rules of those nodes whose input rules have just been satisfied by the arcs completed in subroutine ARCCHK. Determines what nodes are ready to fire. Fires those nodes that are ready.</td>
</tr>
<tr>
<td>15. Subroutine ANDTST</td>
<td>Tests &quot;AND&quot; nodes. Tests the time (t) taken by each arc and stores this information. Saves the time (t) of the longest arc to completion.</td>
</tr>
<tr>
<td>16. Subroutine ORTST</td>
<td>Tests &quot;OR&quot; nodes. Calculates cumulative time needed to satisfy input rule. Stores time (t) taken by each arc.</td>
</tr>
<tr>
<td>Name</td>
<td>Functions</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------</td>
</tr>
<tr>
<td>17. Subroutine ONEONE</td>
<td>Tests and runs one-one nodes ((i/i)).</td>
</tr>
<tr>
<td>18. Subroutine ONEBAR</td>
<td>Tests and runs one-one bar nodes ((i/i)).</td>
</tr>
<tr>
<td>19. Subroutine PREFER</td>
<td>Tests and runs preferred nodes.</td>
</tr>
<tr>
<td>20. Subroutine ITALL</td>
<td>This subroutine will handle all Terminal Node.</td>
</tr>
<tr>
<td>21. Subroutine ENDIT</td>
<td>Stores information on completed arcs.</td>
</tr>
<tr>
<td>22. Subroutine PTERM</td>
<td>Determines which terminal node was reached first in each iteration. Calculates how many times each terminal node was selected. Computes the total cost and completion time for each iteration.</td>
</tr>
<tr>
<td>23. Subroutine SGRAPH</td>
<td>Determines what is to be graphed and supplies titles for each graph. The actual graphing is done by subroutine GRAPH except for the final graph which lists the probabilities of reaching the various terminal nodes.</td>
</tr>
<tr>
<td>24. Subroutine GRAPH</td>
<td>Does the actual printing of the time and cost graphs.</td>
</tr>
<tr>
<td>25. Subroutine TERM</td>
<td>Prints all error messages in MATHNET and RISCA.</td>
</tr>
</tbody>
</table>
Subroutine 1DIN

RETURN

READ RUNID

RUN IDENTIFIER OF 80 CHARACTERS OR LESS
Subroutine ARCIN

105
Subroutine NODIN
Subroutine NODIN (Continued)
READ:
MM, (YI(I), YO(I), I = 1, MM)

DO 405
L = 1, NARC
DO 406
K = 1, MM

IF IARC(NDUM, K) = L
ARC(L) = YI(K)
NO
IF OARC(NDUM, K) = L
ARC(L) = YO(K)
NO
IF IARC NDUM, K) = 100
YES
YES
IF YI(K) = CZZ
NO
K = K + 1
NO
L = L + 1
IF K = MM
YES
IF L = NARC
YES
IF IARCI(NDUM) = IARCI(NDUM) + 1

Subroutine NODIN (Continued)
SET NUMBER OF ITERATIONS SO FAR TO 0
IREPIT=0
MNUM=999999

CALL PTERM
IREPITMREPIT + 1

DO 40
I = 1,NARC
ITERMS = 0
NOW INITIAL NODES

CALL ENDIT (KEY)
CALL ARCCHK
CALL NODCHK

YES

CALL SGGRAPH
BOOLEAN
RETURN

IF IREPIT= ITER

DO 40
I = 1,NARC
TIME (I)= 0.
ISTAT (I)=0

IF I=NARC

DO 50
I = 1,NNODE

TIMEN (I)= 0.

SUBROUTINE SCAN
RETURN

WRITE RUNID, ITER
WRITES HEADING FOR ARCS...
WRITE INFORMATION ON ARCS
WRITE INFORMATION ON NODES

CALL ALLFIR (LM)
CALL PROFIR (LM)

YES

IF OTYPE (LM)=1

IF OTYPE (LM)=2

I=I+1

CALCULATE TIME FOR NODE (I)

CALL SGGRAPH
RETURN

CALL TERM (4)

CALL PTERM IREPIT=IREPIT+ 1

IF KEY=0

ENDIT (KEY)

YES

NO

61

60

(60)

(60)

5C

5C

5A

5A

5B

5B

Subroutine RUNSYS

109
Subroutine REPSET

IF ITER > 1000
YES
STOP

READ ITER

SET NUMBER OF ITERATIONS
Subroutine ALLFIR (I)

Subroutine PROFIR (I)
Subroutine ARCHCK

112
CALCULATES NORMAL DISTR. VARIABLES

A = 0.0

DO 50 I = 1, 12

CALL RANDU(IX, IY, Y)
IX = IY
A = A + Y

IF I = 12
YES

V = (A - 6.0) * 5
S + AM

RETURN

Subroutine GAUSS (IX, S, AN, V)

GENERATES UNIFORM RANDOM DIST.

I = I + 1

NO

IF I = 0
YES

J = I * 25
J = J - (J/67108-864) * 67108864

RETURN

RAN = FLOAT (J) / 67108864.

Subroutine RANDU (I, J, RAN)
THIS ROUTINE WILL CALCULATE RANDOM TRIANG. DIST.

IF C - A = 0
CALL TERM (9)
ERROR

AM = (B - A)/(C - A)
CALL RANDU (IXT, IXY, VAL)
IXT = IXY

CALL RANDU (IXT, IXY, VAL)
IXT = IXY

XI = SQRT (AM* VAL)

XI = 1 - SQRT (1 - AM - VAL + AM* VAL)

IF VAL ≥ AM
RETURN

IF VAL ≤ AM

X = A + XI* (C - A)

RETURN

Subroutine TRIANG (IXT, A, B, C, X)

CALCULATES UNIFORMLY DIST. VARIABLES

CALL RANDU (IX, IY, Y)
IX = IY
VAL = OPT + Y* (PES - OPT)

RETURN

Subroutine UNIF (IX, OPT, PES, VAL)

114
SEE WHAT NODES ARE READY TO FIRE, FIRE THOSE THAT ARE READY

DO I
I = 1, NNODE

IL = I
J = 0

CALL ANDTST (IL, J)

CALL ORTST (IL, J)

CALL ONEONE (IL, J)

CALL ONEBAR (IL, J)

CALL PREFER (IL, J)

IF OTYPE (I) = 5
YES

IF OTYPE (I) = 7
YES

CALL ITALL (I)

CALL PROFIR (I)

CALL ALLFIR (I)

IF I = IRK
NO

 OTHERWISE

DO 22
IJ = IJ + 1

LM = IARC (I, IJ)

IF ISTAT (LM) = 1
YES

IF ISTAT (LM) = 3
YES

IF ISTAT (LM) = 2
YES

IF ISTAT (LM) = 0
YES

Subroutine NODCHK

115
Subroutine NODCHK (Continued)
Subroutine ANDTST (I, J)
Subroutine ORTST (I, J)
Subroutine ONEONE (ILK, J)
Subroutine ONEBAR (ILK, J)
Subroutine ONEBAR (ILK, J) (Continued)
Subroutine PREFER (ILK, J)
Subroutine ITALL (II)

This routine will handle a terminal node being filled first.
Subroutine ENDIT (KEY)
DETERMINES COST OF ITERATION

DO 1
I = 1, NODI
JJ = I

RETURN
YES

IF I = NARC

IF NODN (I) = INS M
I = I + 1

CALL TERM (69)

LM = NCOUNT (JJ)

IRR = IREPIT + 1
TIMEZ (IRR) = SMTIM

DO 3
I = 1, NARC

NODEZ (IRR) = INS M
COSTZ (IRR) = 0.0

COSTZ (IRR) = COSTZ (IRR) + COSTC (I) + COSTV (I) * (TIME (I) - TIMEN (LM))

IF TIME (I) > SMTIM

IF ISTAT (I) = 0

LM = NODEC (I)

IF TIMEN (LM) > SMTIM

Subroutine PTERM
Subroutine SGRAPH

126
Subroutine GRAPH(ARR,LIM)

128
IF LIM ≠ 0

YES

RETURN

NO

IF LIM ≠ 0

YES

WRITE ALL VALUES IN ARRAY ARE SAME.

NO

THIS NODE WAS NEVER A TERMINAL NODE
In MATHNET, the error message refers to an integer $t$ and is stated:

"EXECUTION TERMINATED FOR REASON $t$"

The following is a list of the possible integers and the errors they indicate:

- **a.** $t = 99$, the number of stochastic outputs from node $x$ ($x$ is stated) is not correct - (NODIN)
- **b.** $t = 3030$, no input rules were satisfied (RUNSYS)
- **c.** $t = 1210$, $1/1$ output is not correct (NODCHK)
- **d.** $t = 1211$, $1/1$ output is not correct (NODCHK)
- **e.** $t = 1212$, preferred output is not correct (NODCHK)
- **f.** $t = 100$, probabilistic output node - wrong arc listed (NODIN)
- **g.** $t = 69$, probabilistic output node - wrong number of arcs stated (PTERN)
- **h.** $t = 1$, probabilistic output node - no arc initiated (PROFIR)
- **i.** $t = 444.(6)$ number of iterations set is greater than 1000 (SGRAPH).

In RISCA, the errors which result in termination of the execution are printed as statements.

The following is a list of the RISCA error statements:

- **a.** ***ERROR*** PROBABILISTIC OUTPUT NODE - WRONG NUMBER OF ARCS STATED***NODIN
- **b.** ***ERROR*** PROBABILISTIC OUTPUT NODE - WRONG ARC LISTED***NODIN
- **c.** ***ERROR*** PROBABILISTIC OUTPUT NODE - NO ARC INITIATED***PROFIR
- **d.** ***ERROR*** NO INPUT RULES WERE SATISFIED***RUNSYS
- **e.** ***ERROR*** COULD NOT DETERMINE TERMINAL NODE***PTERM
- **f.** ***ERROR*** LLK AND LL MUST BE EQUAL***SGRAPH
g. ***ERROR*** NO TERMINAL NODES ARE PUNCHED TO PRINT
h. ***ERROR*** CAN NOT FIND TERMINAL NODES FOR SCAN
i. ***ERROR*** CHECK ALL ARC CARDS AT LEAST ONE SHOWS FIRST AND THIRD ARGUMENT EQUAL WHILE TIME DISTRIBUTION TYPE IS TRIANGULAR CHANGE TO CONSTANT.
APPENDIX IV

MATHNET OUTPUT FOR THE GOING TO WORK PROBLEM
IF YOU ARE RUNNING THIS FROM A TERMINAL
PLEASE ENTER A 1, IF RUNNING BATCH YOU SHOULD
HAVE ENTERED A CARD WITH A 0
FORMAT IS II
YOU ARE NOW IN MONITOR MODE, FROM THE FOLLOWING LIST
SELECT THE MODE YOU WISH TO GO INTO
1  ENTER NODES
2  ENTER ARCS
3  SET ITERATION NUMBER
4  SCAN THE NET SO FAR
5  RUN NET
6  ENTER RUN IDENTIFIER
9  END SESSION

ENTER A RUN IDENTIFIER OF 80 CHARACTERS OR LESS

YOU HAVE RETURNED TO MONITOR MODE
SELECT THE MODE YOU WISH TO GO INTO AS INDICATED BEFORE

YOU CAN NOW SET THE NUMBER OF ITERATIONS
ENTER A 5 POSITION INTEGER, RIGHT ADJUSTED

YOU HAVE RETURNED TO MONITOR MODE
SELECT THE MODE YOU WISH TO GO INTO AS INDICATED BEFORE

YOU ARE NOW IN ENTER ARC MODE

ENTER ARC NAME, INPUT NODE NAME, OUTPUT NODE NAME,
TIME DISTRIBUTION TYPE, TIME DISTRIBUTION ARGUMENTS 1,2,3,
CONSTANT COST COEFFICIENT, COEFFICIENT OF TIME TERM IN
PROBABILITY OF SUCCESSFUL ARC COMPLETION
FORMAT IS 3A4,11,6F10.0

TO RETURN TO MONITOR MODE ENTER RETU
YOU HAVE RETURNED TO MONITOR MODE
SELECT THE MODE YOU WISH TO GO INTO AS INDIQUEEA BEFORE

YOU ARE IN ENTER NODE MODE
ENTER NODE NAME, INPUT RULE, OUTPUT RULE
FORMAT IS A4.I1.I1
INPUT AND OUTPUT RULES ARE AS FOLLOWS

<table>
<thead>
<tr>
<th>RULE NUMBER</th>
<th>INPUT RULE</th>
<th>OUTPUT RULE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ANO</td>
<td>ALL FIRE</td>
</tr>
<tr>
<td>2</td>
<td>OR</td>
<td>PROB. FIRE</td>
</tr>
<tr>
<td>4</td>
<td>INITIAL</td>
<td>TERMINAL</td>
</tr>
<tr>
<td>5</td>
<td>1/I</td>
<td>1/I</td>
</tr>
<tr>
<td>6</td>
<td>1/I BAR</td>
<td>1/I BAR</td>
</tr>
<tr>
<td>7</td>
<td>PREFEREO</td>
<td>PREFEREO</td>
</tr>
</tbody>
</table>

YOU HAVE INDICATED A NODE WITH STOCHASTIC OUTPUTS
INPUT NUMBER OF OUTPUT ARCS, NAME OF OUTPUT ARC, PROB.
FORMAT IS I2,101(A4,F6.3)

YOU HAVE INDICATED A 1/1 NOOE
INPUT NUMBER OF ARCS, INPUT ARC NAME, OUTPUT ARC NAME
FORMAT IS I2,101(A4,A4)

YOU HAVE INDICATED A 1/1 BAR NODE
INPUT NUMBER OF ARCS, INPUT ARC NAME, OUTPUT ARC NAME
INPUT A NAME OF ZZZZ FOR THE NO INPUT CONDITION
FORMAT IS I2,101(A4,A4)

YOU HAVE INDICATED A 1/1 BAR NOOE
INPUT NUMBER OF ARCS, INPUT ARC NAME, OUTPUT ARC NAME
INPUT A NAME OF ZZZZ FOR THE NO INPUT CONDITION
FORMAT IS I2,101(A4,A4)

YOU HAVE INDICATED A 1/1 BAR NOOE
INPUT NUMBER OF ARCS, INPUT ARC NAME, OUTPUT ARC NAME
INPUT A NAME OF ZZZZ FOR THE NO INPUT CONDITION
FORMAT IS I2,101(A4,A4)

YOU HAVE INDICATED A 1/1 BAR NOOE
INPUT NUMBER OF ARCS, INPUT ARC NAME, OUTPUT ARC NAME
INPUT A NAME OF ZZZZ FOR THE NO INPUT CONDITION
FORMAT IS I2,101(A4,A4)

YOU HAVE INDICATED A 1/1 BAR NOOE
INPUT NUMBER OF ARCS, INPUT ARC NAME, OUTPUT ARC NAME
INPUT A NAME OF ZZZZ FOR THE NO INPUT CONDITION
FORMAT IS I2,101(A4,A4)
YOU HAVE INDICATED A 1/1 BAR NODE
INPUT NUMBER OF ARCS INPUT ARC NAME, OUTPUT ARC NAME
INPUT A NAME OF IZZZ FOR THE NO INPUT CONDITION
FORMAT IS 12,101A4,A4

YOU HAVE INDICATED A 1/1 BAR NODE
INPUT NUMBER OF ARCS INPUT ARC NAME, OUTPUT ARC NAME
INPUT A NAME OF IZZZ FOR THE NO INPUT CONDITION
FORMAT IS 12,101A4,A4

YOU HAVE RETURNED TO MONITOR MODE
SELECT THE MODE YOU WISH TO GO INTO AS INDICATED BEFORE

<table>
<thead>
<tr>
<th>ARC</th>
<th>INPUT NODE</th>
<th>OUTPUT NODE</th>
<th>TIME DIST</th>
<th>ARG1</th>
<th>ARG2</th>
<th>ARG3</th>
<th>COST</th>
<th>PROB</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARC1</td>
<td>N001</td>
<td>N002</td>
<td>2</td>
<td>10.00</td>
<td>12.00</td>
<td>14.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>ARC2</td>
<td>N002</td>
<td>N003</td>
<td>2</td>
<td>5.00</td>
<td>6.00</td>
<td>7.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>ARC3</td>
<td>N002</td>
<td>N003</td>
<td>2</td>
<td>11.00</td>
<td>13.00</td>
<td>15.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>ARC4</td>
<td>N003</td>
<td>N004</td>
<td>3</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>ARC5</td>
<td>N003</td>
<td>N004</td>
<td>3</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>ARC6</td>
<td>N004</td>
<td>N006</td>
<td>2</td>
<td>4.00</td>
<td>5.00</td>
<td>6.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>ARC7</td>
<td>N005</td>
<td>N007</td>
<td>3</td>
<td>3.00</td>
<td>4.00</td>
<td>5.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>ARC8</td>
<td>N005</td>
<td>N009</td>
<td>3</td>
<td>5.00</td>
<td>5.00</td>
<td>5.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>ARC9</td>
<td>N009</td>
<td>N008</td>
<td>2</td>
<td>3.00</td>
<td>4.00</td>
<td>5.00</td>
<td>0.00</td>
<td>0.900</td>
</tr>
<tr>
<td>AR10</td>
<td>N010</td>
<td>N006</td>
<td>2</td>
<td>4.00</td>
<td>5.00</td>
<td>6.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>AR11</td>
<td>N006</td>
<td>N014</td>
<td>3</td>
<td>15.00</td>
<td>15.00</td>
<td>15.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>AR12</td>
<td>N011</td>
<td>N007</td>
<td>2</td>
<td>4.00</td>
<td>5.00</td>
<td>6.00</td>
<td>0.00</td>
<td>0.800</td>
</tr>
<tr>
<td>AR13</td>
<td>N007</td>
<td>N014</td>
<td>3</td>
<td>15.00</td>
<td>15.00</td>
<td>15.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>AR14</td>
<td>N008</td>
<td>N012</td>
<td>2</td>
<td>4.00</td>
<td>5.00</td>
<td>6.00</td>
<td>0.00</td>
<td>0.600</td>
</tr>
<tr>
<td>AR15</td>
<td>N012</td>
<td>N008</td>
<td>2</td>
<td>4.00</td>
<td>5.00</td>
<td>6.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>AR16</td>
<td>N013</td>
<td>N014</td>
<td>3</td>
<td>15.00</td>
<td>15.00</td>
<td>15.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>AR17</td>
<td>N010</td>
<td>N016</td>
<td>2</td>
<td>8.00</td>
<td>9.00</td>
<td>10.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>AR18</td>
<td>N011</td>
<td>N017</td>
<td>2</td>
<td>8.00</td>
<td>9.00</td>
<td>10.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>AR19</td>
<td>N011</td>
<td>N015</td>
<td>3</td>
<td>5.00</td>
<td>5.00</td>
<td>5.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>AR20</td>
<td>N012</td>
<td>N018</td>
<td>2</td>
<td>8.00</td>
<td>9.00</td>
<td>10.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>AR21</td>
<td>N012</td>
<td>N015</td>
<td>3</td>
<td>5.00</td>
<td>5.00</td>
<td>5.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>AR22</td>
<td>N013</td>
<td>N019</td>
<td>2</td>
<td>8.00</td>
<td>9.00</td>
<td>10.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>AR23</td>
<td>N013</td>
<td>N015</td>
<td>3</td>
<td>5.00</td>
<td>5.00</td>
<td>5.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
<tr>
<td>AR24</td>
<td>N015</td>
<td>N020</td>
<td>2</td>
<td>8.00</td>
<td>9.00</td>
<td>10.00</td>
<td>0.00</td>
<td>1.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NODE</th>
<th>NO. OF INPUT ARCS</th>
<th>NO. OF OUTPUT ARCS</th>
<th>INPUT TYPE</th>
<th>OUTPUT TYPE</th>
</tr>
</thead>
<tbody>
<tr>
<td>N001</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>N002</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>N003</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>5</td>
</tr>
<tr>
<td>N004</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>N005</td>
<td>2</td>
<td>2</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>N006</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>N007</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>5</td>
</tr>
<tr>
<td>Node</td>
<td>Mode 1</td>
<td>Mode 2</td>
<td>Mode 3</td>
<td>Mode 4</td>
</tr>
<tr>
<td>------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
<tr>
<td>N009</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>N008</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>N010</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>N011</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>N012</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>N013</td>
<td>2</td>
<td>2</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>N016</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>N017</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>N015</td>
<td>3</td>
<td>1</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>N018</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>N019</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>N020</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>

* You have returned to Monitor mode.
* Select the mode you wish to go into as indicated before.
| 39.1 | 1 |
| 38.8 | 1 | 0.004 |
| 38.6 | 1 | 0.007 |
| 38.3 | 1 | 0.014 |
| 38.0 | 1 | 0.012 |
| 37.8 | 1 | 0.017 |
| 37.5 | 1 | 0.036 |
| 37.2 | 1 | 0.065 |
| 36.9 | 1 | 0.063 |
| 36.7 | 1 | 0.053 |
| 36.4 | 1 | 0.078 |
| 36.1 | 1 | 0.090 |
| 35.9 | 1 | 0.100 |
| 35.6 | 1 | 0.087 |
| 35.3 | 1 | 0.070 |
| 35.0 | 1 | 0.090 |
| 34.8 | 1 | 0.041 |
| 34.5 | 1 | 0.053 |
| 34.2 | 1 | 0.043 |
| 34.0 | 1 | 0.021 |
| 33.7 | 1 | 0.019 |
| 33.4 | 1 | 0.014 |
| 33.2 | 1 | 0.002 |
| 32.9 | 1 | 0.007 |
| 32.6 | 1 |
| 32.3 | 1 | 0.002 |

---

The mean is 35.97, the variance is 1.37, the median is 36.08, the mode is 35.99.
<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>39.1</td>
<td>1.000</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.8</td>
<td>0.995</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.6</td>
<td>0.987</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>38.3</td>
<td>0.973</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.8</td>
<td>0.960</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.5</td>
<td>0.943</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>37.2</td>
<td>0.907</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36.9</td>
<td>0.841</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36.7</td>
<td>0.778</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36.4</td>
<td>0.724</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>36.1</td>
<td>0.646</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35.9</td>
<td>0.556</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35.6</td>
<td>0.456</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35.3</td>
<td>0.368</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>35.0</td>
<td>0.297</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34.8</td>
<td>0.207</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34.5</td>
<td>0.165</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34.2</td>
<td>0.112</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>34.0</td>
<td>0.068</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33.7</td>
<td>0.046</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33.4</td>
<td>0.026</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>33.2</td>
<td>0.012</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32.9</td>
<td>0.009</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32.6</td>
<td>0.002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>32.3</td>
<td>0.002</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The mean is 35.97. The variance is 1.37. The median is 34.96. The mode is 35.99.

Graph of completion times for terminal node N16.
THE NEW DRIVING TO WORK PROBLEM

ALL VALUES IN THE ARRAY ARE IDENTICAL, AND ARE = 0.0000

GRAPH OF COMPLETION COSTS FOR TERMINAL NODE ND16

THE NEW DRIVING TO WORK PROBLEM
THE MEAN IS 43.09  THE VARIANCE IS 1.72  THE MEDIAN IS 43.00  THE MODE IS 43.36
THE MEAN IS: 43.09
THE VARIANCE IS: 1.72
THE MEDIAN IS: 41.70
THE MODE IS: 43.36
THE NEW DRIVING TO WORK PROBLEM

ALL VALUES IN THE ARRAY ARE IDENTICAL, AND ARE = 0.0000

GRAPH OF COMPLETION COSTS FOR TERMINAL NODE ND17

THE NEW DRIVING TO WORK PROBLEM
<table>
<thead>
<tr>
<th>Value</th>
<th>Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>50.5</td>
<td>1</td>
</tr>
<tr>
<td>50.3</td>
<td>1</td>
</tr>
<tr>
<td>50.2</td>
<td>1</td>
</tr>
<tr>
<td>50.1</td>
<td>1</td>
</tr>
<tr>
<td>50.0</td>
<td>1</td>
</tr>
<tr>
<td>49.8</td>
<td>1</td>
</tr>
<tr>
<td>49.7</td>
<td>1</td>
</tr>
<tr>
<td>49.6</td>
<td>1</td>
</tr>
<tr>
<td>49.5</td>
<td>1</td>
</tr>
<tr>
<td>49.4</td>
<td>1</td>
</tr>
<tr>
<td>49.3</td>
<td>1</td>
</tr>
<tr>
<td>49.2</td>
<td>1</td>
</tr>
<tr>
<td>49.1</td>
<td>1</td>
</tr>
<tr>
<td>49.0</td>
<td>1</td>
</tr>
<tr>
<td>48.9</td>
<td>1</td>
</tr>
<tr>
<td>48.8</td>
<td>1</td>
</tr>
<tr>
<td>48.7</td>
<td>1</td>
</tr>
<tr>
<td>48.6</td>
<td>1</td>
</tr>
<tr>
<td>48.5</td>
<td>1</td>
</tr>
<tr>
<td>48.4</td>
<td>1</td>
</tr>
<tr>
<td>48.3</td>
<td>1</td>
</tr>
<tr>
<td>48.2</td>
<td>1</td>
</tr>
<tr>
<td>48.1</td>
<td>1</td>
</tr>
<tr>
<td>48.0</td>
<td>1</td>
</tr>
</tbody>
</table>

**THE MEAN IS 48.20 THE VARIANCE IS 1.30 THE MEDIAN IS 47.71 THE MODE IS 50.40**

GRAPH OF COMPLETION TIMES FOR TERMINAL NODE ND18
<table>
<thead>
<tr>
<th>60.1</th>
<th>59.7</th>
<th>59.2</th>
<th>58.8</th>
<th>58.4</th>
<th>58.0</th>
<th>57.5</th>
<th>57.1</th>
<th>56.7</th>
<th>56.2</th>
<th>55.8</th>
<th>55.4</th>
<th>54.9</th>
<th>54.5</th>
<th>54.1</th>
<th>53.7</th>
<th>53.2</th>
<th>52.8</th>
<th>52.4</th>
<th>51.9</th>
<th>51.5</th>
<th>51.1</th>
<th>50.7</th>
<th>50.2</th>
<th>49.8</th>
<th>49.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.05R</td>
<td>0.05R</td>
<td>0.235</td>
<td>0.117</td>
<td>0.235</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

THE MEAN IS 51.81  THE VARIANCE IS 5.62  THE MEDIAN IS 51.32  THE MODE IS 52.15
<table>
<thead>
<tr>
<th>Completion Time</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>50.7</td>
<td>0.294</td>
</tr>
<tr>
<td>50.8</td>
<td>0.294</td>
</tr>
<tr>
<td>51.1</td>
<td>0.529</td>
</tr>
<tr>
<td>51.5</td>
<td>0.647</td>
</tr>
<tr>
<td>51.9</td>
<td>0.882</td>
</tr>
<tr>
<td>52.4</td>
<td>0.882</td>
</tr>
<tr>
<td>54.3</td>
<td>0.882</td>
</tr>
<tr>
<td>54.9</td>
<td>0.882</td>
</tr>
<tr>
<td>55.0</td>
<td>0.882</td>
</tr>
<tr>
<td>55.4</td>
<td>0.882</td>
</tr>
<tr>
<td>55.8</td>
<td>0.882</td>
</tr>
<tr>
<td>56.2</td>
<td>0.882</td>
</tr>
<tr>
<td>56.8</td>
<td>0.882</td>
</tr>
<tr>
<td>57.5</td>
<td>0.882</td>
</tr>
<tr>
<td>58.0</td>
<td>0.882</td>
</tr>
<tr>
<td>58.4</td>
<td>0.882</td>
</tr>
<tr>
<td>58.8</td>
<td>0.882</td>
</tr>
<tr>
<td>59.2</td>
<td>0.941</td>
</tr>
<tr>
<td>59.7</td>
<td>0.941</td>
</tr>
<tr>
<td>60.1</td>
<td>0.941</td>
</tr>
<tr>
<td>60.4</td>
<td>0.941</td>
</tr>
</tbody>
</table>

**Graph of Completion Times for Terminal Node No. 19**

The mean is 51.81, the variance is 5.62, the median is 50.46, and the mode is 52.15.
<p>| | | | | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>65.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>64.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>63.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>62.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>61.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>60.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>59.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>58.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>57.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>57.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>56.3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>55.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>54.1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>53.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>52.6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51.9</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>51.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>50.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49.7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>49.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>48.2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>47.5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46.8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The mean is 55.57, the variance is 14.97, the median is 56.16, the mode is 56.70.
THE MEAN IS 55.57 THE VARIANCE IS 14.97 THE MEDIAN IS 51.72 THE MODE IS 56.70

GRAPH OF COMPLETION TIMES FOR TERMINAL NODE 0020
| 65.2 | 1 |
| 63.9 | 1 | 0.002 |
| 62.5 | 1 | 0.006 |
| 61.2 | 1 | 0.002 |
| 59.9 | 1 | 0.002 |
| 58.6 | 1 |
| 57.3 | 1 | 0.010 |
| 56.0 | 1 | 0.028 |
| 54.7 | 1 | 0.012 |
| 53.4 | 1 | 0.012 |
| 52.0 | 1 | 0.014 |
| 50.7 | 1 | 0.016 |
| 49.4 | 1 | 0.012 |
| 48.1 | 1 | 0.006 |
| 46.8 | 1 | 0.010 |
| 45.5 | 1 | 0.006 |
| 44.2 | 1 | 0.006 |
| 42.8 | 1 | 0.016 |
| 41.5 | 1 | 0.014 |
| 40.2 | 1 | 0.006 |
| 38.9 | 1 | 0.002 |
| 37.6 | 1 | 0.062 |
| 36.3 | 1 | 0.264 |
| 35.0 | 1 | 0.334 |
| 33.7 | 1 | 0.137 |
| 32.3 | 1 | 0.020 |

THE MEAN IS 38.72 THE VARIANCE IS 42.12 THE MEDIAN IS 37.14 THE MODE IS 35.63
<table>
<thead>
<tr>
<th>Completion Time (in minutes)</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>65.9</td>
<td>1</td>
</tr>
<tr>
<td>63.3</td>
<td>1</td>
</tr>
<tr>
<td>62.1</td>
<td>1</td>
</tr>
<tr>
<td>61.2</td>
<td>1</td>
</tr>
<tr>
<td>59.9</td>
<td>1</td>
</tr>
<tr>
<td>58.6</td>
<td>1</td>
</tr>
<tr>
<td>57.3</td>
<td>1</td>
</tr>
<tr>
<td>56.0</td>
<td>1</td>
</tr>
<tr>
<td>54.7</td>
<td>1</td>
</tr>
<tr>
<td>53.4</td>
<td>1</td>
</tr>
<tr>
<td>52.0</td>
<td>1</td>
</tr>
<tr>
<td>50.7</td>
<td>1</td>
</tr>
<tr>
<td>49.4</td>
<td>1</td>
</tr>
<tr>
<td>48.1</td>
<td>1</td>
</tr>
<tr>
<td>46.8</td>
<td>1</td>
</tr>
<tr>
<td>45.5</td>
<td>1</td>
</tr>
<tr>
<td>44.2</td>
<td>1</td>
</tr>
<tr>
<td>42.8</td>
<td>1</td>
</tr>
<tr>
<td>41.5</td>
<td>1</td>
</tr>
<tr>
<td>40.2</td>
<td>1</td>
</tr>
<tr>
<td>38.9</td>
<td>1</td>
</tr>
<tr>
<td>37.6</td>
<td>1</td>
</tr>
<tr>
<td>36.3</td>
<td>1</td>
</tr>
<tr>
<td>35.0</td>
<td>1</td>
</tr>
<tr>
<td>33.7</td>
<td>1</td>
</tr>
<tr>
<td>32.3</td>
<td>1</td>
</tr>
</tbody>
</table>

The mean is 38.72, the variance is 42.12, the median is 35.32, and the mode is 35.63.

Graph of completion times for all nodes.
GRAPH OF NODE PROBABILITIES

THE NEW DRIVING TO WORK PROBLEM

* YOU HAVE RETURNED TO MONITOR MODE
SELECT THE MODE YOU WISH TO GO INTO AS INDICATED BEFORE

* Next page is blank.
APPENDIX V

RISCA OUTPUT FOR THE GOING TO WORK PROBLEM

Next page is blank.
# THE NEW DRIVING TO WORK PROBLEM

## 500 ITERATIONS

<table>
<thead>
<tr>
<th>ARC</th>
<th>INP NODE</th>
<th>OUT NODE</th>
<th>TIME DIST</th>
<th>ARG1</th>
<th>ARG2</th>
<th>ARG3</th>
<th>COST</th>
<th>P OF COMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARC1</td>
<td>NOD1</td>
<td>NOD2</td>
<td>TRI</td>
<td>10.00</td>
<td>12.00</td>
<td>14.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC2</td>
<td>NOD2</td>
<td>NOD3</td>
<td>TRI</td>
<td>5.00</td>
<td>6.00</td>
<td>7.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC3</td>
<td>NOD2</td>
<td>NOD3</td>
<td>TRI</td>
<td>11.00</td>
<td>13.00</td>
<td>15.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC4</td>
<td>NOD3</td>
<td>NOD4</td>
<td>UNIF</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC5</td>
<td>NOD3</td>
<td>NOD5</td>
<td>UNIF</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC6</td>
<td>NOD4</td>
<td>NOD6</td>
<td>TRI</td>
<td>3.00</td>
<td>4.00</td>
<td>5.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC7</td>
<td>NOD5</td>
<td>NOD6</td>
<td>TRI</td>
<td>3.00</td>
<td>4.00</td>
<td>5.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC8</td>
<td>NOD5</td>
<td>NOD9</td>
<td>UNIF</td>
<td>5.00</td>
<td>5.00</td>
<td>5.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC9</td>
<td>NOD9</td>
<td>NOD10</td>
<td>TRI</td>
<td>3.00</td>
<td>4.00</td>
<td>5.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC10</td>
<td>NOD6</td>
<td>NOD10</td>
<td>TRI</td>
<td>4.00</td>
<td>5.00</td>
<td>6.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC11</td>
<td>NOD6</td>
<td>NOD14</td>
<td>UNIF</td>
<td>15.00</td>
<td>15.00</td>
<td>15.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC12</td>
<td>NOD7</td>
<td>NOD11</td>
<td>TRI</td>
<td>4.00</td>
<td>5.00</td>
<td>6.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC13</td>
<td>NOD7</td>
<td>NOD14</td>
<td>UNIF</td>
<td>15.00</td>
<td>15.00</td>
<td>15.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC14</td>
<td>NOD8</td>
<td>NOD12</td>
<td>TRI</td>
<td>4.00</td>
<td>5.00</td>
<td>6.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC15</td>
<td>NOD8</td>
<td>NOD14</td>
<td>UNIF</td>
<td>15.00</td>
<td>15.00</td>
<td>15.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC16</td>
<td>NOD8</td>
<td>NOD13</td>
<td>TRI</td>
<td>4.00</td>
<td>5.00</td>
<td>6.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC17</td>
<td>NOD10</td>
<td>NOD16</td>
<td>TRI</td>
<td>8.00</td>
<td>9.00</td>
<td>10.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC18</td>
<td>NOD11</td>
<td>NOD17</td>
<td>TRI</td>
<td>8.00</td>
<td>9.00</td>
<td>10.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC19</td>
<td>NOD11</td>
<td>NOD15</td>
<td>UNIF</td>
<td>5.00</td>
<td>5.00</td>
<td>5.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC20</td>
<td>NOD12</td>
<td>NOD18</td>
<td>TRI</td>
<td>8.00</td>
<td>9.00</td>
<td>10.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC21</td>
<td>NOD12</td>
<td>NOD15</td>
<td>UNIF</td>
<td>5.00</td>
<td>5.00</td>
<td>5.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC22</td>
<td>NOD13</td>
<td>NOD19</td>
<td>TRI</td>
<td>8.00</td>
<td>9.00</td>
<td>10.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC23</td>
<td>NOD13</td>
<td>NOD15</td>
<td>UNIF</td>
<td>5.00</td>
<td>5.00</td>
<td>5.00</td>
<td>0.00</td>
<td>+</td>
</tr>
<tr>
<td>ARC24</td>
<td>NOD15</td>
<td>NOD20</td>
<td>TRI</td>
<td>8.00</td>
<td>9.00</td>
<td>10.00</td>
<td>0.00</td>
<td>+</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>NODE</th>
<th>NO. OF INPUT ARCS</th>
<th>NO. OF OUTPUT ARCS</th>
<th>INPUT RULE</th>
<th>OUTPUT RULE</th>
</tr>
</thead>
<tbody>
<tr>
<td>NOD1</td>
<td>0</td>
<td>1</td>
<td>INIT</td>
<td>ALL</td>
</tr>
<tr>
<td>NOD2</td>
<td>1</td>
<td>2</td>
<td>AND</td>
<td>PROB</td>
</tr>
<tr>
<td>NOD3</td>
<td>2</td>
<td>2</td>
<td>1-1</td>
<td>1-1</td>
</tr>
<tr>
<td>NOD4</td>
<td>1</td>
<td>1</td>
<td>AND</td>
<td>ALL</td>
</tr>
<tr>
<td>NOD5</td>
<td>2</td>
<td>2</td>
<td>1-1-B</td>
<td>1-1-B</td>
</tr>
<tr>
<td>NOD6</td>
<td>2</td>
<td>2</td>
<td>1-1-B</td>
<td>1-1-B</td>
</tr>
<tr>
<td>NOD7</td>
<td>2</td>
<td>2</td>
<td>1-1-B</td>
<td>1-1-B</td>
</tr>
<tr>
<td>NOD9</td>
<td>1</td>
<td>1</td>
<td>AND</td>
<td>ALL</td>
</tr>
<tr>
<td>NOD8</td>
<td>2</td>
<td>2</td>
<td>1-1-B</td>
<td>1-1-B</td>
</tr>
<tr>
<td>NOD10</td>
<td>1</td>
<td>1</td>
<td>AND</td>
<td>ALL</td>
</tr>
<tr>
<td>NOD14</td>
<td>3</td>
<td>1</td>
<td>OR</td>
<td>ALL</td>
</tr>
<tr>
<td>NOD11</td>
<td>2</td>
<td>2</td>
<td>1-1-B</td>
<td>1-1-B</td>
</tr>
<tr>
<td>NOD12</td>
<td>2</td>
<td>2</td>
<td>1-1-B</td>
<td>1-1-B</td>
</tr>
<tr>
<td>NOD13</td>
<td>2</td>
<td>2</td>
<td>1-1-B</td>
<td>1-1-B</td>
</tr>
<tr>
<td>NOD16</td>
<td>1</td>
<td>0</td>
<td>AND</td>
<td>TERM</td>
</tr>
<tr>
<td>NOD17</td>
<td>1</td>
<td>0</td>
<td>AND</td>
<td>TERM</td>
</tr>
<tr>
<td>NOD15</td>
<td>3</td>
<td>1</td>
<td>OR</td>
<td>ALL</td>
</tr>
<tr>
<td>NOD18</td>
<td>1</td>
<td>0</td>
<td>AND</td>
<td>TERM</td>
</tr>
<tr>
<td>NOD19</td>
<td>1</td>
<td>0</td>
<td>AND</td>
<td>TERM</td>
</tr>
<tr>
<td>NOD20</td>
<td>1</td>
<td>0</td>
<td>AND</td>
<td>TERM</td>
</tr>
<tr>
<td>Value</td>
<td>Mean</td>
<td>Variance</td>
<td>Standard Deviation</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>------</td>
<td>----------</td>
<td>--------------------</td>
<td></td>
</tr>
<tr>
<td>0.004</td>
<td>0.07</td>
<td>1.444</td>
<td>1.202</td>
<td></td>
</tr>
<tr>
<td>0.019</td>
<td>0.023</td>
<td>0.35</td>
<td>0.57</td>
<td></td>
</tr>
<tr>
<td>0.033</td>
<td>0.045</td>
<td>0.095</td>
<td>0.31</td>
<td></td>
</tr>
<tr>
<td>0.083</td>
<td>0.107</td>
<td>0.069</td>
<td>0.26</td>
<td></td>
</tr>
<tr>
<td>0.088</td>
<td>0.042</td>
<td>0.083</td>
<td>0.29</td>
<td></td>
</tr>
<tr>
<td>0.057</td>
<td>0.041</td>
<td>0.050</td>
<td>0.23</td>
<td></td>
</tr>
<tr>
<td>0.023</td>
<td>0.033</td>
<td>0.011</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td>0.004</td>
<td>0.004</td>
<td>0.011</td>
<td>0.10</td>
<td></td>
</tr>
<tr>
<td>0.004</td>
<td>0.004</td>
<td>0.004</td>
<td>0.06</td>
<td></td>
</tr>
<tr>
<td>0.022</td>
<td>0.002</td>
<td>0.002</td>
<td>0.05</td>
<td></td>
</tr>
</tbody>
</table>

Mean = 35.983
Variance = 1.444
Standard Deviation = 1.202
<table>
<thead>
<tr>
<th>Time (sec)</th>
<th>Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>38.078</td>
<td>1</td>
</tr>
<tr>
<td>38.711</td>
<td>1</td>
</tr>
<tr>
<td>38.444</td>
<td>1</td>
</tr>
<tr>
<td>38.177</td>
<td>1</td>
</tr>
<tr>
<td>37.910</td>
<td>1</td>
</tr>
<tr>
<td>37.644</td>
<td>1</td>
</tr>
<tr>
<td>37.377</td>
<td>1</td>
</tr>
<tr>
<td>37.110</td>
<td>1</td>
</tr>
<tr>
<td>36.843</td>
<td>1</td>
</tr>
<tr>
<td>36.576</td>
<td>1</td>
</tr>
<tr>
<td>36.309</td>
<td>1</td>
</tr>
<tr>
<td>36.042</td>
<td>1</td>
</tr>
<tr>
<td>35.776</td>
<td>1</td>
</tr>
<tr>
<td>35.509</td>
<td>1</td>
</tr>
<tr>
<td>35.242</td>
<td>1</td>
</tr>
<tr>
<td>34.975</td>
<td>1</td>
</tr>
<tr>
<td>34.708</td>
<td>1</td>
</tr>
<tr>
<td>34.441</td>
<td>1</td>
</tr>
<tr>
<td>34.175</td>
<td>1</td>
</tr>
<tr>
<td>33.908</td>
<td>1</td>
</tr>
<tr>
<td>33.641</td>
<td>1</td>
</tr>
<tr>
<td>33.374</td>
<td>1</td>
</tr>
<tr>
<td>33.107</td>
<td>1</td>
</tr>
<tr>
<td>32.840</td>
<td>1</td>
</tr>
<tr>
<td>32.574</td>
<td>1</td>
</tr>
<tr>
<td>32.307</td>
<td>1</td>
</tr>
</tbody>
</table>

**Mean = 35.983**  
**Variance = 1.444**  
**Standard Deviation = 1.202**

*Graph of completion times for terminal node No. 16*
The new driving to work problem

All values in the array are identical, and are 0.0000

Graph of completion costs for terminal node ND16
<table>
<thead>
<tr>
<th>Value</th>
<th>1.029</th>
</tr>
</thead>
<tbody>
<tr>
<td>45.255</td>
<td>1.029</td>
</tr>
<tr>
<td>45.059</td>
<td>1.029</td>
</tr>
<tr>
<td>44.863</td>
<td></td>
</tr>
<tr>
<td>44.668</td>
<td>1.029</td>
</tr>
<tr>
<td>44.472</td>
<td>1.058</td>
</tr>
<tr>
<td>44.276</td>
<td>1.029</td>
</tr>
<tr>
<td>44.080</td>
<td>1.029</td>
</tr>
<tr>
<td>43.885</td>
<td>1.029</td>
</tr>
<tr>
<td>43.689</td>
<td>1.058</td>
</tr>
<tr>
<td>43.493</td>
<td>1.088</td>
</tr>
<tr>
<td>43.298</td>
<td>1.088</td>
</tr>
<tr>
<td>43.102</td>
<td>1.029</td>
</tr>
<tr>
<td>42.906</td>
<td>1.058</td>
</tr>
<tr>
<td>42.711</td>
<td>1.058</td>
</tr>
<tr>
<td>42.515</td>
<td>1.117</td>
</tr>
<tr>
<td>42.319</td>
<td>1.058</td>
</tr>
<tr>
<td>42.124</td>
<td></td>
</tr>
<tr>
<td>41.928</td>
<td>1.058</td>
</tr>
<tr>
<td>41.732</td>
<td>1.029</td>
</tr>
<tr>
<td>41.537</td>
<td></td>
</tr>
<tr>
<td>41.341</td>
<td>1.029</td>
</tr>
<tr>
<td>41.145</td>
<td>1.029</td>
</tr>
<tr>
<td>40.950</td>
<td></td>
</tr>
<tr>
<td>40.754</td>
<td></td>
</tr>
<tr>
<td>40.558</td>
<td>1.029</td>
</tr>
<tr>
<td>40.363</td>
<td>1.029</td>
</tr>
</tbody>
</table>

**Mean** = 43.080  **Variance** = 1.455  **Standard Deviation** = 1.206
45.255  |  45.059  |  44.563  |  44.668  |  44.472  |  44.276  |  44.080  |  43.885  |  43.689  |  43.493  |  43.298  |  43.102  |  42.906  |  42.711  |  42.515  |  42.319  |  42.124  |  41.928  |  41.732  |  41.537  |  41.341  |  41.145  |  40.950  |  40.754  |  40.558  |  40.363  |
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>0.000</td>
<td>0.970</td>
<td>0.941</td>
<td>0.941</td>
<td>0.911</td>
<td>0.852</td>
<td>0.823</td>
<td>0.794</td>
<td>0.764</td>
<td>0.705</td>
<td>0.617</td>
<td>0.529</td>
<td>0.500</td>
<td>0.441</td>
<td>0.382</td>
<td>0.264</td>
<td>0.205</td>
<td>0.205</td>
<td>0.147</td>
<td>0.117</td>
<td>0.117</td>
<td>0.088</td>
<td>0.058</td>
<td>0.058</td>
<td>0.058</td>
</tr>
</tbody>
</table>

MEAN = 43.080  VARIANCE = 1.455  STANDARD DEVIATION = 1.263

GRAPH OF COMPLETION TIMES FOR TERMINAL NODE ND17
ALL VALUES IN THE ARRAY ARE IDENTICAL, AND ARE = 0.0000

GRAPH OF COMPLETION COSTS FOR TERMINAL NODE NP17

THE NEW DRIVING TO WORK PROBLEM
<table>
<thead>
<tr>
<th>Completion Time (s)</th>
<th>Completion Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>49.509</td>
<td>900</td>
</tr>
<tr>
<td>49.396</td>
<td>666</td>
</tr>
<tr>
<td>49.282</td>
<td>666</td>
</tr>
<tr>
<td>49.169</td>
<td>666</td>
</tr>
<tr>
<td>49.055</td>
<td>666</td>
</tr>
<tr>
<td>48.941</td>
<td>666</td>
</tr>
<tr>
<td>48.828</td>
<td>666</td>
</tr>
<tr>
<td>48.714</td>
<td>666</td>
</tr>
<tr>
<td>48.601</td>
<td>333</td>
</tr>
<tr>
<td>48.487</td>
<td>333</td>
</tr>
<tr>
<td>48.373</td>
<td>333</td>
</tr>
<tr>
<td>48.260</td>
<td>333</td>
</tr>
<tr>
<td>48.146</td>
<td>333</td>
</tr>
<tr>
<td>48.033</td>
<td>333</td>
</tr>
<tr>
<td>47.919</td>
<td>333</td>
</tr>
<tr>
<td>47.805</td>
<td>333</td>
</tr>
<tr>
<td>47.692</td>
<td>333</td>
</tr>
<tr>
<td>47.578</td>
<td>333</td>
</tr>
<tr>
<td>47.465</td>
<td>333</td>
</tr>
<tr>
<td>47.351</td>
<td>333</td>
</tr>
<tr>
<td>47.237</td>
<td>333</td>
</tr>
<tr>
<td>47.124</td>
<td>333</td>
</tr>
<tr>
<td>47.010</td>
<td>333</td>
</tr>
<tr>
<td>46.897</td>
<td>333</td>
</tr>
<tr>
<td>46.783</td>
<td>333</td>
</tr>
<tr>
<td>46.669</td>
<td>333</td>
</tr>
</tbody>
</table>

**Graph of Completion Times for Terminal Node No. 18**

- **Mean (M)**: 48.327
- **Variance (V)**: 2.185
- **Standard Deviation (SD)**: 1.478
| 58.924 |
| 58.564 |
| 58.213 |
| 57.857 |
| 57.501 |
| 57.146 |
| 56.790 |
| 56.434 |
| 56.079 |
| 55.723 |
| 55.367 |
| 55.012 |
| 54.656 |
| 54.300 |
| 53.945 |
| 53.589 |
| 53.233 |
| 52.877 |
| 52.522 |
| 52.166 |
| 51.810 |
| 51.455 |
| 51.099 |
| 50.743 |
| 50.388 |
| 50.032 |

**Mean** = 52.276  
**Variance** = 5.918  
**Standard Deviation** = 2.433
GRAPH OF COMPLETION TIMES FOR TERMINAL NODE ND19

MEAN = 52.276  VARIANCE = 5.918  STANDARD DEVIATION = 2.433
<table>
<thead>
<tr>
<th>Value</th>
<th>Mean</th>
<th>Variance</th>
<th>Standard Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td>64.078</td>
<td>54.609</td>
<td>16.698</td>
<td>4.086</td>
</tr>
<tr>
<td>64,775</td>
<td>63,381</td>
<td>62,684</td>
<td>61,986</td>
</tr>
<tr>
<td>--------</td>
<td>--------</td>
<td>--------</td>
<td>--------</td>
</tr>
</tbody>
</table>

Mean = 54.600  VARIANCE = 16.698  STANDARD DEVIATION = 4.086

GRAPH OF COMPLETION TIMES FOR TERMINAL NODE ND20
<table>
<thead>
<tr>
<th>Value</th>
<th>.002</th>
</tr>
</thead>
<tbody>
<tr>
<td>64.078</td>
<td>1</td>
</tr>
<tr>
<td>62.808</td>
<td>1</td>
</tr>
<tr>
<td>61.537</td>
<td>1</td>
</tr>
<tr>
<td>60.266</td>
<td>1</td>
</tr>
<tr>
<td>58.995</td>
<td>1</td>
</tr>
<tr>
<td>57.724</td>
<td>1</td>
</tr>
<tr>
<td>56.453</td>
<td>1</td>
</tr>
<tr>
<td>55.182</td>
<td>1</td>
</tr>
<tr>
<td>53.911</td>
<td>1</td>
</tr>
<tr>
<td>52.641</td>
<td>1</td>
</tr>
<tr>
<td>51.370</td>
<td>1</td>
</tr>
<tr>
<td>50.099</td>
<td>1</td>
</tr>
<tr>
<td>48.828</td>
<td>1</td>
</tr>
<tr>
<td>47.557</td>
<td>1</td>
</tr>
<tr>
<td>46.286</td>
<td>1</td>
</tr>
<tr>
<td>45.015</td>
<td>1</td>
</tr>
<tr>
<td>43.745</td>
<td>1</td>
</tr>
<tr>
<td>42.474</td>
<td>1</td>
</tr>
<tr>
<td>41.203</td>
<td>1</td>
</tr>
<tr>
<td>39.932</td>
<td>1</td>
</tr>
<tr>
<td>38.661</td>
<td>1</td>
</tr>
<tr>
<td>37.390</td>
<td>1</td>
</tr>
<tr>
<td>36.119</td>
<td>1</td>
</tr>
<tr>
<td>34.848</td>
<td>1</td>
</tr>
<tr>
<td>33.578</td>
<td>1</td>
</tr>
<tr>
<td>32.307</td>
<td>1</td>
</tr>
</tbody>
</table>

Mean = 39.127  Variance = 31.020  Standard Deviation = 5.570
THE NEW DRIVING TO WORK PROBLEM

ALL VALUES IN THE ARRAY ARE IDENTICAL, AND ARE = 0.0000

GRAPH OF COMPLETION COSTS FOR ALL NODES

THE NEW DRIVING TO WORK PROBLEM
<table>
<thead>
<tr>
<th>No. of Copies</th>
<th>Organization</th>
<th>No. of Copies</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>12</td>
<td>Commander</td>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>Defense</td>
<td>U.S. Army</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Documentation</td>
<td>Materiel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Center</td>
<td>Command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATTN:</td>
<td>ATTN:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TIPCR</td>
<td>AMCRD</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Cameron</td>
<td>D.C. 20315</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Station</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Alexandria,</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Virginia 22314</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>U.S. Army</td>
<td>U.S. Army</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Materiel</td>
<td>Materiel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command</td>
<td>Command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATTN:</td>
<td>ATTN:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AMCDL</td>
<td>AMCRD-P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Washington,</td>
<td>D.C. 20315</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D.C. 20315</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>U.S. Army</td>
<td>U.S. Army</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Materiel</td>
<td>Materiel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command</td>
<td>Command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATTN:</td>
<td>ATTN:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AMCMCA</td>
<td>AMCRD-W</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Washington,</td>
<td>D.C. 20315</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D.C. 20315</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>U.S. Army</td>
<td>U.S. Army</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Materiel</td>
<td>Materiel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command</td>
<td>Command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATTN:</td>
<td>ATTN:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AMCMA</td>
<td>AMCRD-T</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Washington,</td>
<td>D.C. 20315</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D.C. 20315</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>U.S. Army</td>
<td>U.S. Army</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Materiel</td>
<td>Materiel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command</td>
<td>Command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATTN:</td>
<td>ATTN:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AMCSA-PM-MBT</td>
<td>AMCRD-R</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Washington,</td>
<td>D.C. 20315</td>
<td></td>
</tr>
<tr>
<td></td>
<td>D.C. 20315</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>U.S. Army</td>
<td>U.S. Army</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Materiel</td>
<td>Materiel</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Command</td>
<td>Command</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ATTN:</td>
<td>ATTN:</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AMCPA-SA</td>
<td>AMCRD-P</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Picatinny</td>
<td>Washington,</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Arsenal</td>
<td>D.C. 20315</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dover, New</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Jersey 07801</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No. of</td>
<td>Organization</td>
<td>No. of</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>--------------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>Copies</td>
<td></td>
<td>Copies</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Director</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MUCOM Operations Research Group</td>
<td>Commander</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Edgewood Arsenal, Maryland 21010</td>
<td>U.S. Army Electronics Command</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>U.S. Army Harry Diamond Laboratories</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Washington, D.C. 20438</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Frankford Arsenal</td>
<td>Commander</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Philadelphia, Pennsylvania 19137</td>
<td>U.S. Army Missile Command</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Frankford Arsenal</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Philadelphia, Pennsylvania 19137</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Commander</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Picatinny Arsenal Plans Office</td>
<td>Commander</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dover, New Jersey 07801</td>
<td>U.S. Army Munitions Command</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Chief, Analytical Sciences Office</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>U.S. Army Biological Defense Research Laboratory</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Dugway, Utah 84022</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Commander</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>U.S. Army Aviation Systems Command</td>
<td>Commander</td>
<td></td>
</tr>
<tr>
<td></td>
<td>P.O. Box 209 St. Louis, Missouri 63166</td>
<td>U.S. Army Management Engineering Training Agency</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

184
<table>
<thead>
<tr>
<th>No. of Copies</th>
<th>Organization</th>
<th>No. of Copies</th>
<th>Organization</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Commander</td>
<td>1</td>
<td>Commander</td>
</tr>
<tr>
<td></td>
<td>U.S. Army Logistics Management Center</td>
<td></td>
<td>Department of Mathematics</td>
</tr>
<tr>
<td></td>
<td>ATTN: CE&amp;AD/SLS (Dr. Runnion)</td>
<td></td>
<td>ATTN: COL Lochry</td>
</tr>
<tr>
<td></td>
<td>Fort Lee, Virginia 23801</td>
<td></td>
<td>U.S. Air Force Academy</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Colorado 80840</td>
</tr>
</tbody>
</table>

_Aberdeen Proving Ground_

<p>| Commander, USATECOM | ATTN: AMSTE-TS | AMSTE-SY (G. Schroeter) |</p>
<table>
<thead>
<tr>
<th>AD</th>
<th>MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL</th>
</tr>
</thead>
<tbody>
<tr>
<td>U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED</td>
<td></td>
</tr>
<tr>
<td>MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&amp;E IP765801MMI102, November 1972</td>
<td></td>
</tr>
</tbody>
</table>

Trace Index Terms
- Network Analysis
- Simulation
- Risk Analysis
- MATHNET
- RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNET 5. RISCA

<table>
<thead>
<tr>
<th>AD</th>
<th>Brooks, Wilbert J.; Foster, Ward V. Maruyama, Richard T.</th>
</tr>
</thead>
<tbody>
<tr>
<td>U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED</td>
<td></td>
</tr>
<tr>
<td>MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DDC; Project No. RDT&amp;E IP765801MMI102, November 1972</td>
<td></td>
</tr>
</tbody>
</table>

Trace Index Terms
- Network Analysis
- Simulation
- Risk Analysis
- MATHNET
- RISCA

1. Network Analysis. 2. Simulation. 3. Risk Analysis. 4. MATHNET 5. RISCA
### Risk Analysis

**U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED**

**MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DOC; Project No. RDT&E IP765801MM1102, November 1972**

<table>
<thead>
<tr>
<th>Trace Index Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Analysis</td>
</tr>
<tr>
<td>Simulation</td>
</tr>
<tr>
<td>Risk Analysis</td>
</tr>
<tr>
<td>MATHNET</td>
</tr>
<tr>
<td>RISCA</td>
</tr>
</tbody>
</table>


---

### MATHNET

**U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED**

**MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DOC; Project No. RDT&E IP765801MM1102, November 1972**

<table>
<thead>
<tr>
<th>Trace Index Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Analysis</td>
</tr>
<tr>
<td>Simulation</td>
</tr>
<tr>
<td>Risk Analysis</td>
</tr>
<tr>
<td>MATHNET</td>
</tr>
<tr>
<td>RISCA</td>
</tr>
</tbody>
</table>


---

### RISCA

**U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005, AMSAA Technical Memorandum 151, Report UNCLASSIFIED**

**MATHNET AND RISCA (NETWORK ANALYZER PROGRAMS) A USERS' MANUAL, Wilbert J. Brooks, Ward V. Foster, Richard T. Maruyama, AMSAA; DOC; Project No. RDT&E IP765801MM1102, November 1972**

<table>
<thead>
<tr>
<th>Trace Index Terms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Analysis</td>
</tr>
<tr>
<td>Simulation</td>
</tr>
<tr>
<td>Risk Analysis</td>
</tr>
<tr>
<td>MATHNET</td>
</tr>
<tr>
<td>RISCA</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Trace Index Terms</th>
<th>Risk Analysis</th>
<th>MATHNET</th>
<th>AD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Analysis</td>
<td>1.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Simulation</td>
<td>2.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Risk Analysis</td>
<td>3.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>RISCA</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

U.S. Army Materiel Systems Analysis Agency (AMSAA), Aberdeen Proving Ground, Maryland 21005; AMSAA Technical Memorandum 151, Report UNCLASSIFIED

Network Analysis
Simulation
Risk Analysis
RISCA


November 1972

1. Network Analysis
2. Simulation
3. Risk Analysis
4. MATHNET
5. RISCA

Trace Index Terms

1. Network Analysis
2. Simulation
3. Risk Analysis
4. MATHNET
5. RISCA
Two network analyzer programs, MATHNET and RISCA, which allow the analyst to simulate a general class of network representations are described and evaluated for the potential user.

Network concepts, program listings, and program flow charts are included for both programs in addition to detailed description of input preparation and output interpretation for a hypothetical example.
<table>
<thead>
<tr>
<th>KEY WORDS</th>
<th>LINK A</th>
<th>LINK B</th>
<th>LINK C</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>ROLE</td>
<td>WT</td>
<td>ROLE</td>
</tr>
<tr>
<td>NETWORK ANALYSIS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SIMULATION</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RISK ANALYSIS</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MATINET</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>RISCA</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>