(NASA-CR-169531) THE GRASP 3: GRAPHICAL RELIABILITY ANALYSIS SIMULATION PROGRAM. VERSION 3: A USERS' MANUAL AND MODELLING GUIDE (Purdue Univ.) 305 p HC A14/MF A01 CSCL 09B
N83-13831
Unclas
G3/61 01175

GRASP III

GRAPHICAL RELIABILITY ANALYSIS SIMULATION PROGRAM

VERSION III

A USERS' MANUAL AND MODELLING GUIDE

PREPARED BY

DON T. PHILLIPS, Ph.D., P.E.

BELKACEM MANSEUR

JOSEPH W. FOSTER, D.E., P.E.

JULY 1982

GRASP III
GENERALIZED RELIABILITY ANALYSIS
SIMULATION PROGRAM
VERSION III

A USERS' MANUAL

AND

MODELLING GUIDE

PREPARED BY

DON T. PHILLIPS, PH.D., P.E.

BELKACEM MANSEUR

JOSEPH W. FOSTER, D.E., P.E.

FOR

NATIONAL AERONAUTICS

AND

SPACE ADMINISTRATION

JULY 1982

PROJECT MONITOR:

MR. WILLIAM L. ANDRE

U.S. ARMY RESEARCH & TECHNOLOGY

LABORATORIES (AVRADCOM)

AMES RESEARCH CENTER

MOFFETT FIELD, CALIFORNIA 94035

ACKNOWLEDGEMENTS

GRASP III is version three of the program GRASP (Graphical Reliability Analysis Simulation Program). GRASP was originally conceived and created by Dr. Don T. Phillips at Purdue University in 1973. Initial programming efforts were conducted by Mr. Edward Shulaker and later by Mr. Joseph Polito, under sponsorship of National Science Foundation Grant ATA 73-07822-A01 with research directed jointly by Dr. Don T. Phillips and Dr. Clifford C. Peterson of Purdue University.

Drs. Don T. Phillips and Joseph W. Foster worked, since 1977, to determine the applicability of GRASP to model reliability and survivability characteristics of aircraft weapon systems. This work is sponsored by a NASA grant NSG-2226 initiated by Mr. William L. Andre at the US Army Research and Technology Laboratories. An amount of $6,000 was received 5/1/1977 at Texas A&M Research Foundation for Phase I research which consists of a feasibility study for modeling a helicopter system and its mission in combat and noncombat environment. An amount of $5,000 was received 3/1/1978 for a Phase research in order to extend the analysis of Phase I to include cost components and illustrate the use of GRASP to simulate the cost of a helicopter flight. Results of Phase I are documented in the report "Determination of the Applicability of GRASP to Model Reliability and Availability Characteristics of Aircraft Weapon Systems," and results of the Phase II are in the report "Cost Analysis of Aircraft Weapon Systems."

As a result of successful efforts in Phase I and II, the need for a complete documentation manual to be used at the US Army Research and Technology Laboratories became apparent. A $28,340 amount was received 2/7/1978 and a preliminary copy of the manual was delivered in January, 1980. To increase the

modeling capability of GRASP, features related to the nodal release mechanism and portions of the data input were significnatly changed by Mr. Belkacem Manseur. Another amount of $20,428 was received 4/17/1980 for more developmental work and extensions. A case study on the application of GRASP III to model the mission effectiveness of the RPV system has been completed, and this manual documents the final version of the program. We wish to thank Mr. Bradley J. Morris who worked with Mr. William L. Andre during the summer 1980, for his corrections and suggestions to improve the earlier version. We also wish to express our sincere appreciation to the National Aeronautics and Space Administration and the US Army Research and Technology Laboratories for sponsorship of this work.

TABLE OF CONTENTS

SECTION 1

INTRODUCTION: PHILOSOPHY, ORGANIZATION, AND RATIONALE

For simple operating systems, reliability analysis can be performed without resorting to simulation or computer programs. Even with a large number of components, if each component has a specific probability of failing during a given mission time, and if all are operating in parallel, the product of individual failure probabilities is the probability of failure of the system (provided that the components are independent). However, for many systems there are computational complicating factors.

Failed units may be repaired and returned to service. Units may be held in reserve and placed into service only when needed. Further, the automatic or manual switching mechanisms which accomplish this may themselves be subject to failure. Systems may fail, be repaired, and fail again according to different failure and repair distributions that are determined by the configuration and by the failure repair characteristics of the component units.

In addition to conventional factors, systems characterized by costs associated with operating and repairing component units may make the definition of system reliability a matter of choice. Reliability may be related to the probability of completing a given mission time, the expected number of failures, expected total down time, expected operating or repair cost, expected down time, cost for a single failure, or the probability of exceeding a threshold level of accumulated costs due to down time. These alternate definitions of system failure create complex analysis for which analytic solutions are available only for simple, special cases. the GRASP methodology is a computer simulation approach for solving all classes of problems in which both failure and repair events are modeled according to the probability laws of the individual components of the system.

The simulation program can handle all of the common probability distributions, and also accommodate a limited number of user specified distributions. The importance of this capability appears in the decomposition of very large systems. Distributions of separate portions of the system are first obtained in a histogram form, and then combined to represent the performance of the entire system. Data input is also minimized by utilizing schemes for replicating identical portions of the system.

The data input consists essentially of the following:

- life time distribution for each of the components
- repair time distribution for repairable components
- switching or delay time distributions, and switch reliability, for standby redundant components when applicable
- cost of affecting repair if cost analysis is desired.

The results of a simulation upon which the comparison of alternative designs may be based, include:

- the usual descriptive statistics of the time to failure and the time to repair for the entire system or any portion of it
- a count of where failures occur. This provides a measure of the relative importance of the components and indicates where redundancy is or is not needed.
- system downtime accumulation and its variability on repeated runs
- accumulated or "one time" repair costs when applicable
- histograms of the values of random variables that were generated. These may be used as an input for a user specified distribution in further analysis.

From the above information, system reliability and availability can readily be calculated.

GRASP is an extension of the GERTS-III Z simulation program developed by A.A.B. Pritsker [12]. GERTS-III Z is a generalized simulator for processes of the semi-Markov type. GRASP incorporates all of the features of GERTS-III Z; indeed with only minor changes any GERTS-III Z data deck can be correctly processed by the GRASP program. In addition, GRASP contains many features which make it particularly suitable for simulating reliability systems. The modifications of GERTS-III Z were programmed by Joseph Polito and Edward R. Shulaker [22] under the sponsorship of National Science Foundation Grant ATA 73-07822 A01, directed by Professor Don T. Phillips of Texas A&M University. Significant extensions that increase the modeling capability of GRASP were added at Texas A&M University without affecting the previous characteristics of the program, under NASA Grant NSG-2226.

The modeling philosophy of GRASP is that reliability configurations can be represented as network diagrams through which the parameters and logical relationships are specified. This network is then easily converted into a set of data cards to be processed by the GRASP program. No programming is required of the user.

GRASP notation will be presented which is an extension of the block diagrams that are so familiar in reliability work. These diagrams are very useful for representing a complex reliability system, and the preparation of equivalent GRASP networks from them is relatively straightforward. The state transition diagrams that are often used in Markov models are highly recommended in addition to the block diagram representations for the purpose of inspiring the construction of the equivalent GRASP network. These concepts will be discussed in subsequent sections.

GRASP is a collection of ANSI FORTRAN subprograms. Hence, the system is easily portable between computer installations. The user is required only to have a FORTRAN compiler, a line printer, and (for some options) disk storage.

In summary, GRASP is a network based, FORTRAN simulation language with special features for analyzing the reliability of complex systems. It is highly portable, requires no programming by the user, and permits a wide variety of complex configurations to be analyzed.

SECTION 2

GRASP MODELING

GRASP is a digital simulation program based upon the methodologies and structure network modeling techniques. GRASP generalizes fundamental network concepts and its basic intent is to provide a framework through which complex operational systems exhibiting reliability, availability and maintainability (RAM) characteristics can be analyzed. GRASP modeling characteristics also make it particularly suitable as a risk analysis and decision analysis tool. However, throughout this manual emphasis is placed on the modeling philosophy as it relates to RAM systems. Basically, the GRASP approach consists of four steps:

- decompose the system into its principal elements
- describe and analyze the characteristics of the elements and build an operational reliability block diagram of the system
- build a GRASP network model from the block diagram
- validate and verify the GRASP model.

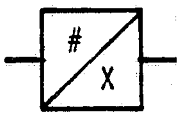
This section describes the basic modeling concepts and shows how system performance can be accessed through statistical collection procedures.

2.1 Reliability Block Diagram Analysis

In order to facilitate the construction of GRASP Networks and to aid in communication, it is helpful to include this information on a GRASP Reliability Block Diagram as a preliminary step to constructing the GRASP network.

The basic component in a GRASP Reliability Block Diagram is a box (Figure 2.1.1). Two pieces of information are contained in each box. The first is the identity number of the component. For instance, if there are ten components in the system, they might be numbered 1,2,3,...,10. The second piece of information is the failure characteristic (denoted by x in Figure 2.1.1). In simple reliability analysis, the component may be considered as having a probability of failure. For this case, the failure characteristic is the failure probability.

In more complex analysis, the component is assumed to have a distribution for time between failures (TBF). If this is the case, denote this characteristic by inserting an F in the lower quadrant. If, in addition, the component may also be repaired, use FR for the failure characteristic. (The identification of the distributions for TBF and TTR, required time to repair, will be discussed in section 4.3 of this report.)



	#	- component number
If:	$0 < X < 1.0$	- probability of failure of component
	X = F or blank	- component has a TBF distribution (no repair)
	X = FR or R	- component has TBF and repair distributions

Figure 2.1.1 Basic Reliability Block

It is necessary to distinguish between a schematic physical diagram and the reliability block diagram. For example, Figure 2.1.2 shows reliability block diagrams that are also schematic physical diagrams.

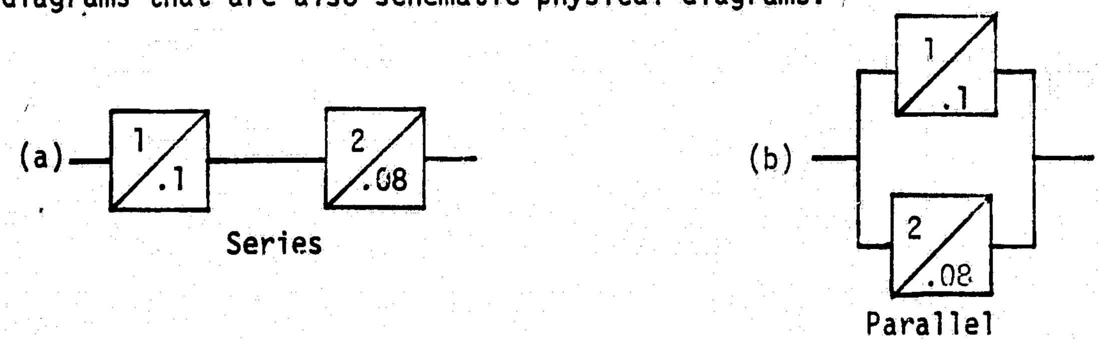


Figure 2.1.2 Arrangement of Switches to Establish Flow of Current

Figure 2.1.2 represents a series (a) and a parallel (b) configuration of two parts with known failure probabilities (.1 and .08 respectively). Assuming that the failures are independent and using the well known probability relations: $Pr(A \cap B) = Pr(A) \times Pr(B)$ and $Pr(A \cup B) = Pr(A) + Pr(B) - Pr(A \cap B)$, then the probability of system failure is $.1 + .08 - (.1)\times(.08) = .172$ in a), and $(.1)\times(.08) = .008$ in b).

It is important to identify the cases where the reliability block diagram is not the same as the physical block diagram. This distinction can be made by considering that the schematic physical diagram corresponds to the physical configuration of the system, while the reliability block diagram shows the functional relationships among system elements. For example, a pipeline with two valves can be physically represented as in Figure 2.1.3 . Its proper

Figure 2.1.3 Schematic Physical Diagram of a pipeline with two valves

reliability block diagram depends upon the definition of system performance. If the two valves are normally open, but are expected to shut on command to stop flow, it is functionally a parallel system because the performance is adequate (i.e. the flow is stopped) if at least one valve works. If they are normally shut but are expected to open on command to provide flow, it is essentially a series system. The reliability block diagram may depend also on the nature of the failure. For instance the cylinders of a helicopter engine can be considered to be either in parallel or in series. If a plug fails, the engine will continue functioning with the remaining cylinders, and this action is similar to parallel operation. However if one of the cylinder connecting rods fails or a piston sticks, all cylinders stop functioning, which corresponds to a series operation.

Figure 2.1.4 shows other simple reliability block diagrams illustrating this notation. Figure 2.1.4 (a) depicts two units in series without repair.

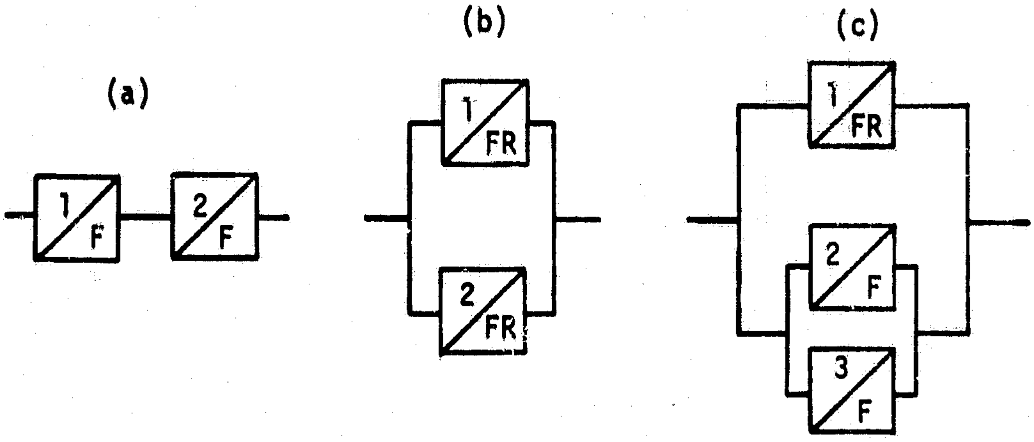


Figure 2.1.4 Other Simple Reliability Block Diagrams

In Figure 2.1.4 (b) each of the components in parallel has a failure distribution as in 2.1.4 (a). Further, these units can also be repaired. Hence, the system fails only when an online unit fails while the other is being repaired. Figure 2.1.4 (c) shows a repairable unit in parallel with two non-repairable units which are themselves in parallel.

In some systems, components are held in standby and are brought on line only if the primary unit fails. Figure 2.1.5 displays the notation for these situations. The block diagram in Figure 2.1.5 indicates that unit 5 is a standby for unit 1.

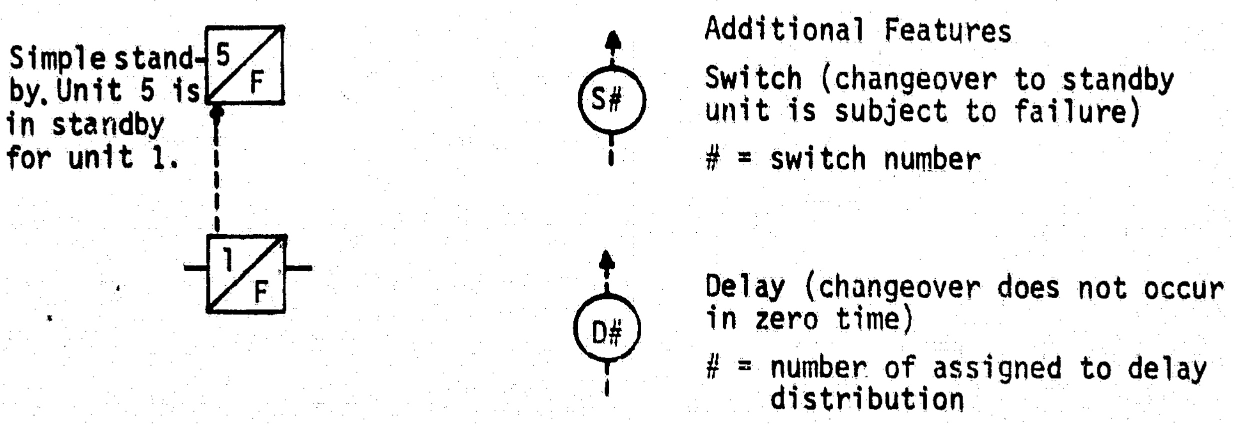


Figure 2.1.5 Notation for Standby Components

Note that Figure 2.1.5 does not imply that unit 1 is a standby for unit 5. The physical interpretation is as follows. Unit one will be used until it fails. When it does fail, Unit 5 will replace it (in zero time) and will remain in the system until it also fails. The system will fail when Unit 5 fails.

More complex situations can be modeled using the additional notation also found in Figure 2.1.5. If the switching function which causes the replacement of the units is subject to failure or there is a probability that the standby part is defective, this can be shown by a circle containing the letter "S" followed by the number of the switch. Also, if there is a delay associated with making the change, a circle with the letter "D" followed by the number of the delay is used. Examples are shown in Figure 2.1.6.

Unit 2 in Figure 2.1.6 (a) has an interchangeable unit in standby. If Unit 2 fails, it is replaced immediately by Unit 1. If Unit 1 fails before Unit 2 is repaired, the system fails. Otherwise, unit 2 is repaired, but it remains in standby until Unit 1 fails and the cycle is repeated. A similar situation is shown in 2.1.6 (b),but in this case the switching mechanism (perhaps an automatic operation) denoted by S1 has a designated probability of failure when Unit 7 replaces Unit 6. In the reverse operation, a delay D1 is incurred (perhaps by a manual operation).

Figure 2.1.6 (c) adds an additional complication. Both switching failure and delays are shown. The order in which they are presented represents the logical relation between them. When Unit 8 replaces Unit 10, the switching function occurs after which a delay (perhaps a warm-up) takes place. In the reverse operation the delay occurs first (operator reaction time), then the switching failure may occur. Figure 2.1.6 (d)

ORIGINAL PAGE IS
OF POOR QUALITY

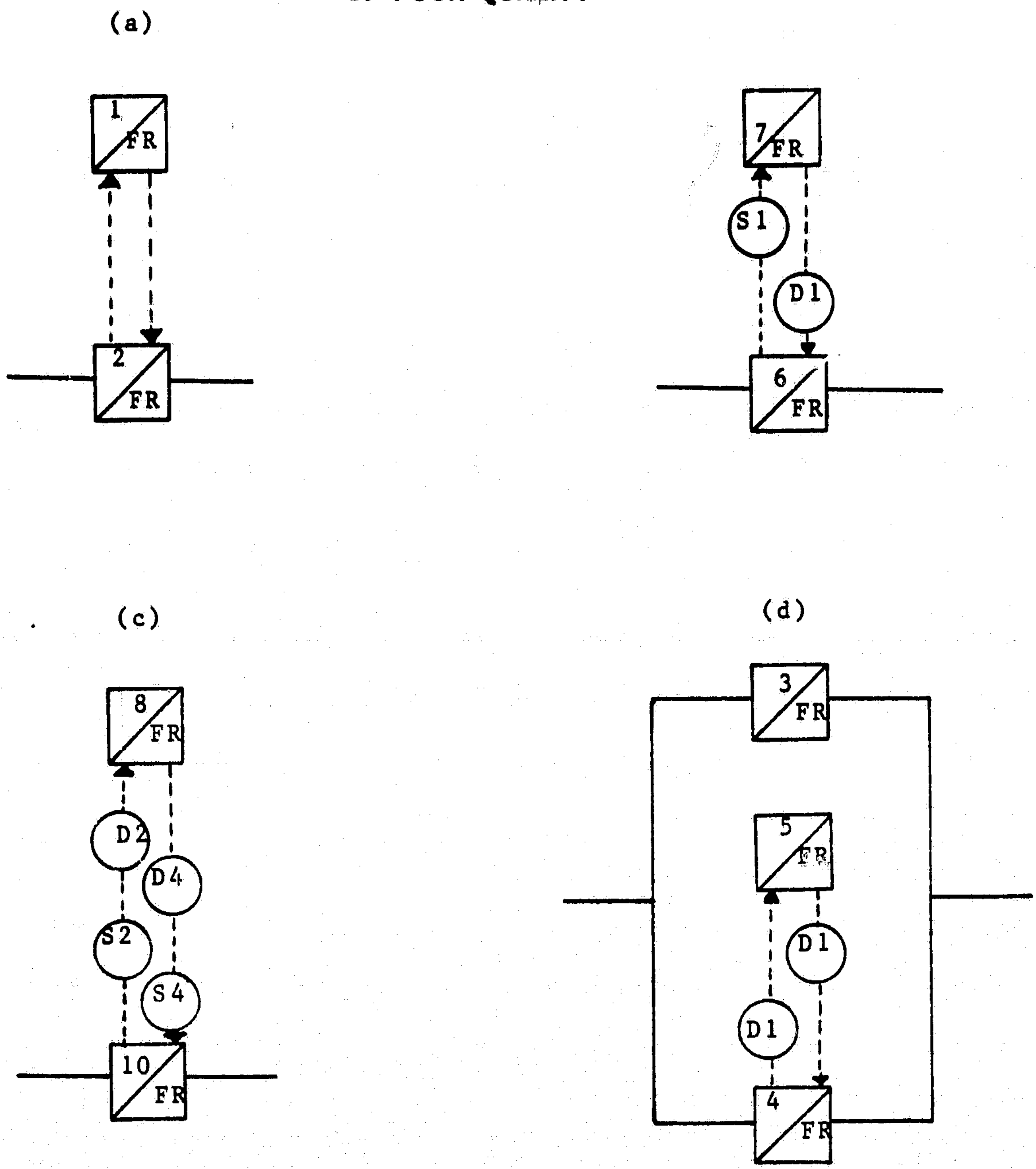


Figure 2.1.6 Examples of standby components

represents a case where a standby unit exists for one of the branches of a parallel system. Units 4 and 5 replace one another when failures occur. A time delay of D1 is incurred each time a replacement occurs.

A Specification Table is included with the GRASP Block Diagram to list time to failure, repair time, delay time, and switch repair distributions. An example of a Specification Table is shown in Figure 2.1.7. This Specification Table would accompany the block diagram in Figure 2.1.6 (c). Each unit with a time to failure distribution (i.e., a failure characteristic that is either an "F" or an "FR") is listed in the first part of the table. The type of time to failure distribution is given in the second column. Subsequent columns contain the mean, standard deviation, and user selected minimum and maximum failure times. The last column is used to include additional information. For example, the time to failure for unit 10 is given by a Weibull distribution. The Weibull distribution is described by parameters A and B which must be input to the program, so they should be included in the Specification Table.

The second section describes the repair time distributions. Time delays are next input in a similar manner. A unique feature of GRASP is the ability to analyze system costs along with system time characteristics. If there are costs associated with repairs or delays, these can be indicated in the last column. The analysis of system costs will be described in a later section. The switches are listed after the delays. The distribution information pertains to the time that it takes to repair the switch. The probability of failure of the switch is given in the "Other Information" column.

The GRASP Block Diagram and Specification Table are useful in that they can be prepared by someone who is not familiar with the GRASP program or modeling philosophy. Also, the mechanics of preparation are simple and easy

ORIGINAL PAGE IS
OF POOR QUALITY

SPECIFICATION TABLE				
Unit	TBF Distribution	Mean	Min	Other Information
		St.Dev.	Max	
8	Normal	100	0	
		20	200	
10	Weibull	1.0	0	
		0.4	10	A = 0.73 B = 2.64
Unit	Repair Distribution	Mean	Min	Other Information
		St.Dev.	Max	
8	Exp	5	0	
		5	50	
10	Erlang-2	0.05	0	
		-	20	
Delay	Distribution	Mean	Min	Other Information
		St.Dev.	Max	
D2	Exp	0.01	0	
		0.01	10	
D4	Gamma	0.01	0	
		0.02	10	

Figure 2.1.7 Sample Specification Table for Figure 2.1.6(c)

to learn. Thus, the GRASP Block Diagram provides a tool to bridge the communication gap between the GRASP modeler and the user/operator of the system. Once the user has learned the concepts of the Block Diagram notation, one can, with the help of the modeler, arrive at a final Block Diagram that suitably describes the system. The Specification Table indicates what information is needed by the program and will direct the modeler in data specification. The modeler can prepare the GRASP Network from the Block Diagram.

2.2 GRASP Methodology

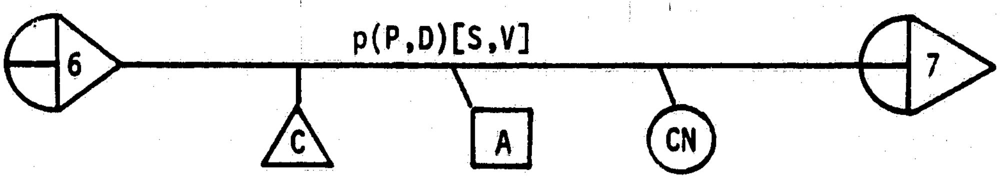
As a graphical technique, GRASP employs an activity on arc philosophy and it has two basic elements:

- nodes represent events in the system such as a failure of a component, an end of repair, or a completion of any other type of activity
- arcs or branches represent time consuming activities (such as time to failure, time to repair), or any event precedance relationship in the system. In this case, the arc has no duration (it is a dummy activity).

Each element carries information which is used to indicate the collection of statistical data in the simulation procedure. This information is normally recorded directly on network symbolism using symbolism concepts that will now be described.

2.2.1 Arc Symbolism

Each arc is associated with a particular activity. Parameters can be written on the arc to uniquely specify operational characteristics of an activity. A general description is shown in Figure 2.2.1.



where

p - probability of the activity being performed, given that the node 6 is released

D - distribution type - specifies probability distribution for time of the activity

P - parameter set number for use with a particular failure or time density function

S - setup cost of activity - this is a fixed cost (optional)

V - variable cost of activity - this is a time dependent cost (optional)

C - activity count type - this is used to record event usage rates

A - activity type - this is for identification purposes (optional)

CN - accumulator node associated with this arc - this is used in accumulating activity costs (optional)

Figure 2.2.1 Arc Symbolism

2.2.1.1 Description of the Arc Symbolism

The probability, p, associated with each arc is used by the output side of the originating node. For deterministic nodes, $p = 1.0$. For probabilistic nodes, $p < 1.0$. If p is not shown, it is assumed equal to one.

The distribution type, D, is a number which specifies the probability distribution which is to be sampled to determine the duration of the activity. The available distribution types and their numbers are:

1. Constant
2. Normal
3. Uniform
4. Erlang
5. Lognormal
6. Poisson
7. Beta
8. Gamma
9. Beta fitted to three parameters as in PERT
10. Constant (scaled)
11. Triangular
12. Weibull
13. Empirical (user specified)

The parameter set number, P, identifies a set of up to four numbers that is specified by the user to uniquely establish the distribution. For example, if D = 2 (Normal distribution), the parameter set will specify the mean, standard deviation, minimum and maximum values for the normal distribution desired. Parameter sets are defined in Table 4.3.2.

Distribution types 10 and 13 are a slight exception to this rule. A <u>network scale factor</u> is defined on initial input and whenever D = 10, the value used for the activity duration is P divided by the scale factor. In other words, if the scale factor is 10 and a constant time of 0.4 is desired on an arc, then P should be set to 4. GRASP will use 4/10 as the constant time. This effectively converts integer constants to real numbers when using D = 10.

Empirical distributions that are determined from the histograms may be inputted and are numbered by the user. If D = 13 is specified, P is equal to the number of the desired empirical distribution. For example, if two empirical distributions are provided by the user and (1,13) appears on an arc, GRASP will sample from the first of the distributions.

Each activity may be assigned a linear cost function consisting of a fixed and variable cost. Cost information then will be maintained on the operation of the network. The fixed cost is incurred each time that the activity is initiated. The variable cost is accumulated at the time the activity is ended or terminated and is equal to V times the duration of the activity. The total cost of the activity is S + VT where T is the accumulated activity duration. T is an accumulation of sampled values from distribution type D with parameter set P.

Three other parameters can be associated with each arc. The first of these is the count type, C. Any arc may be assigned a count type. The count type maintains a count of the number of times the activity is completed. Hence, each time the activity terminates normally (not forced to halt) the count type for the arc is incremented by one. The reason that it is called a count type is that more than one arc in the network may have the same count type. Therefore, the count type represents the number of times that all arcs with that type have been completed. There may be more than one count type in the network. The count types are indicated by putting the number of the count type in a triangle below the arc. The small line connecting the triangle to the arc may be omitted if desired.

In addition to the count type, each arc may be assigned an activity type A. Usually it is convenient to use an activity type that is nonnegative for failure activities and negative for repair activities. When it is not specified, it is assumed to be equal to 1, and more than one arc may have the same activity type. Activity types play an important role in the release mechanism of a node and in network modification. These concepts will be discussed in a later section.

2.2.2 Node Symbolism

A distinction has been made between failure activities (nonnegative activity types) and repair activities (negative activity types). We can

ORIGINAL PAGE IS
OF POOR QUALITY

similarly define two general classes of nodes:

- failure nodes: corresponding usually to failure events in the system, and
- repair nodes: corresponding usually to end of repair events.

Each node type has the same fundamental characteristics and differ only in the sign of their release counter. Such a counter indicates the number and type of activity completions (failures or repairs) that are needed in order to release the node. It is positive for failure nodes and negative for repair nodes. The use of failure and repair nodes will be illustrated in later sections.

Nodes are characterized by an input side and an output side. The input side establishes the conditions which must be satisfied by activities or arcs which terminate at the node before any activities will be scheduled from the output side. Once these input requirements have been satisfied, the output side determines how the activities leaving the node will be routed through the network.

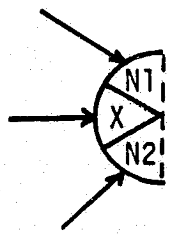


Figure 2.2.2 General Input Side of a Node

Figure 2.2.2 shows a general input side with three pieces of information:

N1 = initial release counter. The absolute value of N1 indicates the number of incoming activity completions of type 'sign (N1)' for first release of the node. If N1 is negative, then |N1| repair activities are needed for the release of the node. If N1 is positive, then nonnegative type activities only will contribute to the release of the node

N2 = release counter for subsequent releases of the node. It has the same working mechanism as N1.

X = Response descriptor. It can be one of the following:

A - the N1 or N2 condition must be satisfied by distinct incoming arcs

H - Halt capability, when the N1 or N2 condition is satisfied stop all activities which are scheduled to end at the node

U - A and H are combined

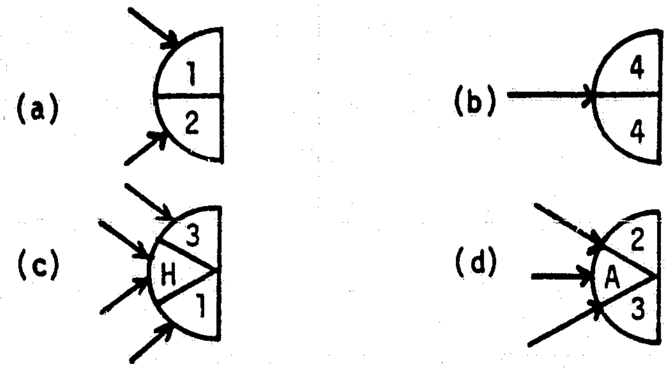


Figure 2.2.3 Example of Input Sides

Figure 2.2.3 shows some examples of input sides of nodes with positive release counter and nonnegative type of incoming activities. Examples with negative release counters and negative activities will be discussed in the release mechanism section.

In Figure 2.2.3 (a) N1 = 1, and N2 = 2 (the response descriptor is omitted when it is blank). There are two arcs which terminate at this node. When the simulation is begun, either of the activities on these arcs can cause the node to be "released". A node is released when the input condition (N1 or N2) is satisfied. A release node initiates the activities that emanate from it, and immediately resets itself to be released when the input condition is again satisfied.

For example, suppose that when the simulation is begun the activities on the arcs are scheduled to end at times of 10 and 20. At time 10, one of the activities will end. Since N1 = 1, the input condition is satisfied at time 10 and the node is released. Once released, all activities which begin at that node can be scheduled to start. It is immediately reset, but now the required condition (N2) is 2 . The terminology is that the "release count" is 2. In other words, two activities must complete at this node before it will be released again. At time 20 when the second activity ends, one more activity is required to release the node a second time.

The current value of the release count is a current indicator of how many more activities need to be completed in order to release the node. When the release count is zero, the node is released. So, at time zero the release count is equal to N1 = 1. After the first release, the release count is set to N2 = 2. After time 20, the release count will be one until a third activity is completed, at which time the node will be released and the release count reset to N2 = 2.

Figure 2.2.3 (b) is similar to (a), but now N1 = N2 = 4. Note that there is only one arc into the node. This means that the same activity must be completed four times in order to release the node once. This example demonstrates that one activity may be performed many times during a simulation and that the input condition may not be directly related to the number of arcs incident to the node.

Figure 2.2.3 (c) has a node with N1 = 3, N2 = 1, and release descriptor = H. Note that there are four incoming arcs. Any three activity completions will release the node (e.g., one from each of three separate arcs; one from one arc and two from another; or 3 from a single arc). When the node is released, the release count is reset (to one) and any incoming activities which have not yet completed are halted (due to the H descriptor). For example, if the node is released at time TNOW, and there are two activities scheduled to end at the node at times greater than TNOW, they will be cancelled (in other words, never allow to complete). Another way to view this option is that after a node is released, only activities which are initiated after the release time can count toward the next release. This is the function of the H descriptor.

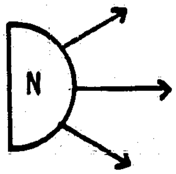
Figure 2.2.3 (d) has response descriptor A. This implies that the activities which will decrement the release count must come from distinct arcs. The node will use only the first completion of each arc in decrementing the release count. Multiple activity completions can occur on any arc, but only the first one will decrement the release count. Hence, the number of incoming arcs must equal or exceed N1 (N2).

No example of a node with response descriptor U is shown. A U indicates both the A and H capabilities are in effect. A further generalization of the release count mechanism will be discussed in the section on node release mechanism.

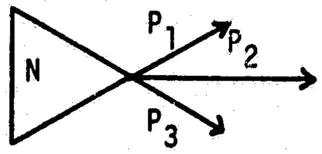
The output side of a node determines how activities are scheduled when the node is released, and in addition, contains the node number. There are two types of output sides: deterministic and probabilistic.

When a deterministic node is released, activities are scheduled on all arcs leaving the node. For probabilistic nodes, only one activity is scheduled. Each arc leaving the probabilistic node has a probability assigned to it. When a probabilistic node is released, one of the arcs is selected according to the relative probabilities of the arcs.

Examples of output sides are shown in Figure 2.2.4. Graphically, deterministic nodes are rounded on the output side, and probabilistic nodes are pointed. Figure 2.2.4 (a) shows Node 2 (a deterministic node) with three activities leaving it. Each time Node 2 is released, an activity will be scheduled for each of these arcs. Node 62 in Figure 2.2.4 (b) is a probabilistic node with three leaving arcs. The arcs are assigned probabilities 0.1, 0.6, and 0.3. Whenever Node 62 is released, the node will select only one of the arcs according to the given probability distribution and schedule an activity on it.



deterministic node

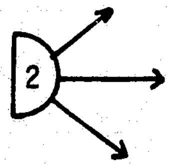


probabilistic node (It is required that $P_1 + P_2 + P_3 = 1$.)

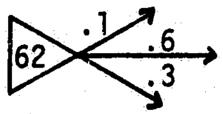
N - node number

NOTE: all node numbers must be greater than or equal to 2

(a)



(b)



(Notice that .1 + .6 + .3 = 1. as required)

Figure 2.2.4 Output Sides of Nodes

Figure 2.2.5 has several examples of GRASP nodes. Note that the input and output sides may be specified in any combination.

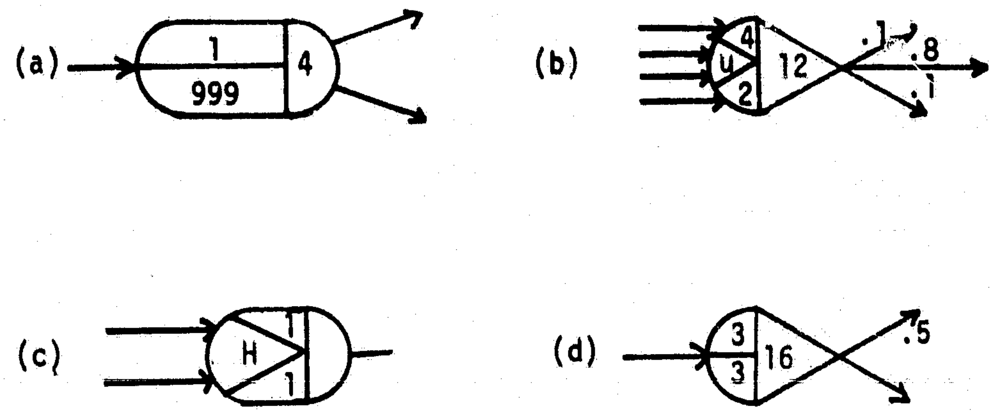


Figure 2.2.5 Examples of Nodes

Further, note that the node in Figure 2.2.5 (c) has no arcs leaving. This is permissible. Also, in Figure 2.2.5 (a), N2 = 999. GRASP approximates infinity with 999. The meaning here is that the node should be released only once. Note, however, if 999 activities complete at Node 9, it will be released again. If a better approximation to infinity is needed, we can put two such nodes in series or increase this number.

Two special types of nodes are called source and sink nodes and are used to initiate and terminate the flow of transactions, respectively. Figure 2.2.6 has examples of source and sink nodes. Graphically, source nodes are usually <u>preceded</u> by a single "wavy" line and sink nodes <u>succeeded</u> by a single "wavy" line.

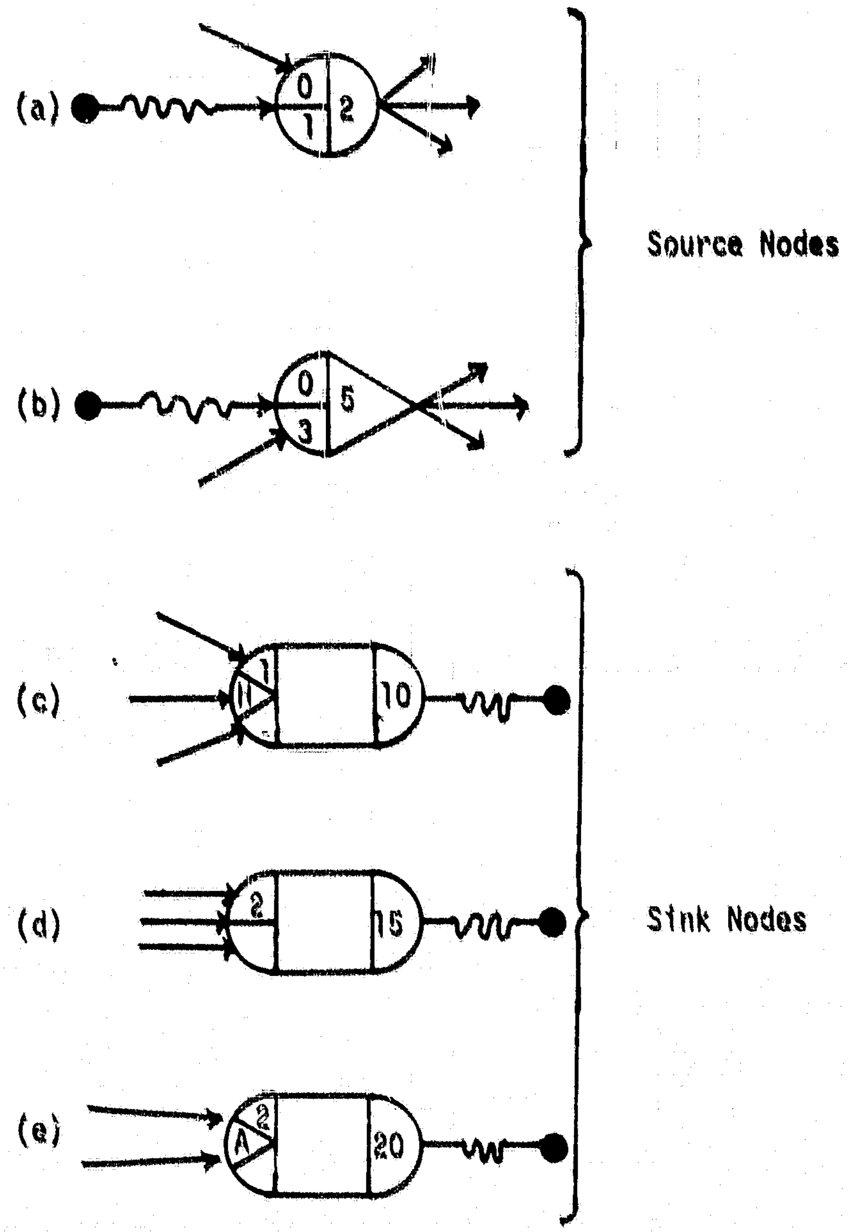


Figure 2.2.6 Source and Sink Nodes

Referring to Figure 2.2.6, nodes 2 and 5 are source nodes. At time zero, all source nodes (there may be more than one) are released by GRASP. This causes activities to be scheduled according to the output sides of the source nodes. Hence, the value of N1 is not relevant for source nodes. N2 can be relevant, however, as is indicated in Node 5. Node 5 has arcs incident to it, and it is possible that it could be again released later in the simulation if the N2 criterion is satisfied. The rule is that source nodes behave exactly like other nodes, except that they are automatically released at time zero. (For future reference, please note that source nodes may not be Mark nodes or Statistics nodes.)

Nodes 10, 15 and 20 are sink nodes and they are used to terminate the simulation run. As with source nodes, there may be more than one sink node in the network. (See variable NSNK in data card NEW-1 p. 150). Several simulation runs may be needed to end the simulation run; this is specified in the variable NRNS in data card NEW-1. For example, if segments c), d) and e) are part of the same network, then there are three end of run possibilities.

The space between the input and output sides of nodes 10, 15 and 20 is for specification of the type of Statistics to be collected at these nodes. This will be explained in the section on Statistics Collection pp. 63-67. For sink nodes, First (F) Statistics will be automatically maintained if nothing is specified.

2.2.3 Node Release Mechanism

Recall that for each node two release counters have been defined : a counter for the first release of the node with initial value N1 and a counter for its subsequent releases with initial value N2. N1 and N2 may be positive or negative integers between -99 and 999 specified in the input card NEW-2 (see p. 117). Since their working mechanism is the same, the initial value of a release counter, N1 or N2, will be referred to as N, and its current value will be referred to as M.

Refering to Figure 2.2.1 p. 14, an arc can have a number associated with it, denoted A and called its Activity type or Arc type. A can be any integer number between -99 and 999. Whenever an incoming nonnegative type arc (that is an arc with A greater than zero or with A unspecified) is completed, GRASP subtracts one from the current value M of the release counter. But if the release counter is negative and equal to its initial value ($M=N<0$), it cannot be decremented below that value and remains unchanged. When an incoming arc with type A less than zero is completed, GRASP subtracts A from the current value of the release counter. In this case M is replaced by M-A which means that the release counter is incremented by the absolute value of A. If the release counter is positive it cannot be incremented beyond its initial value N.

As incoming arcs of different types are completed, the release counter is decremented or incremented depending upon its sign and magnitude. If N is greater than zero then the node is released when M becomes less than or equal to zero. If N is negative then the node is released when M becomes greater than or equal to zero. In most cases this means that a node is released when the current value of its release counter M becomes zero. After a node is released , its release counter is automatically reset to its initial value N2. Figure 2.2.7 summarizes the release mechanism of a node.

Activity type	M = N	0 < M < N
$A \geq 0$	$M - 1$	$M - 1$
$A < 0$	no change	$M - A$ if $M - A \leq N$ N if $M - A > N$

(a) Failure Node ($N > 0$)

Activity type	M = N	N < M < 0
$A \geq 0$	no change	$M - 1$
$A < 0$	$M - A$	$M - A$

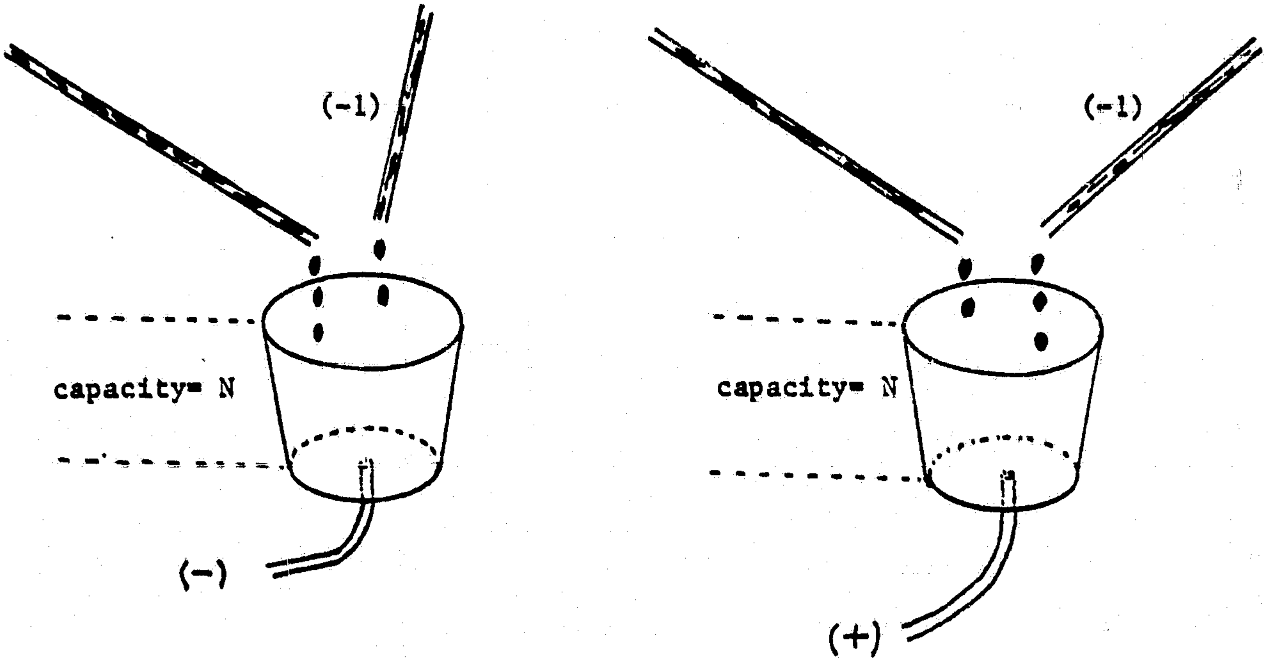
(b) Repair Node ($N < 0$)

Figure 2.2.7 Node Release Mechanism

This node release mechanism, while it may seem obscure at this point, will prove to be a powerful tool in modeling repairable systems. If we think of the system as being characterized by total failures and/or repairs, then the release of a node indicates when system equilibrium is disrupted. At a failure node, the initial value N of the release counter indicates how many components must be simultaneously down in order for the system to go down. As long as repairs counterbalance the number of failures, the system would remain in equilibrium and the node will not be released. Similarly, when the system is down the release counter at a repair node indicates how many components need to be repaired in order for the system to operate, and an equilibrium between failures and repairs in this case indicates that the system is still down. Positive and negative incoming activity numbers can be used to monitor system behavior.

Another intuitive approach in interpreting the node release mechanism is to consider the two types of activities (positive and negative) as two different types of flows through the network. This flow interpretation may help understand how this mechanism works.

A non-negative type arc carries one unit of positive flow (or one positive pulse) everytime it is scheduled. On the other hand, an arc with a negative type equal to A ($A < 0$) carries ($-A$) units of negative flow (or negative pulses). These flows have a travel time from the start to the end of the arc which is the sampled time of the arc. If an arc has no duration (such an arc represents a dummy activity), then it is assumed that the flow traverses the arc instantaneously. A node can be pictured as a bucket with a holding capacity equal to the initial value N of the release counter. (N can be either N1 or N2). A failure node is a node whose release indicates a system failure. It is such that the initial value of its release counter (N1 or N2) is positive, and it holds only positive flow. A repair node or a node whose release indicates an end of repair is such that the initial value of N1 or N2 is negative and it holds only negative flow. Whenever quantities of positive and negative flows are put together at a node, the resulting flow is given by the arithmetic operation between these quantities. Figure 2.2.8 shows a simplistic picture of this flow interpretation.



(a) Failure node.
holds positive flow only.

(b) Repair node.
holds negative flow only.

Figure 2.2.8 Network Flow Interpretation of the Node Release Mechanism

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 2.2.8 (a) shows a failure node, or a node monitoring component failures. We can imagine a selective hole at the bottom of the bucket that lets only negative flow units through if they ever reach the bottom. A negative unit and a positive unit cancel each other. The repair node in Figure 2.2.8 (b) works the same way and lets only positive flow out. In either case, the bucket is initially empty and when it gets full (nodal release) we empty it and start over again.

Let us now look at the examples in Figure 2.2.9.

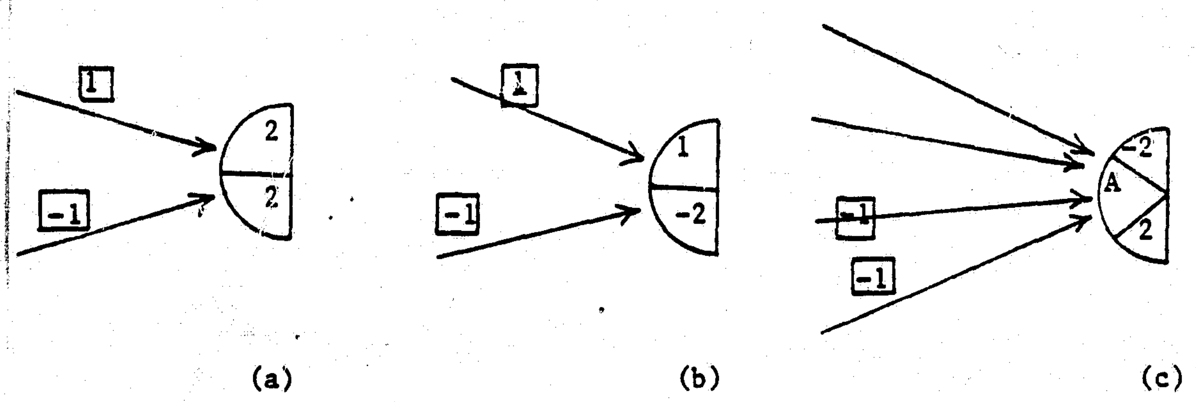


Figure 2.2.9 Examples of input side of nodes

Figure 2.2.9 (a) shows a failure node with two types of incoming arcs. Assume that activity type 1 is completed at times 1, 3, 5, and 6 and activity type -1 is completed at times 2, 4, 7, and 8. Let us observe how the current value of the release counter, M, changes over time.

consideration since the simulation run stops when the sink node is released. Node release counters are initialized to their corresponding N1 values at the beginning of each run. The examples in Figure 2.2.10 illustrate further the node release mechanism. Computer results for two simulation runs are given in Figures 2.2.11-15. They show that for a source node, N1 is irrelevant and for a sink node, N2 is irrelevant. Notice that for the example #5, the simulation is abnormally terminated; can you see why?

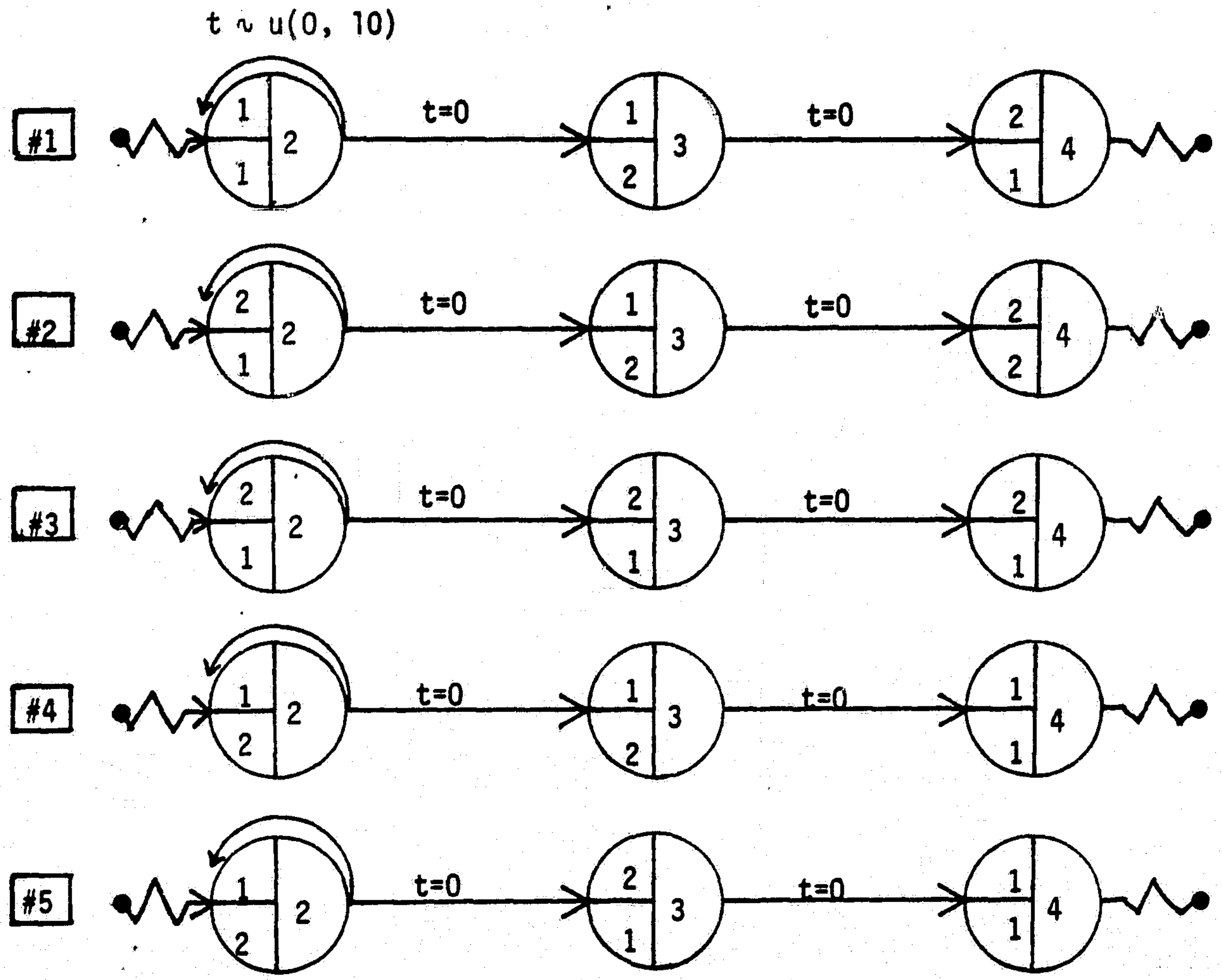


Figure 2.2.10 Examples of Node Release

The node is released at time 6 and its release counter is immediately reset to 2. At time 7, completion of activity type (-1) does not decrement the release counter. In line (b) activity completions are the same as in Line (a). Note that the node is released for the first time at 1 and at time 8 a second time.

Time	0	1	2	3	4	5	6	7	8
Completion of activity type (1)		x		x		x	x		
Completion of activity type (-1)			x		x			x	x
Current value of release counter (a)	2 →	1 →	2 →	1 →	2 →	1 →	0:2 →	2 →	2
Current value of release counter for (b)	1 →	0:-2 →	-1 →	-2 →	-1 →	-2 →	-2 →	-1 →	0:-2

Any release descriptors can be combined with negative type activities and negative node release counters. The response descriptor applies to both types of activities. The node shown in Figure 2.2.9 (c) will be released for the first time when both negative type activities are completed, and these arcs are not counterbalanced by the completion of other activities. For subsequent releases nonnegative type activity completions are required. Source, sink and statistics nodes may have N1 and/or N2 negative. However, for a source node the value of N1 does not play any role and it is usually taken to be zero. Similarly, for a sink node, the value of N2 is not taken into

A run corresponds to one complete independent simulation of a network model. Two simulation runs are independent in the sense that they use two independent sequences of random numbers. The user specifies the starting seed for random number generation for the first run and the program uses the random number that was sampled last at the completion of the first run, as a starting seed for the next run. The final results (see computer output) summarize the statistics accumulated over all the runs. A simulation run can normally be terminated in two ways:

- a sink node has been released (that is the case in examples 1 through 4) or,
- the T2 or C2 criterion of a C-node has been satisfied (this is explained in the next section).

It can also be abnormally terminated in two ways:

- an error has been detected in the model or in the program, in which case an appropriate message is given (that is the case in example 5), or
- the limits on the execution time or the output space specified in the job control language cards (specific to the computer installation) has been violated. An appropriate message is also printed.

Various options for defining a run are used in examples 1 through 5, and the reader should compare them. For each example, the computer output includes an event trace for run 1, an event trace for run 2, and the final results for both runs. The tracing option is included in the GRASP program to facilitate the user in validating his model. Basically, it consists of printing the nodes and arcs affected at each event during certain intervals of the simulation. There are 3 ways the user can request such a tracing:

a) Specify the run number to start tracing, NSTR, and the run number to end tracing, NETR. The program will print node and arc characteristics at each event during the specified simulation runs.

b) Specify the time to begin tracing, TSTR, and the time to end tracing, TETR. The program will print the same information as in case a) at each event between TSTR and TETR, during all the simulation runs.

c) Specify NSTR, NETR, TSTR and TETR. The program will trace whatever happens between TSTR and TETR for the specified runs.

The program expects an input that NETR $\geq$ NSTR and TETR $\geq$ TSTR. An error message is printed otherwise. The tracing option can be summarized as follows:

TSTR	NSTR	Type of Tracing
0.	0	no tracing
0	> 0	a)
> 0.	0	b)
> 0.	> 0	c)

** EVENT TRACE FOR RUN 1 **

END OF ARC TIME	ARC NUMBER	END NODE	PARAM. TYPE	DISTR. TYPE	COUNT TYPE = (	)	RELEASE COUNT	ARC COST	ARC DURATION
0.0	0	3	0	10	0	0	0	0.0	0.0
0.0	0	4	0	10	0	0	1	0.0	0.0
0.352890E+01	0	2	1	3	0	0	0	0.0	0.352890E+01
0.352890E+01	0	3	0	10	0	0	1	0.0	0.0
0.389861E+01	0	2	1	3	0	0	0	0.0	0.369710E+00
0.389861E+01	0	3	0	10	0	0	0	0.0	0.0
0.389861E+01	0	4	0	10	0	0	0	0.0	0.0

** EVENT TRACE FOR RUN 2 **

END OF ARC TIME	ARC NUMBER	END NODE	PARAM. TYPE	DISTR. TYPE	COUNT TYPE = (	)	RELEASE COUNT	ARC COST	ARC DURATION
0.0	0	3	0	10	0	0	0	0.0	0.0
0.0	0	4	0	10	0	0	1	0.0	0.0
0.942175E+01	0	2	1	3	0	0	0	0.0	0.942175E+01
0.942175E+01	0	3	0	10	0	0	1	0.0	0.0
0.118286E+02	0	2	1	3	0	0	0	0.0	0.240681E+01
0.118286E+02	0	3	0	10	0	0	0	0.0	0.0
0.118286E+02	0	4	0	10	0	0	0	0.0	0.0

GRASP SIMULATION PROJECT : EXAMPLE ON THE USE OF THE RELEASE COUNTER. (# 1)

FINAL RESULTS FOR 2 SIMULATION RUN(S)

NODE	PROB./ COUNT	MEAN	STAND. DEV.	NO OF OBS.	MIN	MAX	NODE TYPE
4	1.0000	0.786358E+01	0.560732E+01	2.	0.389861E+01	0.118286E+02	F

Figure 2.2.11

Computer Results for Example #1 in Figure 2.2.10

ORIGINAL PAGE IS OF POOR QUALITY

** EVENT TRACE FOR RUN 1 **

END OF ARC TIME	ARC NUMBER	END NODE	PARAM. TYPE	DISTR. TYPE	COUNT TYPE = (	)	RELEASE COUNT	ARC COST	ARC DURATION
0.0	0	3	0	10	0	0	0	0.0	0.0
0.0	0	4	0	10	0	0	1	0.0	0.0
0.352890E+01	0	2	1	3	0	0	0	0.0	0.352890E+01
0.352890E+01	0	3	0	10	0	0	1	0.0	0.0
0.389861E+01	0	2	1	3	0	0	0	0.0	0.369710E+00
0.389861E+01	0	3	0	10	0	0	0	0.0	0.0
0.389861E+01	0	4	0	10	0	0	0	0.0	0.0

** EVENT TRACE FOR RUN 2 **

END OF ARC TIME	ARC NUMBER	END NODE	PARAM. TYPE	DISTR. TYPE	COUNT TYPE = (	)	RELEASE COUNT	ARC COST	ARC DURATION
0.0	0	3	0	10	0	0	0	0.0	0.0
0.0	0	4	0	10	0	0	1	0.0	0.0
0.942175E+01	0	2	1	3	0	0	0	0.0	0.942175E+01
0.942175E+01	0	3	0	10	0	0	1	0.0	0.0
0.118286E+02	0	2	1	3	0	0	0	0.0	0.240681E+01
0.118286E+02	0	3	0	10	0	0	0	0.0	0.0
0.118286E+02	0	4	0	10	0	0	0	0.0	0.0

GRASP SIMULATION PROJECT : EXAMPLE ON THE USE OF THE RELEASE COUNTER. (# 2)

FINAL RESULTS FOR 2 SIMULATION RUN(S)

NODE	PROB./ COUNT	MEAN	STAND. DEV.	NO OF OBS.	MIN	MAX	NODE TYPE
4	1.0000	0.786358E+01	0.560732E+01	2.	0.389861E+01	0.118286E+02	F

Figure 2.2.12

Computer Results for Example #2 in Figure 2.2.10

** EVENT TRACE FOR RUN 1 **

END OF ARC TIME	ARC NUMBER	END NODE	PARAM. TYPE	DISTR. TYPE	COUNT TYPE = (	)	RELEASE COUNT	ARC COST	ARC DURATION
0.0	0	3	0	10	0	0	1	0.0	0.0
0.352890E+01	0	2	1	3	0	0	0	0.0	0.352890E+01
0.352890E+01	0	3	0	10	0	0	0	0.0	0.0
0.352890E+01	0	4	0	10	0	0	1	0.0	0.0
0.389861E+01	0	2	1	3	0	0	0	0.0	0.369710E+00
0.389861E+01	0	3	0	10	0	0	0	0.0	0.0
0.389861E+01	0	4	0	10	0	0	0	0.0	0.0

** EVENT TRACE FOR RUN 2 **

END OF ARC TIME	ARC NUMBER	END NODE	PARAM. TYPE	DISTR. TYPE	COUNT TYPE = (	)	RELEASE COUNT	ARC COST	ARC DURATION
0.0	0	3	0	10	0	0	1	0.0	0.0
0.942175E+01	0	2	1	3	0	0	0	0.0	0.942175E+01
0.942175E+01	0	3	0	10	0	0	0	0.0	0.0
0.942175E+01	0	4	0	10	0	0	1	0.0	0.0
0.118286E+02	0	2	1	3	0	0	0	0.0	0.240681E+01
0.118286E+02	0	3	0	10	0	0	0	0.0	0.0
0.118286E+02	0	4	0	10	0	0	0	0.0	0.0

GRASP SIMULATION PROJECT : EXAMPLE ON THE USE OF THE RELEASE COUNTER. (# 3)

FINAL RESULTS FOR 2 SIMULATION RUN(S)

NODE	PROB./ COUNT	MEAN	STAND. DEV.	NO OF OBS.	MIN	MAX	NODE TYPE
4	1.0000	0.786358E+01	0.560732E+01	2.	0.389861E+01	0.118286E+02	F

Figure 2.2.13

Computer Results for Example #3 in Figure 2.2.10

ORIGINAL PAGE IS
OF POOR QUALITY

** EVENT TRACE FOR RUN 1 **

END OF ARC TIME	ARC NUMBER	END NODE	PARAM. TYPE	DISTR. TYPE	COUNT TYPE =	( )	RELEASE COUNT	ARC COST	ARC DURATION
0.0	0	3	0	10	0	0	0	0.0	0.0
0.0	0	4	0	10	0	0	0	0.0	0.0

** EVENT TRACE FOR RUN 2 **

END OF ARC TIME	ARC NUMBER	END NODE	PARAM. TYPE	DISTR. TYPE	COUNT TYPE =	( )	RELEASE COUNT	ARC COST	ARC DURATION
0.0	0	3	0	10	0	0	0	0.0	0.0
0.0	0	4	0	10	0	0	0	0.0	0.0

GERT SIMULATION PROJECT 1 EXAMPLE OF THE USE OF THE RELEASE COUNTER. (# 4)

FINAL RESULTS FOR 2 SIMULATION RUN(S)

NODE	PROB./COUNT	MEAN	STAND. DEV.	NO OF OBS.	MIN	MAX	NODE TYPE
4	1.0000	0.0	0.0	2.	0.0	0.0	F

Figure 2.2.14

Computer Results for Example #4 in Figure 2.2.10

```
                         ** EVENT TRACE FOR RUN    1 **

END OF ARC    ARC     END   FIRST  DISTR.  COUNT             RELEASE          ARC            ARC
  TIME      NUMBER  NODE   TYPE   TYPE   TYPE = (    )   COUNT           COST        DURATION

0.0             0       3      0     10      0      0         1    0.0                 0.0
0.3523562E+01   0       2      1      3      0      0         1    0.0                 0.3528562E+01


                         ERROR EXIT, TYPE 11 ERROR.

FILE STATUS AT TIME  0.3529E+01
   P=PREDECESSOR POINTER  S=SUCCESSOR POINTER

FILE    2

CELL=    1    P= 99999   S=    15
ITEMS              2           1           3           0           0           0
ATRIB       0.1000E+01  0.0        0.0         0.0         0.0         0.0

CELL=   15    P=     1   S= 77777
ITEMS              3           0          10           0           0           0
ATRIB       0.1000E+01  0.0        0.0         0.0         0.0         0.0

FILE    3

CELL=   25    P= 99999   S= 77777
ITEMS              0           0          10           0           0           0
ATRIB       0.1000E+01  0.0        0.0         0.0         0.0         0.0

STAT, SUM OF SQ., NUMB. OF OBS., MIN., MAX.

  +( 0)    0.0           0.0          0.0          0.1000E+21  -0.1000E+21
  +COST    0.0           0.0          0.0          0.1000E+21  -0.1000E+21

ARRAY ICELS

  +TIME                0    0    0    0    0    0    0    0    0    0    0
                       0    0    0    0    0    0    0    0    0    0    0
                       0    0    0    0    0    0    0    0    0    0

  +COST                0    0    0    0    0    0    0    0    0    0    0
                       0    0    0    0    0    0    0    0    0    0    0
                       0    0    0    0    0    0    0    0    0    0
```

Figure 2.2.15

Computer Results for Example #5 in Figure 2.2.10

2.2.4 Special Features

The concepts previously introduced are enough to model most repairable systems. Two more features greatly increase the capability of GRASP. These concepts are the ability to cause network modifications dynamically during the simulation and the addition of special nodes called C-nodes which can be used to monitor accumulated system costs. C-nodes characteristics will be described in the section on statistical collection.

Network modifications are triggered by the completion of a certain activity type. Recall that different arcs in the network can have the same number so that network modification can be triggered by the first completion of one of several possible arcs. Symbolism for this mechanism is shown in Figure 2.2.16.

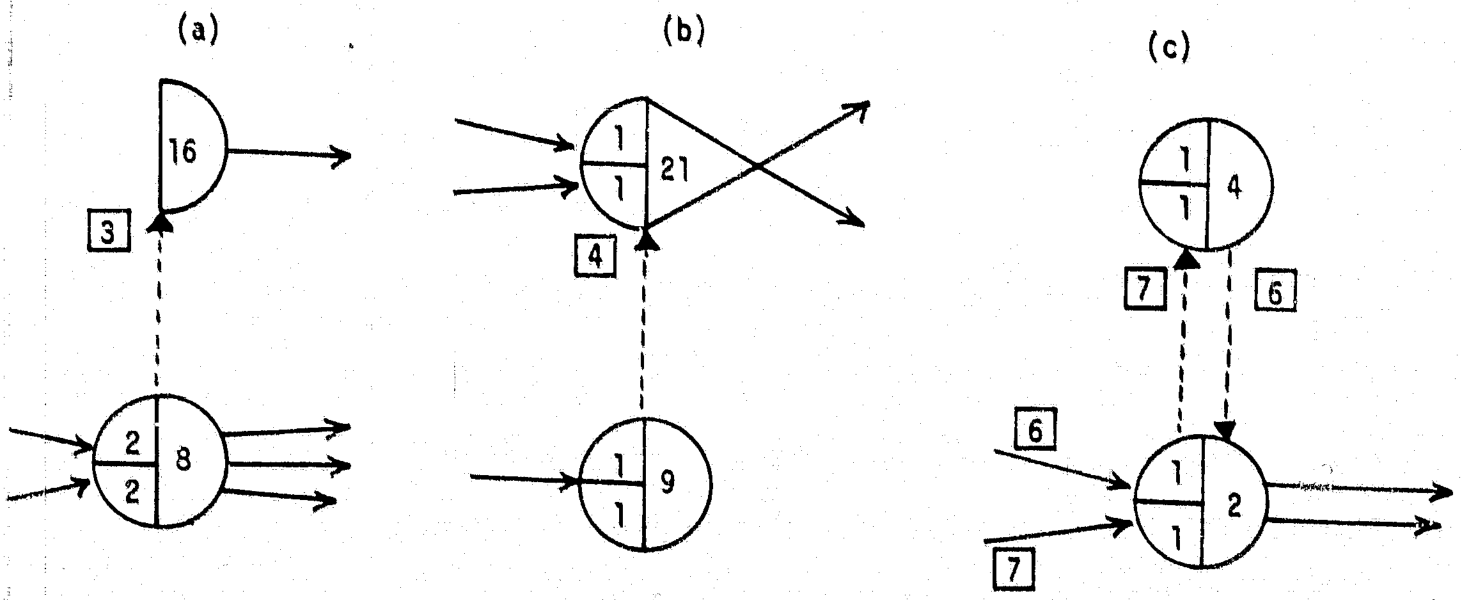


Figure 2.2.16 Node Replacement

In Figure 2.2.16 (a) if any arc with activity number type 3 is completed, then the <u>output side</u> of node 8 will be replaced by the <u>output side</u> of node 16. In other words, when node 8 is released the activities from node 16 will be scheduled instead of those from node 8. The input sides of node 8 and 16 are not affected. It is customary to omit the input side, as in node 16, if it has no input of its own. This is not a requirement, however, as Figure 2.2.16 (b) illustrates. In this case node 21 has an input of its own and operates independently of node 9. However, when node 9 is released after an activity of type 4 has completed, the arcs from node 21 will be scheduled. When this happens, it should be noted that it is node 9 that is released, <u>not</u> node 21. In other words, the conditions which determine node release are associated with the input sides, and node replacement affects only the output side of nodes.

In both Figures 2.2.16 (a) and (b) the network is permanently modified. Activities from the output side of node 8 will never be scheduled again once the network is modified. If we want to go back to the original network, we need to trigger another modification of the network, which will re-insert nodes 8 and 9, respectively.

Figure 2.2.16 (c) illustrates the procedure. The desired outcome is that when node 2 is released by activity 7, no action be taken. (Transactions are terminated at node 4). If activity 6 releases node 2, then activities from node 2 are scheduled. This is accomplished by a double replacement of nodes 4 and 2. Examining the other examples of Figure 2.2.16, it is seen that the replacement arrow always points to the node which is inserted by the associated activity type. GRASP processes network modifications before it examines the end node of the arc to see if it is released. Thus, when release of node 2 occurs, the output side of node 4 is already in the network instead of the output side of node 2. No activities are scheduled. This condition will continue until type 6 activity is completed (i.e., the node replacement will remain in effect until the end of the simulation or until a type 6 activity is completed). Then, as is indicated by the arrow, the output side of node 2 will be returned to the network and the activities leaving it scheduled.

One last point, it is possible in 2.2.16 (c) that activity 6 is completed and node 2 is already in the network. This causes no difficulty. Activities will be scheduled from node 2 as expected.

> NOTE: Non-positive activity numbers cannot be used to facilitate network modifications. Hence, only positive activity types can be used to cause network modifications.

2.2.5 Examples of Network Modifications:

The ability to cause network modifications during a simulation run is an important feature that deserves more attention. Its misuse can cause errors in the model that may be hard for a beginner to detect. The examples below illustrate different cases of node replacements. They are artificial and do not correspond to any reliability system. Their purpose is just to show how the node replacement really works.

Example TIED 1: With Simultaneous Arc Completions:

This model illustrates the effect of two simultaneous arc completions on a node replacement. In the GRASP diagram represented below, all the arcs have a duration equal to zero except arc (3, 5). Arcs (5, 3) and (5, 2) which are identified with numbers 1 and 2 respectively, terminate at the same time. Because of this, the replacement of node 3 by node 4 cannot occur.

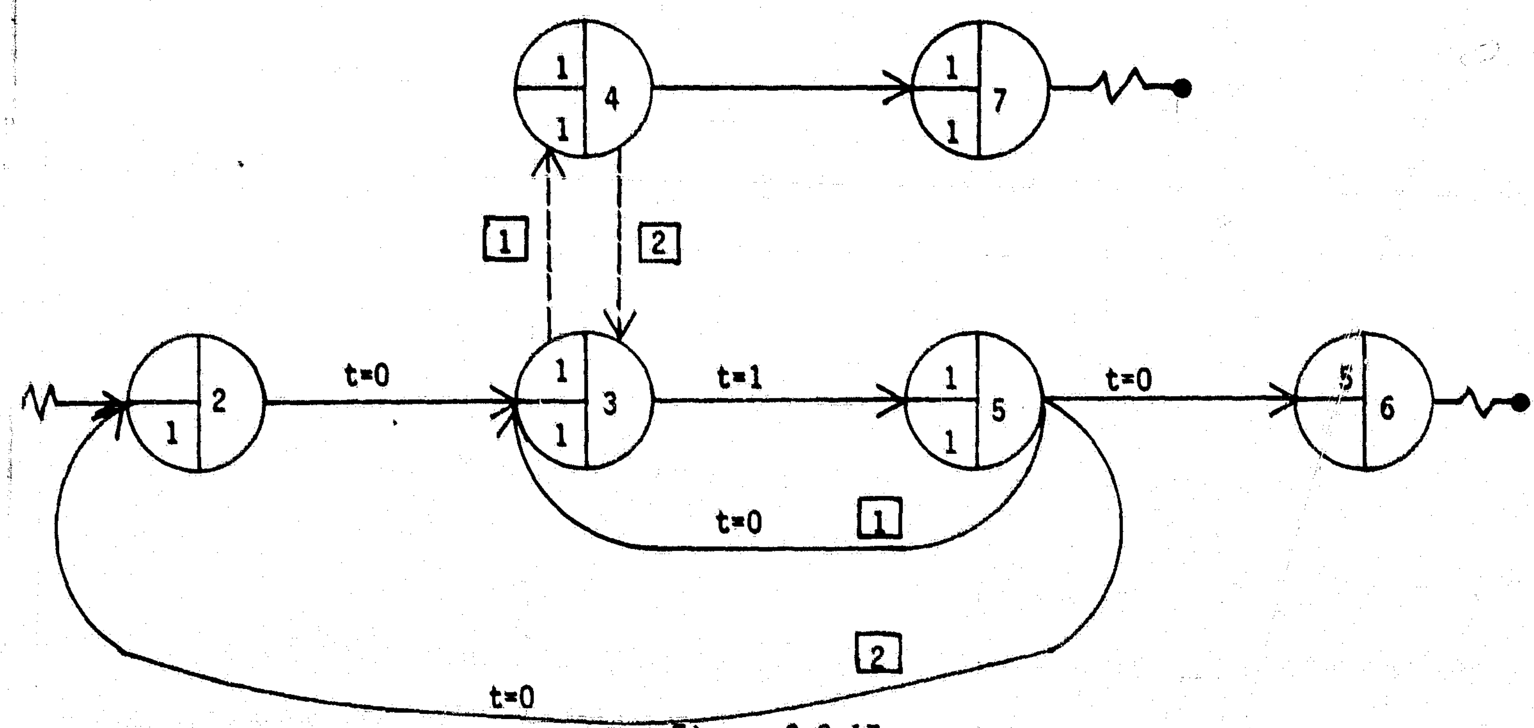


Figure 2.2.17

The results are shown in Figure 2.2.18. Notice that no statistics were recorded at node 7. The event trace for this run shows some interesting points. Node 3 is released at time 0. Since arc number 1 is not completed yet, arc (3, 5) is scheduled. At time t = 1, node 5 is released. Arc (5, 6) decreases the release counter of node 6 from 5 to 4. Arcs (5, 3) and (5, 2) terminate at the same time and they both release nodes 3 and 2 respectively. They are ranked in the event file as (5, 3) first and

RUN

** EVENT TRACE FOR RUN 1 **

END OF ARC TIME	ARC NUMBER	END NODE	PARAM. TYPE	DISTR. TYPE	COUNT TYPE = (	)	RELEASE COUNT	ARC COST	ARC DURATION
0.0	0	3	0	10	0	0	0	0.0	0.0
0.100000E+01	0	5	1	1	0	0	0	0.0	0.100000E+01
0.100000E+01	0	6	0	10	0	0	4	0.0	0.0
0.100001E+01	2	2	0	10	0	0	0	0.0	0.953674E-05
0.100001E+01	1	3	0	10	0	0	0	0.0	0.0
0.100001E+01	0	3	0	10	0	0	0	0.0	0.0
0.200001E+01	0	5	1	1	0	0	0	0.0	0.100000E+01
0.200002E+01	0	5	1	1	0	0	-1	0.0	0.953674E-05
0.200002E+01	0	6	0	10	0	0	3	0.0	0.0
0.200003E+01	2	2	0	10	0	0	0	0.0	0.953674E-05
0.200003E+01	1	3	0	10	0	0	0	0.0	0.0
0.200003E+01	0	3	0	10	0	0	0	0.0	0.0
0.300003E+01	0	5	1	1	0	0	0	0.0	0.100000E+01
0.300004E+01	0	5	1	1	0	0	-1	0.0	0.953674E-05
0.300005E+01	0	6	0	10	0	0	2	0.0	0.0
0.300005E+01	2	2	0	10	0	0	0	0.0	0.953674E-05
0.300005E+01	1	3	0	10	0	0	0	0.0	0.0
0.300005E+01	0	3	0	10	0	0	0	0.0	0.0
0.400005E+01	0	5	1	1	0	0	0	0.0	0.100000E+01
0.400006E+01	0	5	1	1	0	0	-1	0.0	0.953674E-05
0.400006E+01	0	6	0	10	0	0	1	0.0	0.0
0.400007E+01	2	2	0	10	0	0	0	0.0	0.953674E-05
0.400007E+01	1	3	0	10	0	0	0	0.0	0.0
0.400007E+01	0	3	0	10	0	0	0	0.0	0.0
0.500007E+01	0	5	1	1	0	0	0	0.0	0.100000E+01
0.500008E+01	0	5	1	1	0	0	-1	0.0	0.953674E-05
0.500008E+01	0	6	0	10	0	0	0	0.0	0.0
0.500009E+01	2	2	0	10	0	0	0	0.0	0.953674E-05
0.500009E+01	1	3	0	10	0	0	0	0.0	0.0

GRASP SIMULATION PROJECT : TIED1: AN EXAMPLE OF SIMULTANEOUS ARC COMPLETIONS.

FINAL RESULTS FOR 1 SIMULATION RUN(S)

NODE	PROB./COUNT	MEAN	STAND. DEV.	NO OF OBS.	MIN	MAX	NODE TYPE
6	1.0000	0.500009E+01	0.0	1.	0.500009E+01	0.500009E+01	F
7	0	NO VALUES RECORDED					

Figure 2.2.18 Computer results for the example in Figure 2.2.17

ORIGINAL PAGE IS OF POOR QUALITY

and (5, 2) second. This is reflected in the event trace by a duration equal to .953674 E - 5 for arc (5, 2). At time 1 node 3 is released twice, the first time by arc (5, 3) and the second time by arc (2, 3). At time 2, two arcs hit node 5, that is why its release counter drops to -1. It is released just once and the same cycle starts again.

<u>Example TIED 2: Misuse of the Node Replacement Option</u>

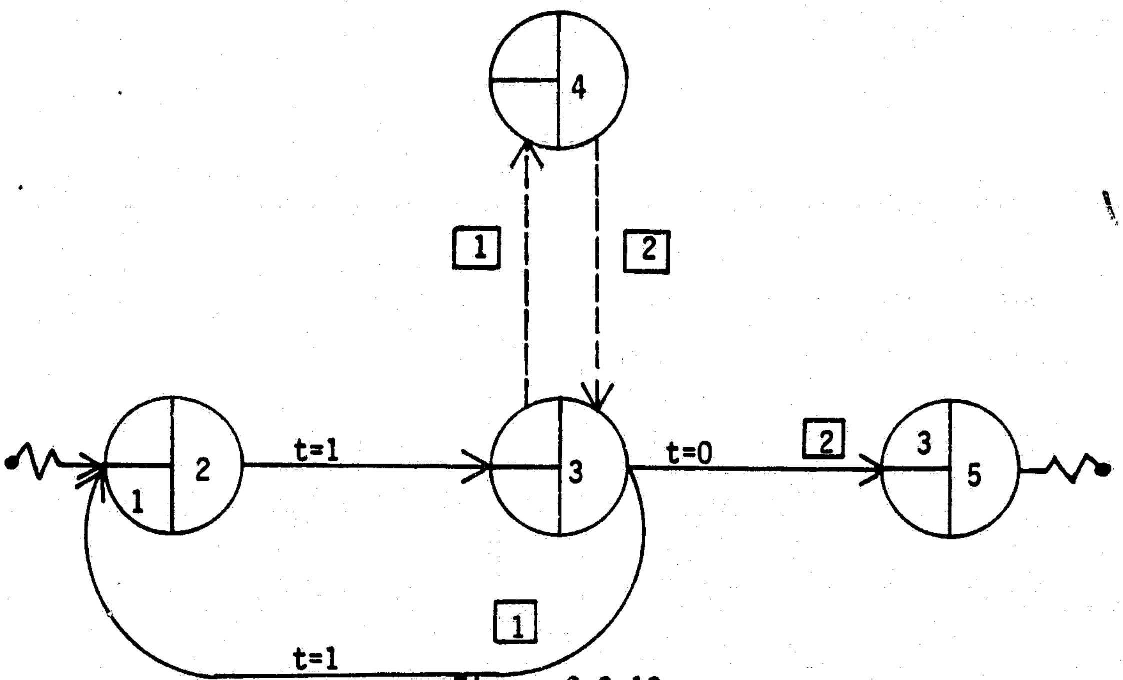


Figure 2.2.19

A node N with no arc going out of it, does not have a file of its own in the array NSET/QSET, and therefore MFE(N), which indicates the starting location of its file, is zero. This does not cause a problem when N is a sink node or there is no attempt to release the node. Otherwise, the program prints an error message and if the event file is empty, terminates abnormally. The computer results are shown in Figure 2.2.20. Only file 2 status is shown. The point of this example is to illustrate a kind of modeling error that should be avoided.

```
RUN

                                   ** EVENT TRACE FOR RUN    1 **

END OF *ARC      ARC     END    PARAM.   DISTR.   COUNT            RELEASE          ARC
   TIME        NUMBER   NODE    TYPE     TYPE    TYPE = (    )    COUNT            COST

0.100000E+01     0        3       1        1        0        0        0     0.0
0.100000E+01     2        5       0       10        0        0        2     0.0
0.200000E+01     1        2       1        1        0        0        0     0.0
0.300000E+01     0        3       1        1        0        0        0     0.0

                                  ERROR EXIT, TYPE 11 ERROR.

FILE STATUS AT TIME  0.3000E+01
    P=PREDECESSOR POINTER  S=SUCCESSOR POINTER

FILE    2

CELL=      1     P= 99999   S= 77777
JTRIB                     3             1             1             0             0
ATRIB           0.1000E+01  0.0           0.0           0.0           0.0           0.0
```

Figure 2.2.20   Computer results for the example in Figure 2.2.19

Example NDR1: Multiple Node Replacements

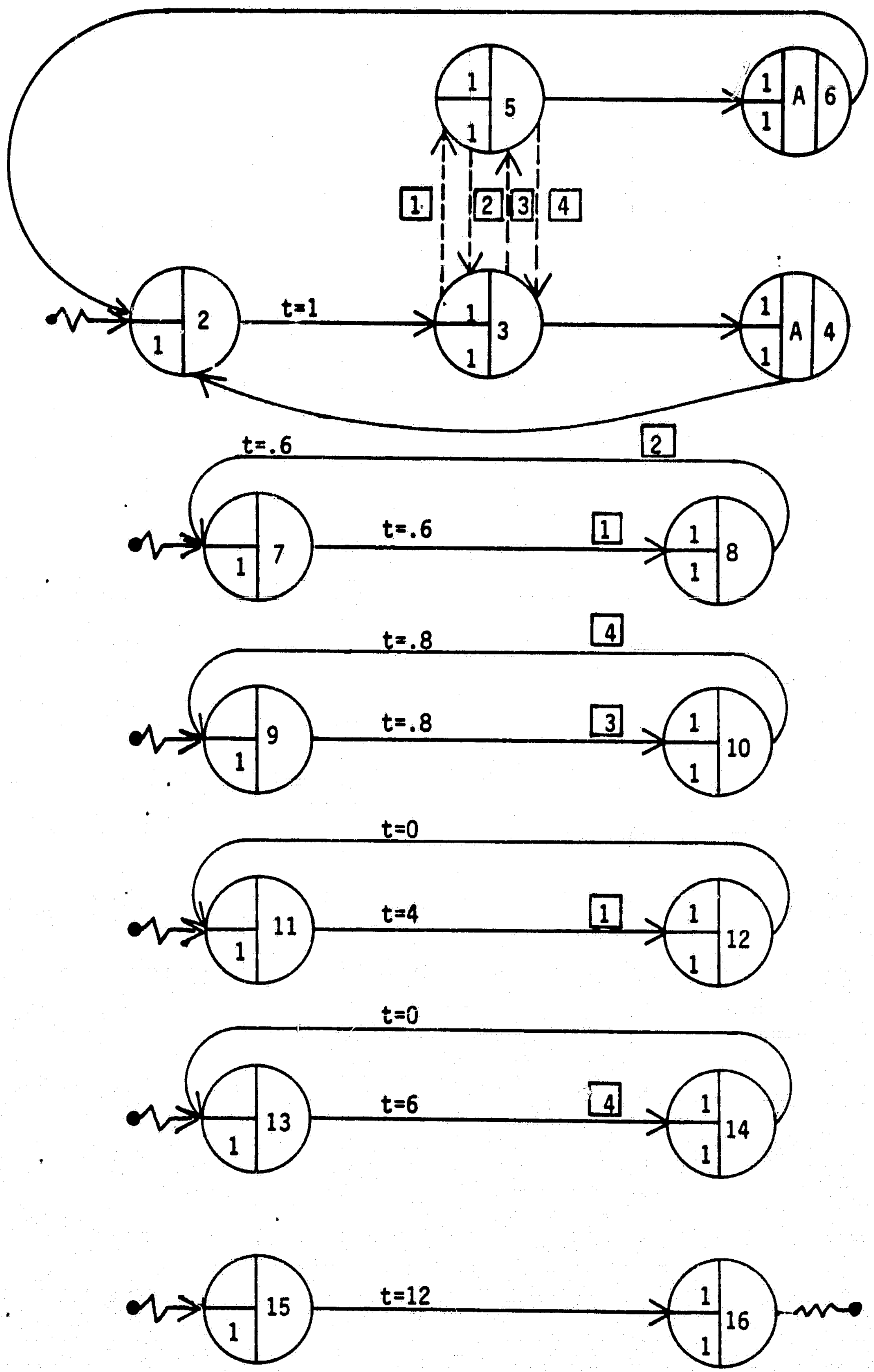


Figure 2.2.21

The arc durations in this example are chosen in such a way that ties occur in all combinations of completions of arcs numbered 1 through 4. The GRASP diagram is shown in Figure 2.2.21.

In the actual replacements of node 3 by node 5, arcs identified with numbers 1 and 2 are assumed to be equivalent. So are the arcs numbered 3 and 4. In Figure 2.2.21, arcs numbered 1 are: (7, 8) and (11, 12). Only one arc is identified with number 2, that is (8, 7), and only one arc is identified with number 3; that is (9, 10). Two arcs are identified with number 4: (10, 9) and (13, 14). The computer results are shown in Figure 2.2.22. A summary of the node releases is shown in Figure 2.2.23.

If arcs (7, 8) and (9, 10) represent for instance failures of different items, and the repair of one cannot be substituted for the repair of the other one, then the model above is no longer valid. The next example shows how this can be accomplished.

Figure 2.2.22

Computer Results for the Example in Figure 2.2.21

** EVENT TRACE FOR RUN 1 **

END OF ARC TIME	ARC NUMBER	END NODE	PARAM. TYPE	DISTR. TYPE	COUNT TYPE = (	)	RELEASE COUNT	ARC COST	ARC DURATION
0.600000E+00	1	8	2	1	0	0	0	0.0	0.600000E+00
0.800000E+00	3	10	3	1	0	0	0	0.0	0.200000E+00
0.100000E+01	0	3	1	1	0	0	0	0.0	0.200000E+00
0.100000E+01	0	6	0	10	0	0	0	0.0	0.0
0.100000E+01	0	2	0	10	0	0	0	0.0	0.0
0.120000E+01	2	7	2	1	0	0	0	0.0	0.200000E+00
0.160000E+01	4	9	3	1	0	0	0	0.0	0.400000E+00
0.180000E+01	1	8	2	1	0	0	0	0.0	0.200000E+00
0.200000E+01	0	3	1	1	0	0	0	0.0	0.200001E+00
0.200000E+01	0	6	0	10	0	0	0	0.0	0.0
0.200000E+01	0	2	0	10	0	0	0	0.0	0.0
0.240000E+01	3	10	3	1	0	0	0	0.0	0.399999E+00
0.240001E+01	2	7	2	1	0	0	0	0.0	0.953674E-05
0.300000E+01	0	3	1	1	0	0	0	0.0	0.599992E+00
0.300001E+01	1	8	2	1	0	0	0	0.0	0.762939E-05
0.300001E+01	0	6	0	10	0	0	0	0.0	0.0
0.300001E+01	0	2	0	10	0	0	0	0.0	0.0
0.320001E+01	4	9	3	1	0	0	0	0.0	0.200000E+00
0.360001E+01	2	7	2	1	0	0	0	0.0	0.400000E+00
0.400000E+01	1	12	5	1	0	0	0	0.0	0.399993E+00
0.400001E+01	3	10	3	1	0	0	0	0.0	0.667572E-05
0.400001E+01	0	3	1	1	0	0	0	0.0	0.953674E-06
0.400001E+01	0	11	0	10	0	0	0	0.0	0.0
0.400002E+01	0	6	0	10	0	0	0	0.0	0.953674E-05
0.400002E+01	0	2	0	10	0	0	0	0.0	0.0
0.420001E+01	1	8	2	1	0	0	0	0.0	0.199989E+00
0.480001E+01	2	7	2	1	0	0	0	0.0	0.599999E+00
0.480001E+01	4	9	3	1	0	0	0	0.0	0.953674E-06
0.500002E+01	0	3	1	1	0	0	0	0.0	0.200010E+00
0.500002E+01	0	4	0	10	0	0	0	0.0	0.0
0.500002E+01	0	2	0	10	0	0	0	0.0	0.0
0.540001E+01	1	8	2	1	0	0	0	0.0	0.399989E+00
0.560001E+01	3	10	3	1	0	0	0	0.0	0.200000E+00
0.600000E+01	4	14	6	1	0	0	0	0.0	0.399994E+00
0.600001E+01	2	7	2	1	0	0	0	0.0	0.572205E-05
0.600001E+01	0	13	0	10	0	0	0	0.0	0.0
0.600002E+01	0	3	1	1	0	0	0	0.0	0.114441E-04
0.600002E+01	0	4	0	10	0	0	0	0.0	0.0
0.600002E+01	0	2	0	10	0	0	0	0.0	0.0
0.640001E+01	4	9	3	1	0	0	0	0.0	0.399988E+00
0.660001E+01	1	8	2	1	0	0	0	0.0	0.200000E+00
0.700002E+01	0	3	1	1	0	0	0	0.0	0.400012E+00
0.700002E+01	0	6	0	10	0	0	0	0.0	0.0
0.700002E+01	0	2	0	10	0	0	0	0.0	0.0
0.720000E+01	3	10	3	1	0	0	0	0.0	0.199987E+00
0.720001E+01	2	7	2	1	0	0	0	0.0	0.953674E-05
0.780001E+01	1	8	2	1	0	0	0	0.0	0.599999E+00
0.800001E+01	4	9	3	1	0	0	0	0.0	0.200000E+00
0.800002E+01	1	12	5	1	0	0	0	0.0	0.381470E-05
0.800003E+01	0	3	1	1	0	0	0	0.0	0.953674E-05
0.800003E+01	0	11	0	10	0	0	0	0.0	0.0
0.800004E+01	0	6	0	10	0	0	0	0.0	0.953674E-05
0.800004E+01	0	2	0	10	0	0	0	0.0	0.0
0.840001E+01	2	7	2	1	0	0	0	0.0	0.399977E+00
0.880003E+01	3	10	3	1	0	0	0	0.0	0.400013E+00
0.900001E+01	1	8	2	1	0	0	0	0.0	0.199986E+00
0.900004E+01	0	3	1	1	0	0	0	0.0	0.238419E-04
0.900004E+01	0	6	0	10	0	0	0	0.0	0.0
0.900004E+01	0	2	0	10	0	0	0	0.0	0.0
0.960001E+01	2	7	2	1	0	0	0	0.0	0.599976E+00
0.960003E+01	4	9	3	1	0	0	0	0.0	0.133514E-04
0.100000E+02	0	3	1	1	0	0	0	0.0	0.400011E+00
0.100000E+02	0	4	0	10	0	0	0	0.0	0.0
0.100000E+02	0	2	0	10	0	0	0	0.0	0.0
0.102000E+02	1	8	2	1	0	0	0	0.0	0.199975E+00
0.104000E+02	3	10	3	1	0	0	0	0.0	0.200013E+00
0.108000E+02	2	7	2	1	0	0	0	0.0	0.399986E+00
0.110000E+02	0	3	1	1	0	0	0	0.0	0.200026E+00
0.110000E+02	0	4	0	10	0	0	0	0.0	0.0
0.110000E+02	0	2	0	10	0	0	0	0.0	0.0
0.112000E+02	4	9	3	1	0	0	0	0.0	0.199987E+00
0.114000E+02	1	8	2	1	0	0	0	0.0	0.199986E+00
0.120000E+02	0	16	4	1	0	0	0	0.0	0.599993E+00
0.120000E+02	4	14	6	1	0	0	0	0.0	0.572205E-05
0.120000E+02	2	7	2	1	0	0	0	0.0	0.381470E-05

GRASP SIMULATION PROJECT : NDR1: AN EXAMPLE ON SIMULTANEOUS NODE REPLACEMENTS

FINAL RESULTS FOR 1 SIMULATION RUN(S)

NODE	PROB./ COUNT	MEAN	STAND. DEV.	NO OF OBS.	MIN	MAX	NODE TYPE
16	1.0000	0.120000E+02	0.0	1.	0.120000E+02	0.120000E+02	F
6	1.0000	0.485716E+01	0.313203E+01	7.	0.100000E+01	0.900004E+01	A
[illegible]	1.0000	0.800003E+01	0.294392E+01	4.	0.500002E+01	0.110000E+02	A

Nodes \ Time	0	.6	.8	1	1.2	1.6	1.8	2	2.4	3	3.2	3.6	4	4.2	4.8	5	5.4	5.6	6	6.4	6.6	7	7.2	7.8	8	8.4	8.8	9	9.6	10	10.2	10.4	10.8	11	11.2	11.4	12
2	X			X				X		X			X			X			X			X			X			X		X				X			X
3				X				X		X			X			X			X			X			X			X		X				X			X
4																X			X											X				X			
5																																					
6				X				X		X			X									X			X			X									
7	X				X				X			X			X				X				X			X			X				X				X
8		X					X			X				X			X				X			X				X			X					X	
9	X					X					X				X					X					X				X						X		
10			X						X				X					X					X				X					X					X
11	X												X												X												X
12													X												X												X
13	X																		X																		X
14																			X																		X

Figure 2.2.23

A Summary of the Node Releases
for the Example in Figure 2.2.21

Example NDR2: Multiple Node Replacements

The model is basically the same as NDR1 except for the first sub-network which is modified as shown in Figure 2.2.24.

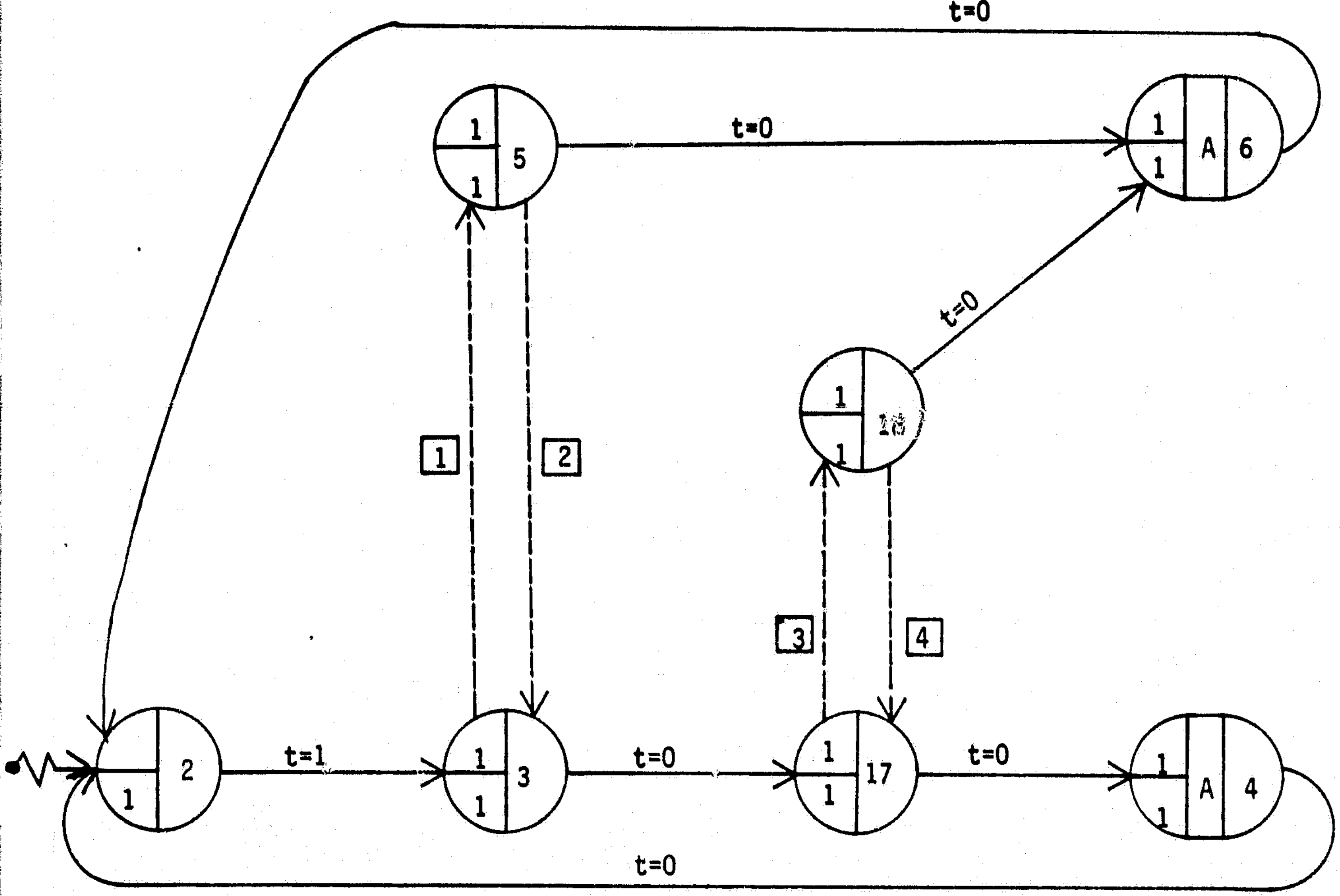


Figure 2.2.24

The results from this example are shown in Figures 2.2.25-26. The tracing option allows to make comparisons with the previous example. For instance, at time 11 the transaction goes thorugh node 6 and not through node 4.

Figure 2.2.25

Computer Results for

the Example in Figure 2.2.24

END OF ARC TIME	ARC NUMBER	END NODE	PARAM. TYPE	DISTR. TYPE	COUNT TYPE = (	)	RELEASE COUNT	ARC COST	ARC DURATION
0.600000E+00	1	8	2	1	0	0	0	0.0	0.600000E+00
0.800000E+00	3	10	3	1	0	0	0	0.0	0.200000E+00
0.100000E+01	0	3	1	1	0	0	0	0.0	0.200000E+00
0.100000E+01	0	6	0	10	0	0	0	0.0	0.0
0.100000E+01	0	2	0	10	0	0	0	0.0	0.0
0.120000E+01	2	7	2	1	0	0	0	0.0	0.200000E+00
0.160000E+01	4	9	3	1	0	0	0	0.0	0.400000E+00
0.180000E+01	1	8	2	1	0	0	0	0.0	0.200000E+00
0.200000E+01	0	3	1	1	0	0	0	0.0	0.200001E+00
0.200000E+01	0	6	0	10	0	0	0	0.0	0.0
0.200000E+01	0	2	0	10	0	0	0	0.0	0.0
0.240000E+01	3	10	3	1	0	0	0	0.0	0.399999E+00
0.240001E+01	2	7	2	1	0	0	0	0.0	0.953674E-05
0.300000E+01	0	3	1	1	0	0	0	0.0	0.599992E+00
0.300001E+01	1	8	2	1	0	0	0	0.0	0.762939E-05
0.300001E+01	0	6	0	10	0	0	0	0.0	0.0
0.300001E+01	0	2	0	10	0	0	0	0.0	0.0
0.320001E+01	4	9	3	1	0	0	0	0.0	0.200000E+00
0.360001E+01	2	7	2	1	0	0	0	0.0	0.400000E+00
0.400000E+01	1	12	5	1	0	0	0	0.0	0.399993E+00
0.400001E+01	3	10	3	1	0	0	0	0.0	0.667572E-05
0.400001E+01	0	3	1	1	0	0	0	0.0	0.953674E-06
0.400001E+01	0	11	0	10	0	0	0	0.0	0.0
0.400002E+01	0	6	0	10	0	0	0	0.0	0.953674E-05
0.400002E+01	0	2	0	10	0	0	0	0.0	0.0
0.420001E+01	1	8	2	1	0	0	0	0.0	0.199989E+00
0.480001E+01	2	7	2	1	0	0	0	0.0	0.599999E+00
0.480001E+01	4	9	3	1	0	0	0	0.0	0.953674E-06
0.500002E+01	0	3	1	1	0	0	0	0.0	0.200010E+00
0.500002E+01	0	17	0	10	0	0	0	0.0	0.0
0.500002E+01	0	4	0	10	0	0	0	0.0	0.0
0.500002E+01	0	2	0	10	0	0	0	0.0	0.0
0.540001E+01	1	8	2	1	0	0	0	0.0	0.399989E+00
0.560001E+01	3	10	3	1	0	0	0	0.0	0.200000E+00
0.600000E+01	4	14	6	1	0	0	0	0.0	0.399994E+00
0.600001E+01	2	7	2	1	0	0	0	0.0	0.572205E-05
0.600001E+01	0	13	0	10	0	0	0	0.0	0.0
0.600002E+01	0	3	1	1	0	0	0	0.0	0.114441E-04
0.600002E+01	0	17	0	10	0	0	0	0.0	0.0
0.600002E+01	0	4	0	10	0	0	0	0.0	0.0
0.600002E+01	0	2	0	10	0	0	0	0.0	0.0
0.640001E+01	4	9	3	1	0	0	0	0.0	0.399988E+00
0.660001E+01	1	8	2	1	0	0	0	0.0	0.200000E+00
0.700002E+01	0	3	1	1	0	0	0	0.0	0.400012E+00
0.700002E+01	0	6	0	10	0	0	0	0.0	0.0
0.700002E+01	0	2	0	10	0	0	0	0.0	0.0
0.720000E+01	3	10	3	1	0	0	0	0.0	0.199987E+00
0.720001E+01	2	7	2	1	0	0	0	0.0	0.953674E-05
0.780001E+01	1	8	2	1	0	0	0	0.0	0.599999E+00
0.800001E+01	4	9	3	1	0	0	0	0.0	0.200000E+00
0.800002E+01	1	12	5	1	0	0	0	0.0	0.381470E-05
0.800003E+01	0	3	1	1	0	0	0	0.0	0.953674E-05
0.800003E+01	0	11	0	10	0	0	0	0.0	0.0
0.800004E+01	0	6	0	10	0	0	0	0.0	0.953674E-05
0.800004E+01	0	2	0	10	0	0	0	0.0	0.0
0.840001E+01	2	7	2	1	0	0	0	0.0	0.399977E+00
0.880003E+01	3	10	3	1	0	0	0	0.0	0.400013E+00
0.900001E+01	1	8	2	1	0	0	0	0.0	0.199986E+00
0.900004E+01	0	3	1	1	0	0	0	0.0	0.238419E-04
0.900004E+01	0	6	0	10	0	0	0	0.0	0.0
0.900004E+01	0	2	0	10	0	0	0	0.0	0.0
0.960001E+01	2	7	2	1	0	0	0	0.0	0.599976E+00
0.960003E+01	4	9	3	1	0	0	0	0.0	0.133514E-04
0.100000E+02	0	3	1	1	0	0	0	0.0	0.400011E+00
0.100000E+02	0	17	0	10	0	0	0	0.0	0.0
0.100000E+02	0	4	0	10	0	0	0	0.0	0.0
0.100000E+02	0	2	0	10	0	0	0	0.0	0.0
0.102000E+02	1	8	2	1	0	0	0	0.0	0.199975E+00
0.104000E+02	3	10	3	1	0	0	0	0.0	0.200013E+00
0.108000E+02	2	7	2	1	0	0	0	0.0	0.399986E+00
0.110000E+02	0	3	1	1	0	0	0	0.0	0.200026E+00
0.110000E+02	0	17	0	10	0	0	0	0.0	0.0
0.110000E+02	0	6	0	10	0	0	0	0.0	0.0
0.110000E+02	0	2	0	10	0	0	0	0.0	0.0
0.112000E+02	4	9	3	1	0	0	0	0.0	0.199987E+00
0.114000E+02	1	8	2	1	0	0	0	0.0	0.199986E+00
0.120000E+02	0	16	4	1	0	0	0	0.0	0.599990E+00
0.120000E+02	4	14	6	1	0	0	0	0.0	0.572205E-05
0.120000E+02	2	7	2	1	0	0	0	0.0	0.381470E-05

GRASP SIMULATION PROJECT : NDR2: AN EXAMPLE ON SIMULTANEOUS NODE REPLACEMENTS

FINAL RESULTS FOR 1 SIMULATION RUN(S)

NODE	PROB./ COUNT	MEAN	STAND. DEV.	NO OF OBS.	MIN	MAX	NODE TYPE
16	1.0000	0.120000E+02	0.0	1.	0.120000E+02	0.120000E+02	F
6	1.0000	0.562502E+01	0.362286E+01	8.	0.100000E+01	0.110000E+02	A

ORIGINAL PAGE IS
OF POOR QUALITY

Nodes \ Time	0	.6	.8	1	1.2	1.6	1.8	2	2.4	3	3.2	3.6	4	4.2	4.8	5	5.4	5.6	6	6.4	6.6	7	7.2	7.8	8	8.4	8.8	9	9.6	10	10.2	10.4	10.8	11	11.2	11.4	12
2	X			X				X		X			X			X			X			X			X			X		X				X			X
3				X				X		X			X			X			X			X			X			X		X				X			X
17																X			X											X				X			
4																X			X											X							
6				X				X		X			X									X			X			X						X			
7	X				X				X			X			X				X				X			X			X				X				X
8		X					X			X				X			X				X			X				X			X					X	
9	X					X					X				X					X					X				X						X		
10			X						X				X					X					X				X					X					X
11	X												X												X												X
12													X												X												X
13	X																		X																		X
14																			X																		X

Figure 2.2.26

A Summary of the Node Releases
for the Example in Figure 2.2.24

Example NDR3: Multiple Node Replacements

This example illustrates another case of complex node replacements. It differs from the previous two examples in the first subnetwork only, which is represented in Figure 2.2.27.

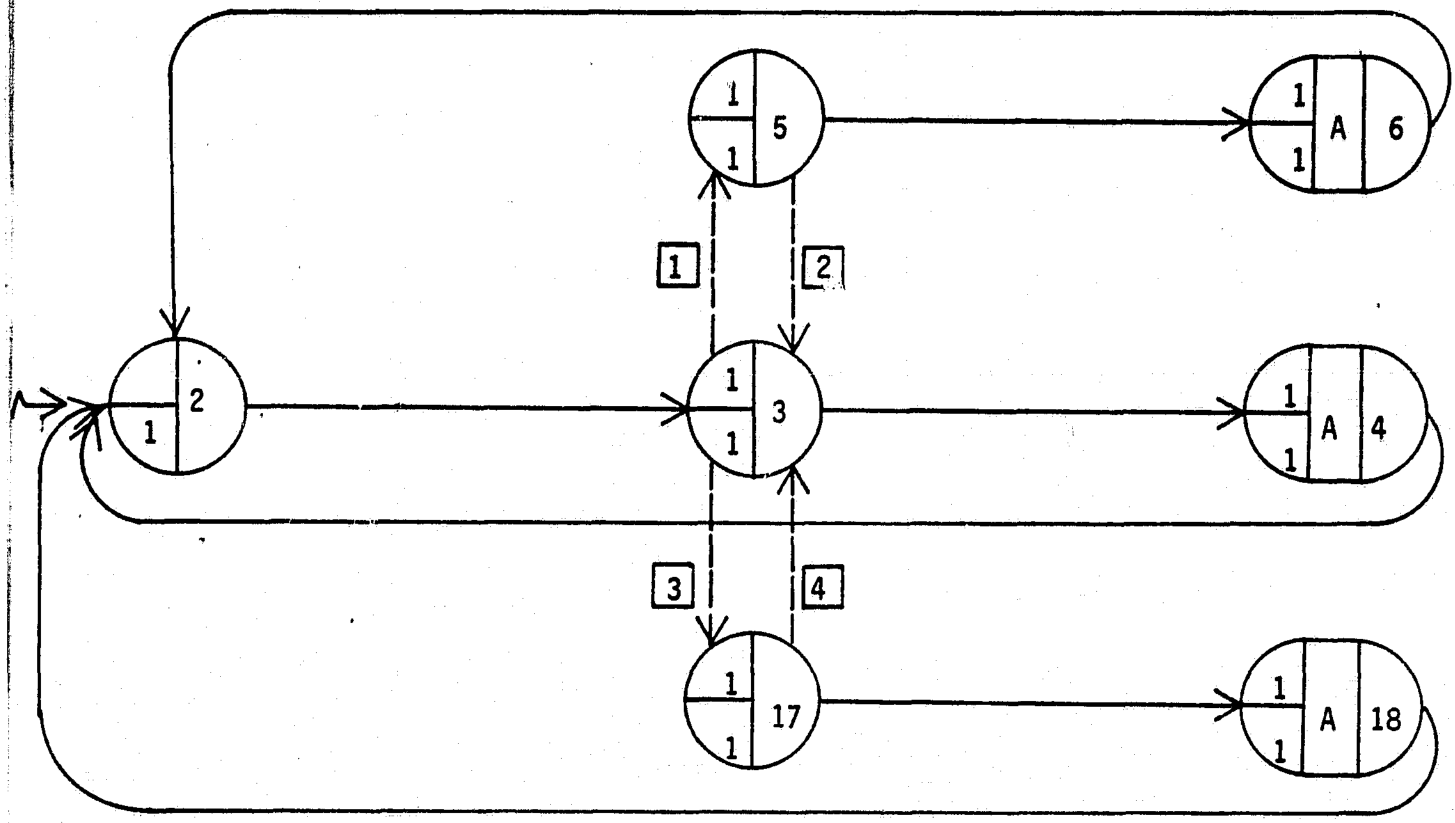


Figure 2.2.27

The computer results are given in Figure 2.2.28, and a summary of all the releases of the nodes is provided in Figure 2.2.29. It shows how one node can be replaced by two other nodes at the same time. That is what happened at times 1, 3, 4 and 9. In these cases, two arcs, namely (6, 2) and (18, 2) terminate at node 2 at the same time, but node 2 is released just once. Other combinations of arc completions occur also. Node 3 is replaced by node 5 only at times 2, 7 and 8. It is replaced by node 17 only at time 11 and it is not replaced at all at times 5, 6 and 10. The reader can verify, using Figure 2.2.29, that the statistics given in the computer results are correct.

Figure 2.2.28

Computer Results for the

Example in Figure 2.2.27

0.600000E+00	1	8	2	1	0	0	0	0.0	0.600000E+00
0.800000E+00	3	10	3	1	0	0	0	0.0	0.200000E+00
0.100000E+01	0	3	1	1	0	0	0	0.0	0.200000E+00
0.100000E+01	0	6	0	10	0	0	0	0.0	0.0
0.100001E+01	0	18	0	10	0	0	0	0.0	0.953674E-05
0.100001E+01	0	2	0	10	0	0	0	0.0	0.0
0.100002E+01	0	2	0	10	0	0	-1	0.0	0.953674E-05
0.120000E+01	2	7	2	1	0	0	0	0.0	0.199981E+00
0.160000E+01	4	9	3	1	0	0	0	0.0	0.400000E+00
0.180000E+01	1	8	2	1	0	0	0	0.0	0.200000E+00
0.200002E+01	0	3	1	1	0	0	0	0.0	0.200020E+00
0.200002E+01	0	6	0	10	0	0	0	0.0	0.0
0.200002E+01	0	2	0	10	0	0	0	0.0	0.0
0.240000E+01	3	10	3	1	0	0	0	0.0	0.399980E+00
0.240001E+01	2	7	2	1	0	0	0	0.0	0.953674E-05
0.300001E+01	1	8	2	1	0	0	0	0.0	0.599999E+00
0.300002E+01	0	3	1	1	0	0	0	0.0	0.114441E-04
0.300002E+01	0	18	0	10	0	0	0	0.0	0.0
0.300003E+01	0	6	0	10	0	0	0	0.0	0.953674E-05
0.300003E+01	0	2	0	10	0	0	0	0.0	0.0
0.300004E+01	0	2	0	10	0	0	-1	0.0	0.953674E-05
0.320001E+01	4	9	3	1	0	0	0	0.0	0.199969E+00
0.360001E+01	2	7	2	1	0	0	0	0.0	0.400000E+00
0.400000E+01	1	12	5	1	0	0	0	0.0	0.399993E+00
0.400001E+01	3	10	3	1	0	0	0	0.0	0.667572E-05
0.400001E+01	0	11	0	10	0	0	0	0.0	0.0
0.400004E+01	0	3	1	1	0	0	0	0.0	0.314713E-04
0.400004E+01	0	6	0	10	0	0	0	0.0	0.0
0.400005E+01	0	18	0	10	0	0	0	0.0	0.953674E-05
0.400005E+01	0	2	0	10	0	0	0	0.0	0.0
0.400006E+01	0	2	0	10	0	0	-1	0.0	0.953674E-05
0.420001E+01	1	8	2	1	0	0	0	0.0	0.199949E+00
0.480001E+01	4	9	3	1	0	0	0	0.0	0.599999E+00
0.480002E+01	2	7	2	1	0	0	0	0.0	0.953674E-05
0.500006E+01	0	3	1	1	0	0	0	0.0	0.200042E+00
0.500006E+01	0	4	0	10	0	0	0	0.0	0.0
0.500006E+01	0	2	0	10	0	0	0	0.0	0.0
0.540001E+01	1	8	2	1	0	0	0	0.0	0.399958E+00
0.560001E+01	3	10	3	1	0	0	0	0.0	0.200000E+00
0.600000E+01	4	14	6	1	0	0	0	0.0	0.399985E+00
0.600000E+01	0	13	0	10	0	0	0	0.0	0.0
0.600001E+01	2	7	2	1	0	0	0	0.0	0.143051E-04
0.600006E+01	0	3	1	1	0	0	0	0.0	0.429153E-04
0.600006E+01	0	4	0	10	0	0	0	0.0	0.0
0.600006E+01	0	2	0	10	0	0	0	0.0	0.0
0.640001E+01	4	9	3	1	0	0	0	0.0	0.399957E+00
0.660001E+01	1	8	2	1	0	0	0	0.0	0.200000E+00
0.700006E+01	0	3	1	1	0	0	0	0.0	0.400043E+00
0.700006E+01	0	6	0	10	0	0	0	0.0	0.0
0.700006E+01	0	2	0	10	0	0	0	0.0	0.0
0.720001E+01	3	10	3	1	0	0	0	0.0	0.199956E+00
0.720002E+01	2	7	2	1	0	0	0	0.0	0.953674E-05
0.780002E+01	1	8	2	1	0	0	0	0.0	0.599999E+00
0.800001E+01	1	12	5	1	0	0	0	0.0	0.199985E+00
0.800001E+01	0	11	0	10	0	0	0	0.0	0.0
0.800002E+01	4	9	3	1	0	0	0	0.0	0.152588E-04
0.800006E+01	0	3	1	1	0	0	0	0.0	0.352859E-04
0.800006E+01	0	6	0	10	0	0	0	0.0	0.0
0.800006E+01	0	2	0	10	0	0	0	0.0	0.0
0.840002E+01	2	7	2	1	0	0	0	0.0	0.399964E+00
0.880002E+01	3	10	3	1	0	0	0	0.0	0.400000E+00
0.900002E+01	1	8	2	1	0	0	0	0.0	0.200000E+00
0.900006E+01	0	3	1	1	0	0	0	0.0	0.362396E-04
0.900006E+01	0	18	0	10	0	0	0	0.0	0.0
0.900007E+01	0	6	0	10	0	0	0	0.0	0.953674E-05
0.900007E+01	0	2	0	10	0	0	0	0.0	0.0
0.900008E+01	0	2	0	10	0	0	-1	0.0	0.953674E-05
0.960002E+01	4	9	3	1	0	0	0	0.0	0.599944E+00
0.960003E+01	2	7	2	1	0	0	0	0.0	0.953674E-05
0.100001E+02	0	3	1	1	0	0	0	0.0	0.400046E+00
0.100001E+02	0	4	0	10	0	0	0	0.0	0.0
0.100001E+02	0	2	0	10	0	0	0	0.0	0.0
0.102000E+02	1	8	2	1	0	0	0	0.0	0.199953E+00
0.104000E+02	3	10	3	1	0	0	0	0.0	0.200000E+00
0.108000E+02	2	7	2	1	0	0	0	0.0	0.400000E+00
0.110001E+02	0	3	1	1	0	0	0	0.0	0.200047E+00
0.110001E+02	0	18	0	10	0	0	0	0.0	0.0
0.110001E+02	0	2	0	10	0	0	0	0.0	0.0
0.112000E+02	4	9	3	1	0	0	0	0.0	0.199952E+00
0.114000E+02	1	8	2	1	0	0	0	0.0	0.200000E+00
0.120000E+02	0	16	4	1	0	0	0	0.0	0.599972E+00
0.120000E+02	1	12	5	1	0	0	0	0.0	0.667572E-05
0.120000E+02	4	14	6	1	0	0	0	0.0	0.286102E-05

ORIGINAL PAGE IS OF POOR QUALITY

GRASP SIMULATION PROJECT : NDR3: AN EXAMPLE ON SIMULTANEOUS NODE REPLACEMENTS

FINAL RESULTS FOR 1 SIMULATION RUN(S)

NODE	PROB./ COUNT	MEAN	STAND. DEV.	NO OF OBS.	MIN	MAX	NODE TYPE
16	1.0000	0.120000E+02	0.0	1.	0.120000E+02	0.120000E+02	F
18	1.0000	0.560005E+01	0.421903E+01	5.	0.100001E+01	0.110001E+02	A
6	1.0000	0.485718E+01	0.313204E+01	7.	0.100001E+01	0.900007E+01	A

ORIGINAL PAGE IS
OF POOR QUALITY

Nodes \ Time	0	.6	.8	1	1.2	1.6	1.8	2	2.4	3	3.2	3.6	4	4.2	4.8	5	5.4	5.6	6	6.4	6.6	7	7.2	7.8	8	8.4	8.8	9	9.6	10	10.2	10.4	10.8	11	11.2	11.4	12
2	X			X				X		X			X			X			X			X			X			X		X				X			X
3				X				X		X			X			X			X			X			X			X		X				X			X
4																X			X											X							
6				X				X		X			X									X			X			X									
18				X						X			X															X						X			
7	X				X				X			X			X				X				X			X			X				X				X
8		X					X			X				X			X				X			X				X			X					X	
9	X					X					X				X					X					X				X						X		
10			X						X				X					X					X				X					X					X
11	X												X												X												X
12													X												X												X
13	X																		X																		X
14																			X																		X

Figure 2.2.29

A Summary of the Node Releases
for the Example in Figure 2.2.27

Example NDR4: Another Misuse of Node Replacements

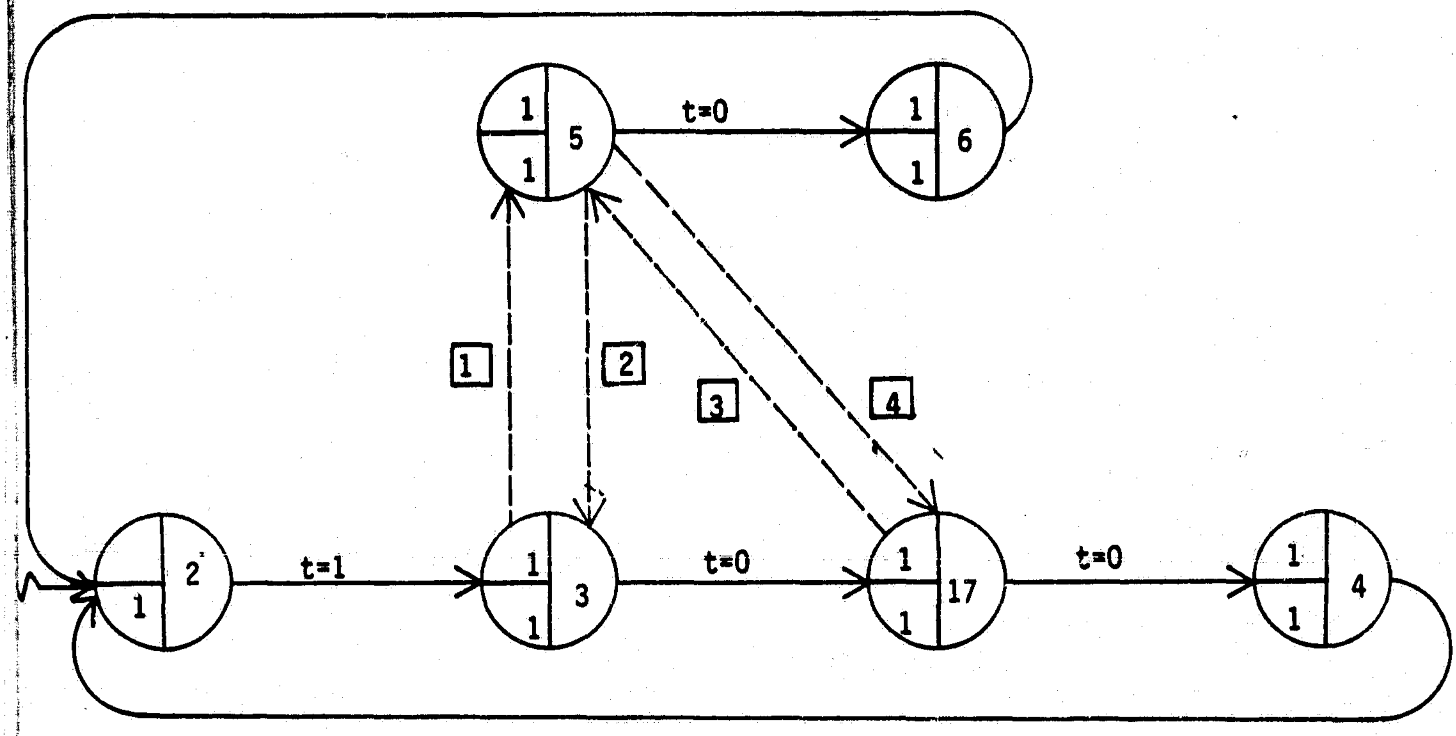


Figure 2.2.30

This example illustrates another problem that can occur with the use of node replacements. It is easy to tell from the network diagram in Figure 2.2.30 that a completion of arcs numbered 3 and 2 will cause the output of node 17 to be replaced by the output of node 3. Since arc (3, 17) has a duration equal to zero, this replacement will occur over and over again at the same time, causing the number of simultaneous arc completions to exceed the maximum allowed. The program detects and prints an error message as shown in Figure 2.2.31. If the time an arc (3, 17) were greater than zero, then such an error would not have been detected by the program.

Figure 2.2.31

Computer Results for the

Example in Figure 2.2.30

0.100000E+01	0	2	0	10	0	0	0	0.0	0.0
0.120000E+01	2	7	2	1	0	0	0	0.0	0.200000E+00
0.160000E+01	4	9	3	1	0	0	0	0.0	0.400000E+00
0.180000E+01	1	8	2	1	0	0	0	0.0	0.200000E+00
0.200000E+01	0	3	1	1	0	0	0	0.0	0.200001E+00
0.200000E+01	0	4	0	10	0	0	0	0.0	0.0
0.200000E+01	0	2	0	10	0	0	0	0.0	0.0
0.240000E+01	3	10	3	1	0	0	0	0.0	0.399999E+00
0.240001E+01	2	7	2	1	0	0	0	0.0	0.953674E-05
0.300000E+01	0	3	1	1	0	0	0	0.0	0.599992E+00
0.300001E+01	1	8	2	1	0	0	0	0.0	0.762939E-05
0.300001E+01	0	6	0	10	0	0	0	0.0	0.0
0.300001E+01	0	2	0	10	0	0	0	0.0	0.0
0.320001E+01	4	9	3	1	0	0	0	0.0	0.200000E+00
0.360001E+01	2	7	2	1	0	0	0	0.0	0.400000E+00
0.400000E+01	1	12	5	1	0	0	0	0.0	0.399993E+00
0.400001E+01	3	10	3	1	0	0	0	0.0	0.667572E-05
0.400001E+01	0	3	1	1	0	0	0	0.0	0.953674E-06
0.400001E+01	0	11	0	10	0	0	0	0.0	0.0
0.400002E+01	0	6	0	10	0	0	0	0.0	0.953674E-05
0.400002E+01	0	2	0	10	0	0	0	0.0	0.0
0.420001E+01	1	8	2	1	0	0	0	0.0	0.199989E+00
0.480001E+01	2	7	2	1	0	0	0	0.0	0.599999E+00
0.480001E+01	4	9	3	1	0	0	0	0.0	0.953674E-06
0.500002E+01	0	3	1	1	0	0	0	0.0	0.200010E+00
0.500002E+01	0	17	0	10	0	0	0	0.0	0.0
0.500002E+01	0	4	0	10	0	0	0	0.0	0.0
0.500002E+01	0	2	0	10	0	0	0	0.0	0.0
0.540001E+01	1	8	2	1	0	0	0	0.0	0.399989E+00
0.560001E+01	3	10	3	1	0	0	0	0.0	0.200000E+00
0.600000E+01	4	14	6	1	0	0	0	0.0	0.399994E+00
0.600001E+01	2	7	2	1	0	0	0	0.0	0.572205E-05
0.600001E+01	0	13	0	10	0	0	0	0.0	0.0
0.600002E+01	0	3	1	1	0	0	0	0.0	0.114441E-04
0.600002E+01	0	17	0	10	0	0	0	0.0	0.0
0.600002E+01	0	4	0	10	0	0	0	0.0	0.0
0.600002E+01	0	2	0	10	0	0	0	0.0	0.0
0.640001E+01	4	9	3	1	0	0	0	0.0	0.399988E+00
0.660001E+01	1	8	2	1	0	0	0	0.0	0.200000E+00
0.700002E+01	0	3	1	1	0	0	0	0.0	0.400012E+00
0.700002E+01	0	4	0	10	0	0	0	0.0	0.0
0.700002E+01	0	2	0	10	0	0	0	0.0	0.0
0.720000E+01	3	10	3	1	0	0	0	0.0	0.199987E+00
0.720001E+01	2	7	2	1	0	0	0	0.0	0.953674E-05
0.780001E+01	1	8	2	1	0	0	0	0.0	0.599999E+00
0.800001E+01	4	9	3	1	0	0	0	0.0	0.200000E+00
0.800002E+01	1	12	5	1	0	0	0	0.0	0.381470E-05
0.800003E+01	0	3	1	1	0	0	0	0.0	0.953674E-05
0.800003E+01	0	11	0	10	0	0	0	0.0	0.0
0.800004E+01	0	4	0	10	0	0	0	0.0	0.953674E-05
0.800004E+01	0	2	0	10	0	0	0	0.0	0.0
0.840001E+01	2	7	2	1	0	0	0	0.0	0.399977E+00
0.880003E+01	3	10	3	1	0	0	0	0.0	0.400013E+00
0.900001E+01	1	8	2	1	0	0	0	0.0	0.199986E+00
0.900004E+01	0	3	1	1	0	0	0	0.0	0.238419E-04
0.900004E+01	0	6	0	10	0	0	0	0.0	0.0
0.900004E+01	0	2	0	10	0	0	0	0.0	0.0
0.960001E+01	2	7	2	1	0	0	0	0.0	0.599976E+00
0.960003E+01	4	9	3	1	0	0	0	0.0	0.133514E-04
0.100000E+02	0	3	1	1	0	0	0	0.0	0.400011E+00
0.100000E+02	0	17	0	10	0	0	0	0.0	0.0
0.100000E+02	0	4	0	10	0	0	0	0.0	0.0
0.100000E+02	0	2	0	10	0	0	0	0.0	0.0
0.102000E+02	1	8	2	1	0	0	0	0.0	0.199975E+00
0.104000E+02	3	10	3	1	0	0	0	0.0	0.200013E+00
0.108000E+02	2	7	2	1	0	0	0	0.0	0.399986E+00
0.110000E+02	0	3	1	1	0	0	0	0.0	0.200026E+00
0.110000E+02	0	17	0	10	0	0	0	0.0	0.0
0.110000E+02	0	17	0	10	0	0	0	0.0	0.0
0.110000E+02	0	17	0	10	0	0	0	0.0	0.0
0.110000E+02	0	17	0	10	0	0	0	0.0	0.0
0.110000E+02	0	17	0	10	0	0	0	0.0	0.0
0.110000E+02	0	17	0	10	0	0	0	0.0	0.0
0.110000E+02	0	17	0	10	0	0	0	0.0	0.0

.
.
.
.
.

ORIGINAL PAGE IS
OF POOR QUALITY

NODE 3 MAY BE TRAPPED IN AN INFINITE LOOP.
THERE ARE 100 SIMULTANEOUS ARC COMPLETIONS AT TIME 11.00

ERROR EXIT, TYPE 10 ERROR.

FILE STATUS AT TIME 0.1100E+02
P=PREDECESSOR POINTER S=SUCCESSOR POINTER

FILE 1

CELL= 239 P= 99999 S= 211
JTRIB 9 3 1 0 4 0
ATRIB 0.1120E+02 0.0 0.0 0.0 0.0 0.0

Other examples of complex node modifications are presented in Figure 2.2.32. It is left to the reader to study them and find out in which way they are different.

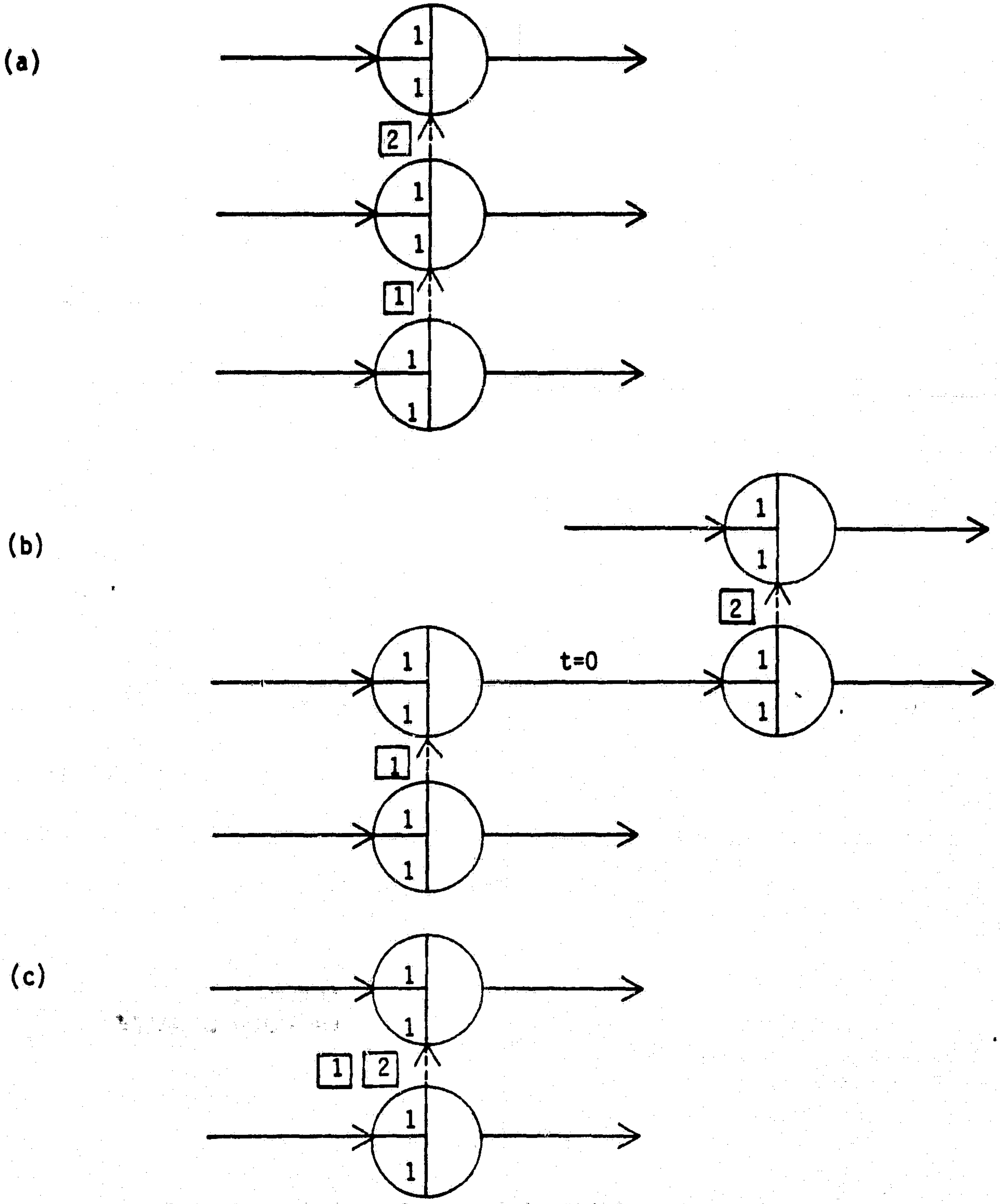


Figure 2.2.32 More Examples on Node Modifications

2.2.5 Statistics Collection

In developing a GRASP network, the modeler includes, directly on the network, nodes for collecting data. Such nodes are referred to as statistics nodes. Any sink node is automatically a statistics node and any node, except a source node, may be designated as a statistics node. They behave exactly as other nodes and the information they collect refers to a time that a specified node is released. Several types of variables can be defined using the release times of a node. Figure 2.2.33 shows a general statistics node, and T denotes the type of statistics to be collected.

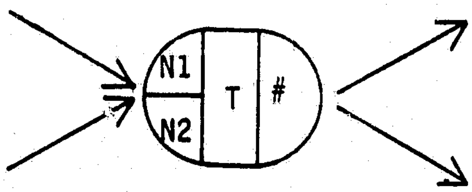


Figure 2.2.33

T can be one of the following:

F - First statistics: record statistics on time and cost at first release of node only

A - All statistics: record statistics on time and cost at all releases of node

B - Between statistics: record statistics on time interval and cost between releases of the node. The time of first release is used only as a reference point for the first value of the time between releases

D - Delay statistics: this statistics relates to the nodes at which activity completions are accumulated, that is, nodes for which the number of incoming activity completions (N_1, N_2) is greater than one. The delay time is the time interval from the first completion of an incoming arc until the node is released. A delay time is computed each time a node is released.

C - Accumulator statistics: like "D" statistics, except time and cost intervals are accumulated in master time and cost accumulators (for "C" nodes, it is required that N1 = N2 = 2). Arcs into C-nodes may not have count types, and C-nodes may not be sink nodes.

M - Mark node: no statistics are collected; every activity which leaves the node is "marked" with the current time and cost

I - Interval statistics: record statistics on the interval between the current time and cost and the "marked" by mark nodes on the incoming activities

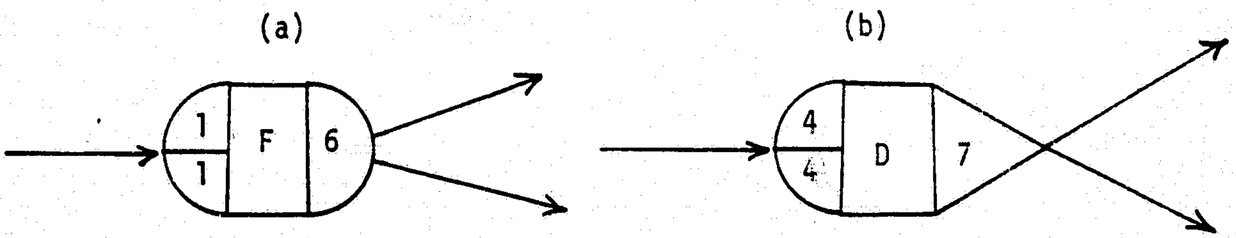


Figure 2.2.34 Examples of Statistics Nodes

As an example of how statistics are maintained by GRASP, consider Figure 2.2.34 (a). Node 6 collects FIRST or "F" statistics. The first time node 6 is released in a simulation, the time and the cost at the time of release are recorded. After the first release, no further data is recorded, but the node continues to function normally. So, one simulation run causes one pair of data points to be recorded. Hence, by repeating the simulation, say for 1000 runs, 1000 pairs of data points are recorded. The program will automatically compute the mean, standard deviation, minimum and maximum for the observed data. In addition, histograms of time and cost to first release of node 6 will be prepared.

Figure 2.2.34 (b) is an example of a node which collects DELAY statistics. In this case, the statistic recorded is the time (and cost) interval between the time of the first activity completion at node 7 and the time (cost) when the node is released. Since node 7 might be released many times in a simulation run, it is possible that many data points may be recorded during each run. If the simulation is repeated a number of times, all observations from all runs are utilized to compute the mean, etc., and to form the histogram. A restriction on Delay (and Accumulator) nodes is that arcs into these nodes may not have count types.

Accumulator or "C" nodes are a special kind of Delay node. For Time statistics, C-nodes collect the same statistics as Delay nodes. An additional restriction for C-nodes is that N1 = N2 = 2. The first input will always activate time (cost) accumulation and the second will terminate collection for that collection period. In addition, however, GRASP will maintain a master C-node accumulator to which all Time intervals collected by all C-nodes will be added. Thus, each simulation run will result in one datum being recorded in the master Time accumulator. By performing repeated runs, a mean, standard deviation, and histogram of these accumulated Times will be produced. Thus, C-nodes permit "adding up" all of the individual data points recorded by a Delay node. This corresponds to accumulated costs. The Time statistics recorded and reported at a C-node are identical to those that would be produced by a Delay node.

C-nodes also permit accumulation of costs but in a somewhat different manner. In this case, the CN parameter on the arcs is used. If a C-node is associated with an arc by the CN parameter, then that C-node will collect the cost of that activity each time it completes. A C-node can be associated with more than one activity in the network. Note that the cost feature of the C-node is not keyed by the arcs that are incident to the C-node. The cost feature is related only to the CN parameters. The statistics produced by an individual C-node will be the mean, standard deviation, and histogram of the costs of all arcs that list that C-node in their CN parameter.

Also, a master C-node accumulator for cost is maintained. The master cost assumulator will maintain a total of all costs recorded by all C-nodes in the network. Hence, each simulation run records one datum in the master cost accumulator, and multiple runs will yield a mean, standard deviation and histogram of accumulated costs.

It is possible to specify two target values called, T2 and C2, which are upper limits for the master time and cost accumulators, respectively. GRASP will compare the values of the master accumulators with the T2 and C2 values. At the end of the simulation, a message will be printed indicating the number of times these values were exceeded. An option permits the exceeding of the T2 and/or C2 values to be considered a system failure.

Hence, excessive accumulated down time or cost can be defined as a system failure.

ALL and BETWEEN statistics are similar to FIRST and DELAY statistics. MARK nodes and INTERVAL statistics require some additional comments, however. If a node is a mark node, then when the node is released, each activity which leaves the node is "marked" with the release time (and cost). These activities carry this mark with them throughout the network. For example, if a marked activity ends at a node and releases it, all activities scheduled from the end of node carry the same mark. If a marked activity passes through another mark node, the old mark is destroyed and a new one implanted.

When a marked activity arrives at a node that collects INTERVAL statistics, the time (and cost) interval from the mark to the I-node is computed and recorded. The mark is not destroyed by this process. In this way, it is possible to measure the time (and cost) for an activity to traverse a given portion of a network.

2.3 State Transition Diagram Analysis

State transition diagram analysis goes far beyond an interpretation of simple block diagram representation. It provides a complete interpretation of the entire system, and can be used to represent most probabilistic models. It also provides a systematic way for describing the behavior of a probabilistic system, under various constraints and assumptions, and also for checking the validity of any model. This section will explain some of the basic logic behind GRASP and illustrate how a GRASP network can be described by a state transition diagram. This section may be skipped without a loss of continuity in understanding the remaining sections. However, it shows an interesting facet of the subject matter which might otherwise be overlooked.

The following description of a system in terms of its states and transitions may exhibit technical abstraction in the beginning, but it also possesses a strong intuitive appeal. A state transition diagram is a directed network where a node represents a state and an arc represents a transition from the state at its origin node to the state at its end node. We first need to look at the states and transitions of a single component.

Assume that whatever the conditions in the system might represent, a component can be in one of the following states:

S_1 = the component is operating

S_2 = the component is under repair

S_3 = the component is idle waiting for operation

S_4 = the component is idle waiting for repair.

S_1 or S_2 are active states and S_3 or S_4 are inactive (idle states). These states are sufficient for most modeling purposes and even some of these may be inadmissible in particular cases. For instance, if we assume that a component goes into repair as soon as it fails and is put back into operation as soon as it is repaired, then the idle states S_3 and S_4 are not needed. The time during which a component remains in one state, whether it is operating (S_1), under repair (S_2), waiting for operation (S_3) or waiting for repair (S_4), can be a fixed constant or a random variable specified by a particular probability law. When a component exhibits states S_1 and S_2 only, the state transition diagram is a very simple network (Figure 2.3.1):

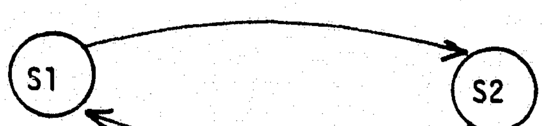


Figure 2.3.1 State Transition Diagram of a Component with States S1 and S2 Only

The GRASP network corresponding to Figure 2.3.1 is also trivial. Figure 2.3.2 gives four possible ways to draw a GRASP network equivalent to the state transition diagram in Figure 2.3.1, depending upon how we want to stop the simulation run. In Figure 2.3.1 (a) the simulation run ends when

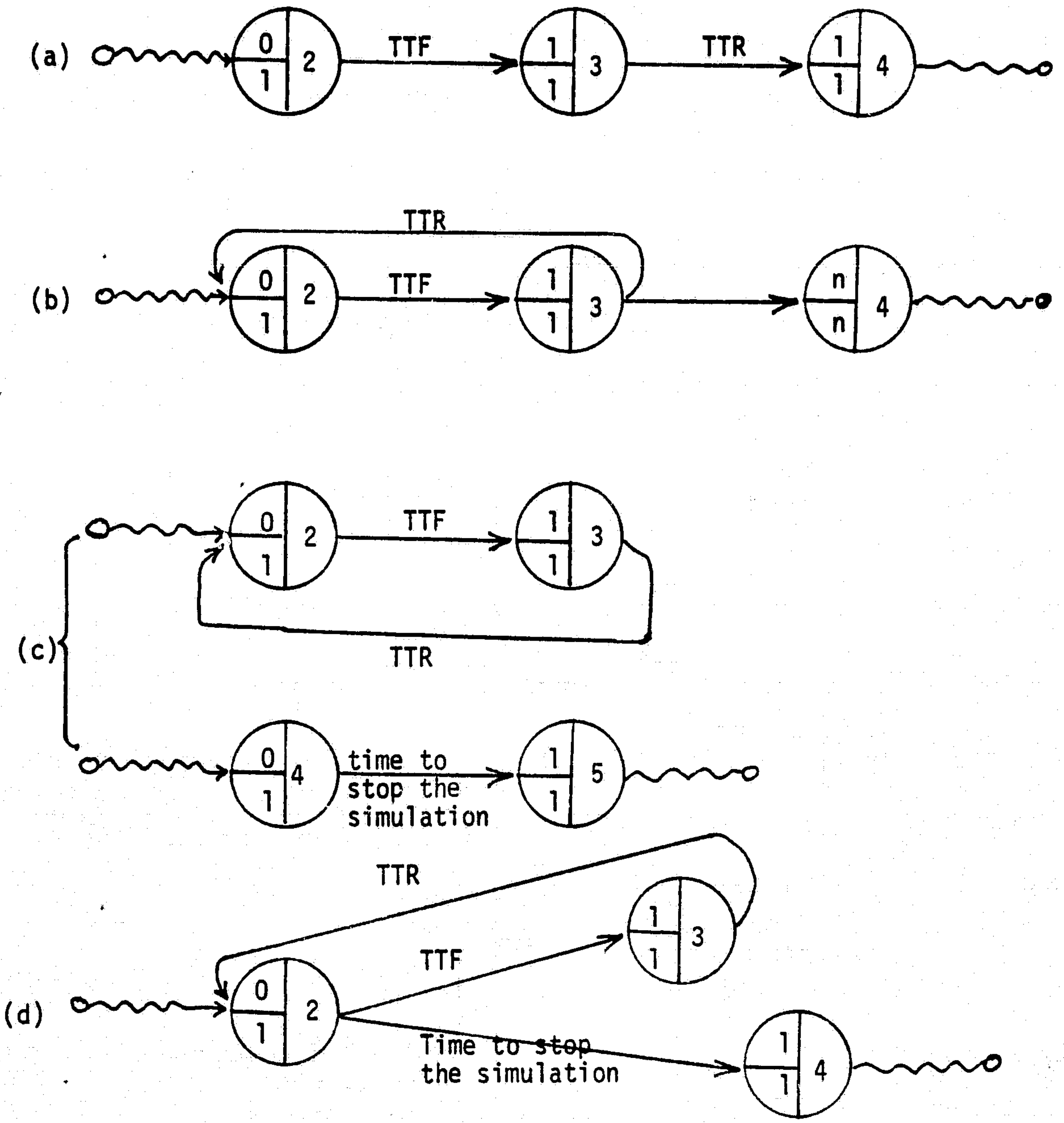


Figure 2.3.2 GRASP Networks Corresponding to the State Transition Diagram in Figure 2.3.1

the component gets repaired. If we want to collect meaningful statistics about failure times and repair times we need to make several runs by specifying multiple runs in the input data.

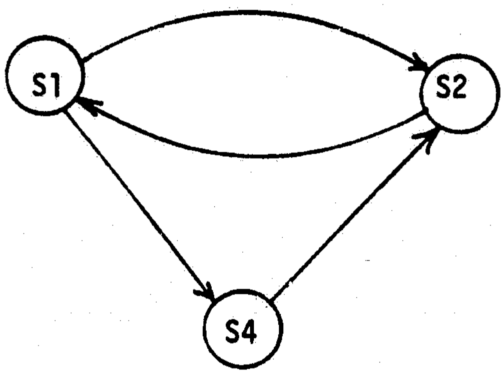
In Figure 2.3.1 (b) we can make only one run and still achieve the same results as in Figure 2.3.1 (a) by specifying the number of times we have a failure by the release counter at node 4.

In Figure 2.3.1 (c) we stop the simulation at a specified time T, by defining another source (4) and a sink node (5); the arc (4,5) has a length T.

Figure 2.3.1 (d) is basically the same as Figure 2.3.1 (c). Other termination criteria can similarly be modeled.

To provide an example of a situation in which a component can be in states S3 or S4, let us look at the case when a repairman, spare parts, or any other resource needed for repair may not be available, or sometimes the component needs to be shipped to a distant repair facility. Similarly, when the component is repaired, putting it back into operation may involve many activities, and in these cases time (and cost) consuming arcs need to be added to the network and considered in collecting the statistics of interest.

Consider another example involving only one component. Assume that when the component fails it goes immediately into repair if the repairman is not busy at that time. If the repairman is busy with another task, then we must wait until he finishes. Assume that we know the probability that the repairman will be busy at any given time. The state transition diagram is shown in Figure 2.3.3.



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 2.3.3 State Transition Diagram for One Component With Repair and Possibly a Waiting Time Before Going into Repair

The transition from S_1 to S_2 means that the component goes into repair as soon as it fails because the repairman is available at that time. Transition from S_1 to S_4 means that the component has to wait for repair, the repairman being busy. Transition from S_4 to S_2 means that the component goes directly to repair and this occurs only when the repairman becomes available. Transition from S_2 to S_1 means that the component is repaired and starts operating. Notice that transitions from S_4 to S_1 and from S_2 to S_4 are not admissible, and they represent the trivial assumptions that a component does not stop operating unless it fails and that the repairman does not stop working until the component is repaired. Although these assumptions are not explicitly stated at the beginning, they appear clearly in the state transition diagram. Now let us look at the corresponding GRASP representation. If p, the probability of the repairman being busy is known, and the distribution of the waiting time is also known, then we can obtain the equivalent network in Figure 2.3.4. Sink nodes are not represented because of the various termination criteria one can use. They are similar to the ones already considered in the previous case.

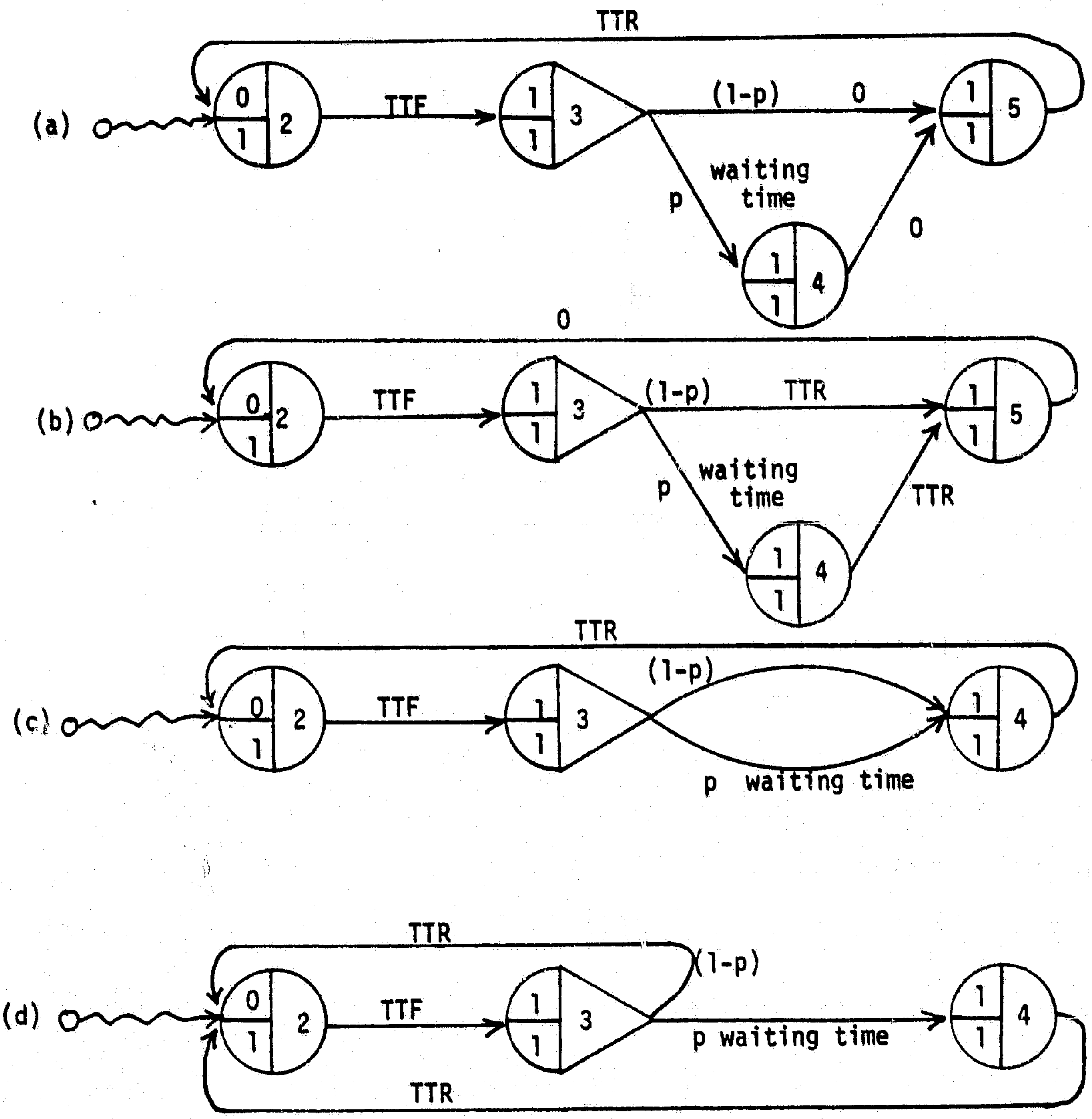


Figure 2.3.4 Equivalent GRASP Diagrams for a Component with Repair and Possibly a Waiting Time before Going into Repair

In Figure 2.3.4 (a) node 2 is a source node and arc (2,3) represents the time until failure. Node 3 has a probabilistic output: arc (3,4) occurs with probability p (i.e., probability that the repairman is busy) and its duration is the waiting time. Arc (3,5) occurs with probability (1-p) and has no duration. Arc (4,5) occurs when the repairman becomes available and also has no duration. Node 5 is released when at least one of the arcs (3,5) or (4,5) occurs, and in this case only one of them can occur because of the nature of node 3. Arc (5,2) represents the time to repair the component.

Figure 2.3.4 (b) differs from Figure 2.3.4 (a) in the duration of the arcs (3,5), (4,5) and (5,2) as shown in the network.

Figure 2.3.4 (c) and Figure 2.3.4 (d) are simplifications of Figures 2.3.4 (a) and (b), respectively, and they are easy to follow. Since the size of the network depends upon the number of nodes, it is better to select the equivalent models with a minimum number of nodes.

If the probability that the repairman is busy and the waiting time distributions are not explicitly known, then by including in the model all the activities in which the repairman is involved we obtain the model in Figure 2.3.5. However, this network might be only a single part of a larger network representing all the activities that may keep the repairman busy when he is needed.

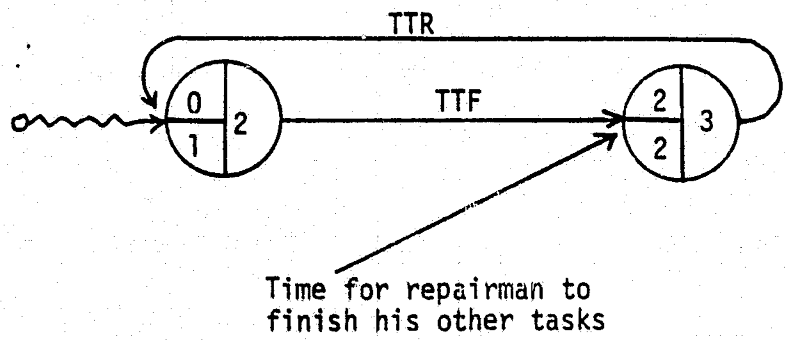


Figure 2.3.5 One Component with Repair, and Possibly Waiting for Repair if the Repairman is not Available

When a system has more than one component, its states are a function of the states of its individual components and the transitions between the states depend on the structural configuration of the components. For a system with two components A and B, a state of the system is characterized by the couple state (S_A, S_B), where S_A is the state of component A and S_B is the state of component B. Without any confusion we can obtain the state of the system as merely the couple (S_A, S_B). One need not distinguish between (S_A, S_B) and (S_B, S_A). If A can be in 2 states A_1 (operating) or A_2 (under repair), and B can similarly be in B_1 (operating) and B_2 (under repair), then the system can have the following states:

(A_1, B_1) = both A and B are operating

(A_1, B_2) = A is operating and B is under repair

(A_2, B_1) = A is under repair and B is operating

(A_2, B_2) = Both A and B are under repair.

The actual meaning or interpretation of these states depends on the logical configuration of the system (parallel, series, standby). The transitions between these states occur whenever the state of at least one component changes. However, in practically all real life situations, the probability that two or more changes in the states of the components occur at the same point in time is negligible. For instance, in a system involving two components, it is assumed that the components cannot fail simultaneously or get repaired simultaneously; all these events must be different. This is particularly true when life time and repair time distributions are continuous. For this reason, it is one of the major assumptions in the design of the GRASP program that events do not occur exactly simultaneously.

Some of the GRASP models in the next section may not be valid if the assumption above does not hold. If we are dealing with discrete probability distributions, or if for any reason it is suspected that the probability that two or more components change their states at the same time, then these combination events would need to be explicitly considered. However, with a little more effort it is not difficult to extend these models to handle such situations.

SECTION 3
GRASP MODELS OF COMMON RELIABILITY SYSTEMS

Several network examples which illustrate the capabilities of GRASP will now be presented and explained. Further documentation related to data input and output from the GRASP simulations will be presented in Section 4. Commonly encountered configurations are first presented. Depending upon the underlying assumptions and the type of statistics that need to be collected, several different GRASP models can be built from the same reliability block diagram. For some examples there may be several equivalent GRASP networks, in which case the model with the least number of nodes will be selected. The state transition approach will be used to describe some of the models. Once the networks shown in this section are understood, they can be considered as standard GRASP networks that can be utilized directly in modeling more complex systems. Models for simple examples with no repair and replacement are first presented. These simple models are then progressively extended to other familiar reliability configurations and the state transition diagram analysis will prove to be very helpful in describing and validating the GRASP models.

3.1 Simple Systems with No Repair or Replacement

Case 1: n Units in Series with Known Failure Probabilities

No failure distributions are needed because during the given mission time the failure probabilities are known. Figure 3.1.1 (a) shows the reliability block diagram and is self-explanatory. Part (b) shows the corresponding GRASP network.

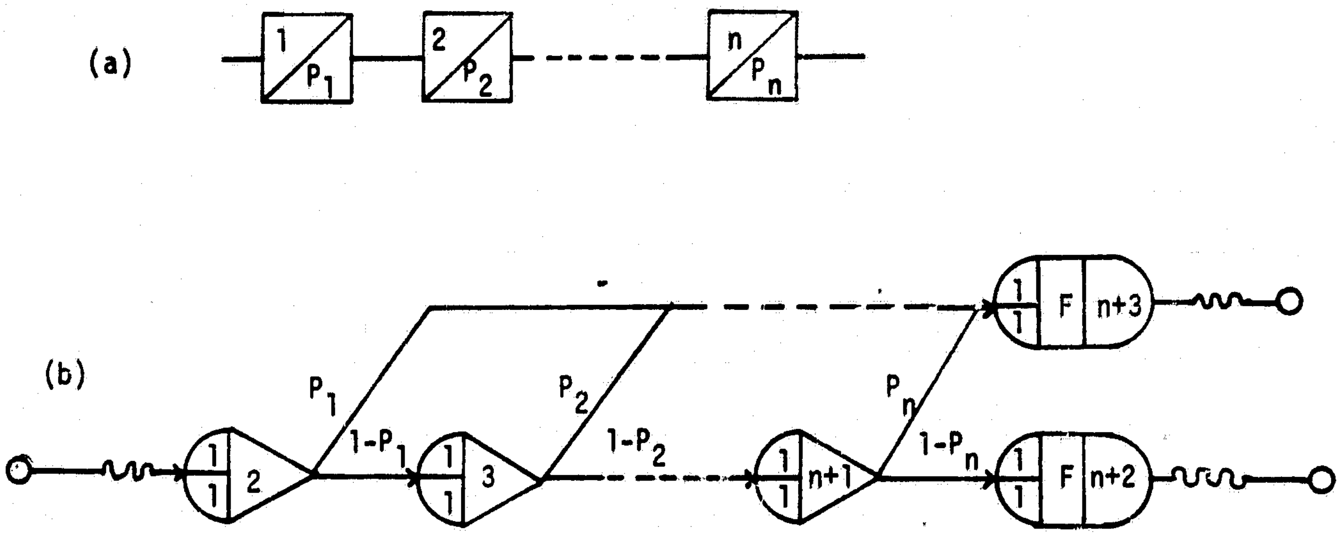


Figure 3.1.1 n Simple Units in Series

Each of the Nodes 2, 3, ..., n+1 corresponds to a component; Node 2 is also a source node. Notice the probabilistic output corresponding to whether a unit fails or not. There are two sink nodes. A simulation run that ends at node (n+2) indicates that the system did not fail. If termination is at node (n+3), it indicates that the system failed because at least one unit failed.

The duration of every arc is taken to be zero and the model is independent of what types of statistics are to be collected at the sink nodes. First statistics are shown in the figure for completeness only. By specifying a large number of runs, we can determine the reliability of the system as being the ratio of the number of times node (n+2) was released to the number of runs. Figure 3.1.2 shows another equivalent GRASP network with n source nodes and 2 sink nodes.

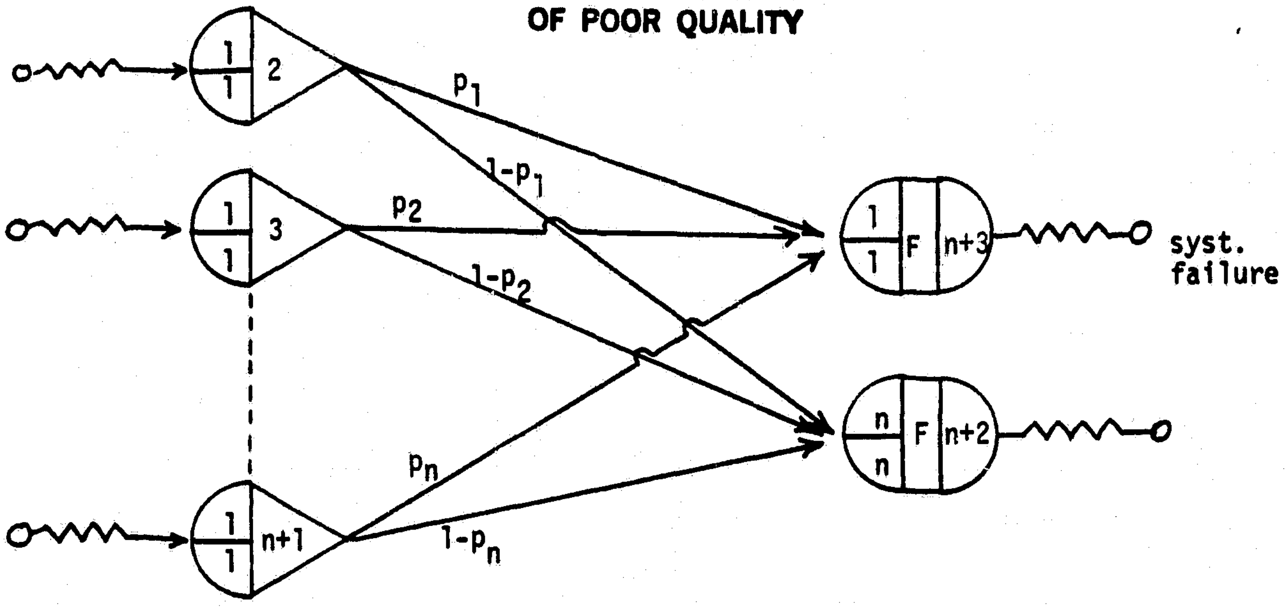


Figure 3.1.2 Another GRASP Network for
n Simple Units in Series

It is obvious that a better way to determine reliability, R, of such a system is to analytically compute it directly as $R = 1 - \prod_{i=1}^{n} (1-R_i) = 1 - \prod_{i=1}^{n} p_i$, R_i being the reliability of unit i. The purpose of this example is to illustrate the GRASP modeling concepts.

Case 2: n Units in Series with Random Failure Times

All units start operating at the same time and the system fails if any one unit fails. The time between failures for each unit is represented by an arc from node 2 to node 3 in Figure 3.1.3. The duration of the arcs are generated from the failure time distributions of the units.

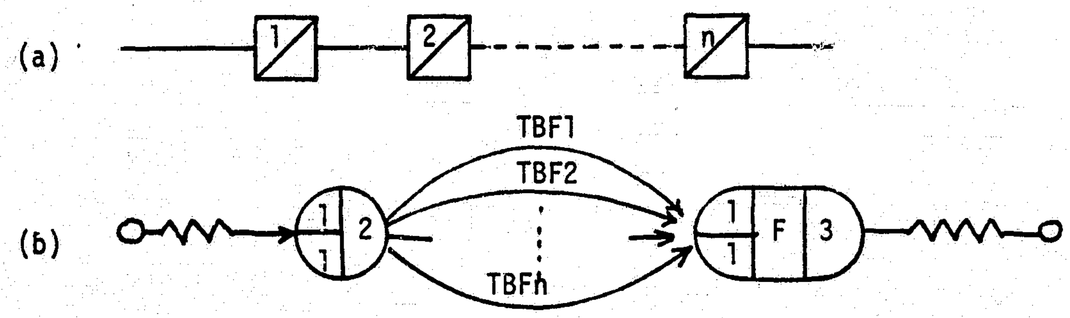


Figure 3.1.3 n Units in Series with Sampled Failure Times

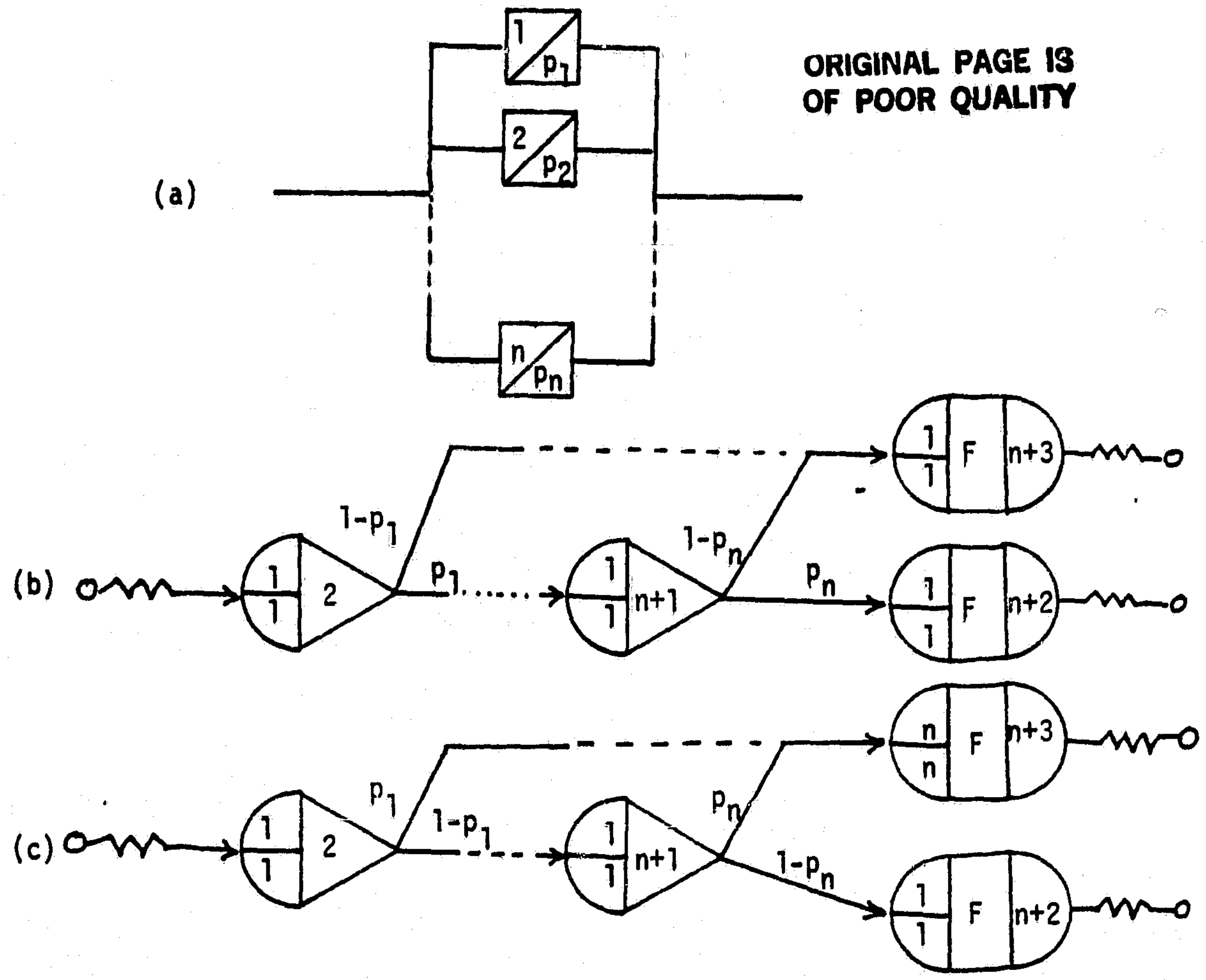


Figure 3.1.4 n Simple Units in Parallel

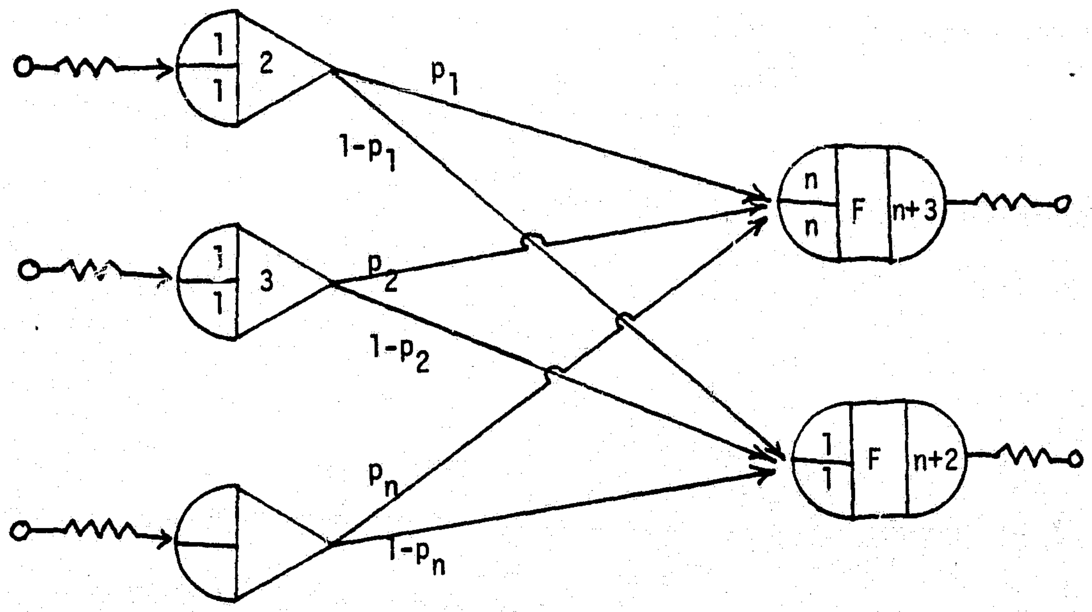


Figure 3.1.5 Another Network for n Simple Units in Parallel

Case 3: n Units in Parallel with Known Failure Probabilities

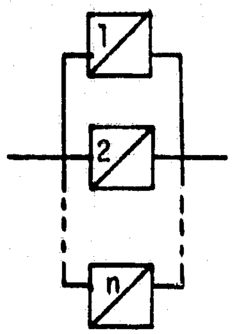
The system fails if all units fail. This case is similar to case 1. The only difference between the GRASP networks in Figures 3.1.1 (b) and 3.1.4 (b) is in the way the probabilities are assigned to the output sides of nodes 2, 3, ..., n+1, and in the meaning of nodes (n+2) and (n+3). Apart from the probabilities assigned to the arcs, the network looks exactly the same as in Figure 3.1.1 (b) for the first case. When node (n+3) is released, this means that the system did not fail. Node (n+2) is released only if all units fail and this indicates a system failure. Figure 3.1.4 (c) shows another GRASP network that is also very similar to Figure 3.1.1 (b) and Figure 3.1.4 (b). NOdes (n+2) and (n+3) play the same role as in Figure 3.1.1 (b) (i.e., reliability estimation is based on the number of releases of node (n+2)). The probabilities on the output sides of nodes 2, 3, ..., n+1 remain the same as in Figure 3.1.1 (b). The release counter of node (n+3) is modified such that the node is released only if we have n incoming arc completions, which correspond to a system failure.

Another equivalent network can also be obtained by interchanging nodes (n+2) and (n+3) in the network shown in Figure 3.1.2. The resulting network is shown in Figure 3.1.5. Notice that the only difference compared to Figure 3.1.2 is in the meaning of the releases of nodes (n+2) and (n+3). Other equivalent networks corresponding to cases 1 and 2 can easily be found.

Case 4: n Units in Parallel with Random Failure Times

This case is very similar to case 2. The release counter of node 3 is n, meaning that the system fails only if all units fail. This case is represented in Figure 3.1.6.

(a)



(b)

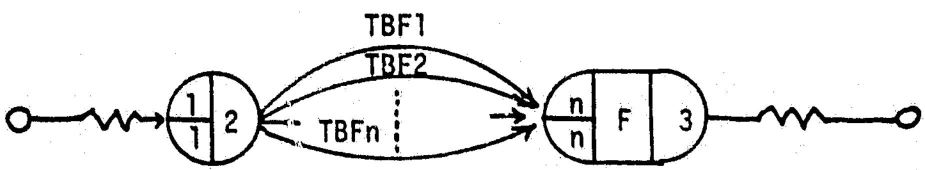


Figure 3.1.6 n Units in Parallel with Sampled Failure Times

Case 5: Example of a k-out-of-n system

A k-out-of-n system is a system that fails only if k units fail. As in the previous examples, we can identify two models depending on whether the probabilities of failure of the components are explicitly known or only the life time distributions are known. The first model is shown in Figure 3.1.7, and the second in Figure 3.1.8. Note that a block diagram for such a system would be very cumbersome, while the GRASP network uses the same number of nodes as in the previous models.

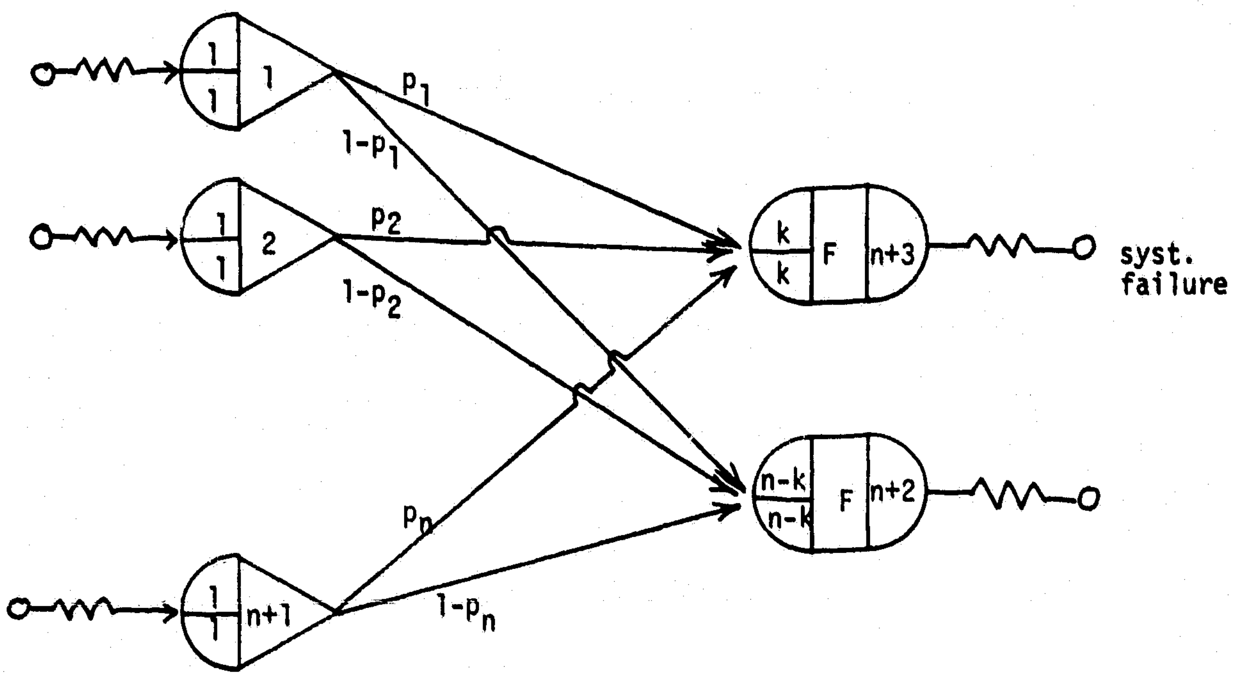


Figure 3.1.7 A k-out-of-n system, Known Probabilities of Failure

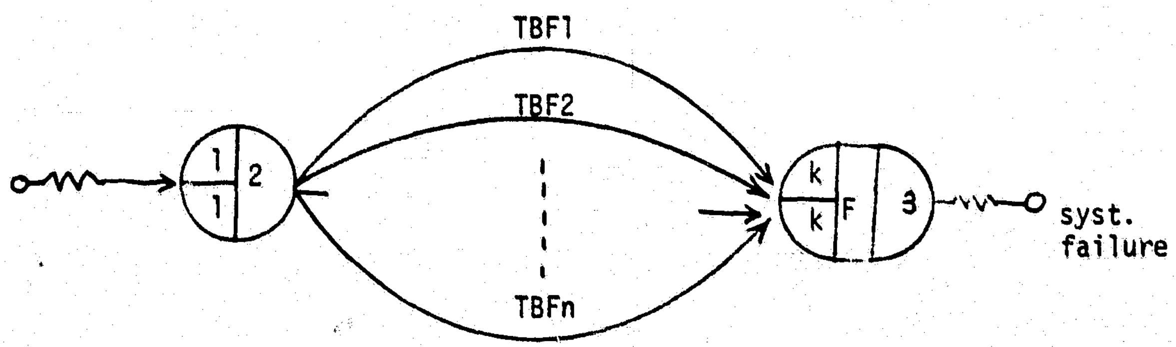


Figure 3.1.8 A k-out-of-n System with Sampled Failure Times

3.2 Simple Systems with Replacement, No Repair

In each of the following examples we assume that one has access to a finite pool of spare units, and that failed units are replaced through a switching mechanism with a probability of failure and/or a delay in its functioning. Failures are also assumed to be independent.

Case 1: One Unit, n Identical Spares and Perfect Switching

If the probability of failure of each unit is known, we obtain the model shown in Figure 3.2.1.

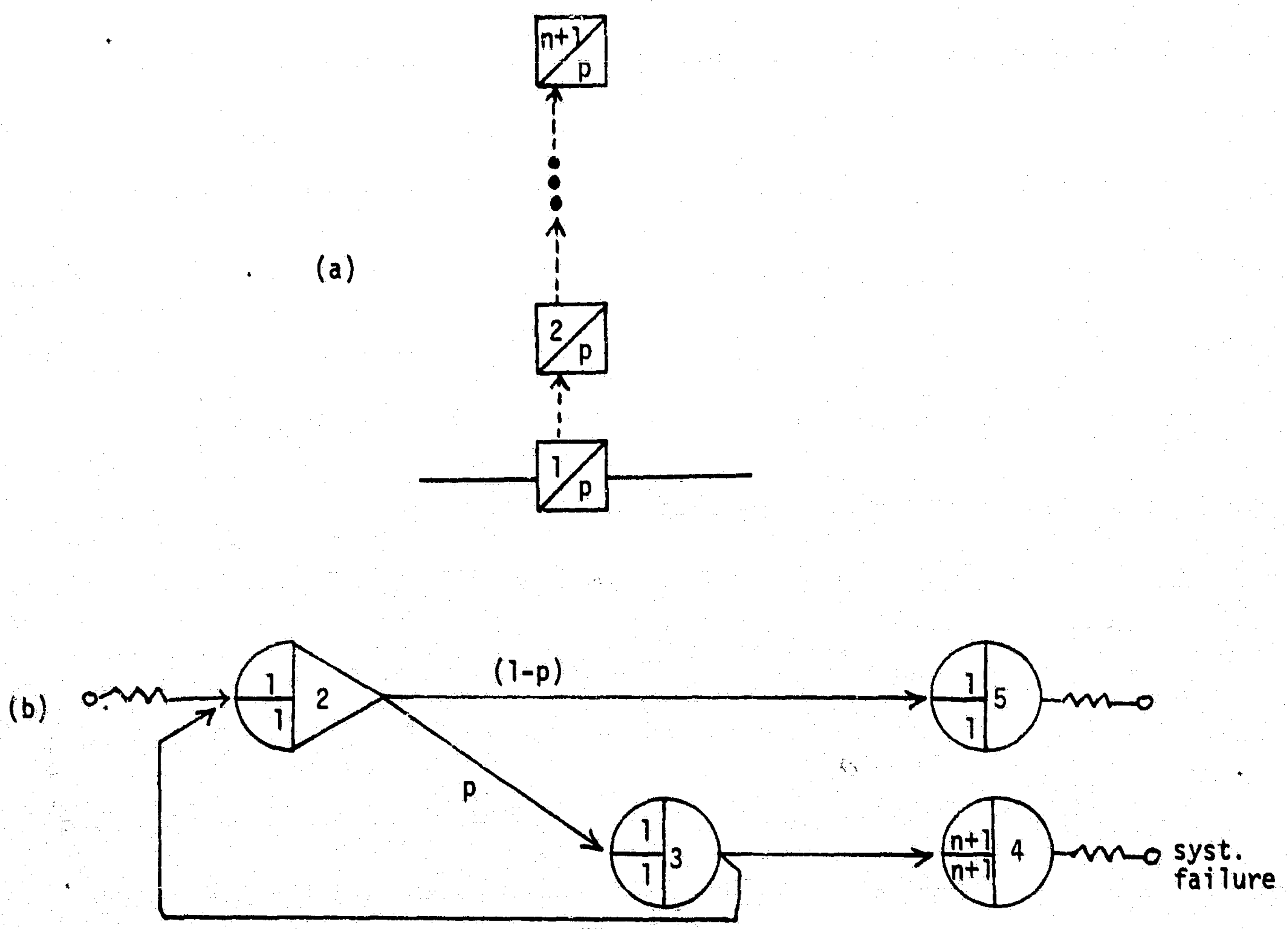


Figure 3.2.1 One Unit With n Spares, Perfect Switching and Known Probabilities of Failure

Let p be the probability of failure of a unit. In the GRASP network, node 5 is released if at least one unit did not fail. We can analytically compute the probability that this will occur; that is $(1 - p^{n+1})$. By performing multiple simulation runs, this result is estimated by the ratio of the number of times node 5 is released to the total number of runs. This result should be close to the theoretical value. Each time a failure occurs, the release counter of node 4 is <u>decremented</u> by one, and this node is released when all units fail (release counter equal to zero).

If we are required to use the failure rate density function of each unit we obtain the equivalent network shown in Figure 3.2.2. The release counter of node 4 in Figure 3.2.2 (a) is initially set to $n+1$ because we have a total of $n+1$ units, and the release counter is decremented each time a unit fails.

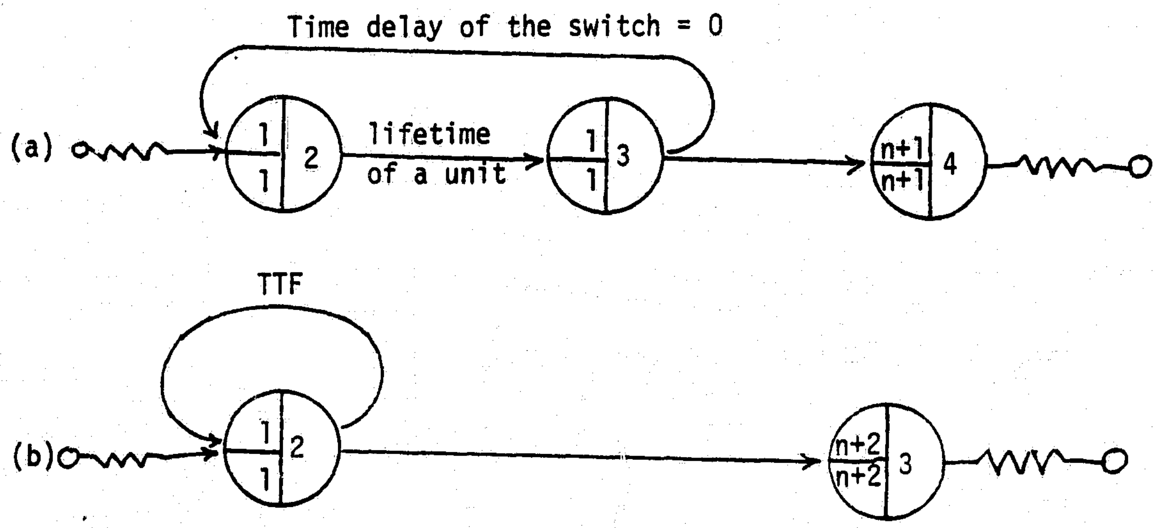


Figure 3.2.2 One Unit with n Spares, Perfect Switching and Known Lifetime Distribution

The duration of arc (2,3) is the lifetime of the unit which is sampled from a known assumed distribution. Arc (3,2) corresponds to the replacement activity and has a zero duration. Arc (3,4) has also a zero duration. We can reduce the number of nodes from 3 to 2 for this model as shown in Figure 3.2.2 (b). Node 3 is now a sink node and its release counter is initially n+2 . The life cycle of a component is represented by the self loop (2,2). Since arc (2,3) has no duration, the released counter at node 3 is decremented to n+1 at time zero. Subsequent decreases of the release counter occur each time a unit fails.

Case 2: One Unit, Identical Spares and Imperfect Switching

The difference in this example and case 1 is that when a unit fails, an automatic switch that executes the replacement can fail. We assume that when a switch fails, another switch corresponding to another unit is triggered; if the new switch fails again, the replacement process is again repeated until the last spare unit is reached (parallel redundancy). Another arbitrary assumption is that all switches are equivalent and that when a switch

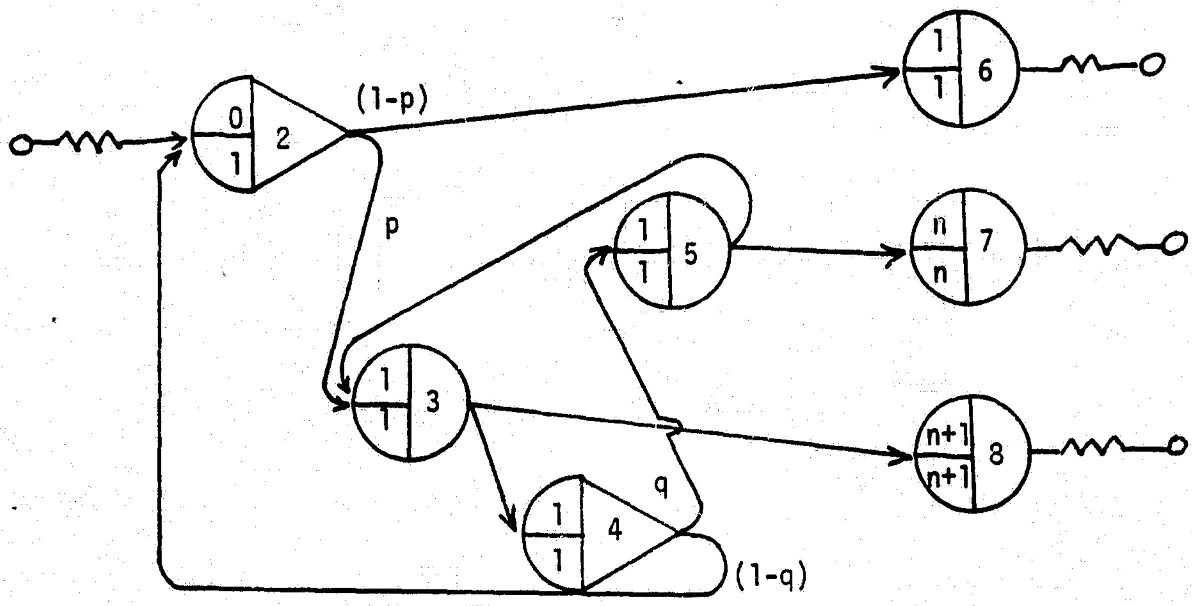


Figure 3.2.3 One Unit, n Spares, Switching Subject to Failure and Shut-off Capability

fails, the corresponding unit cannot be used although it is good. For this case we can identify several models. Figure 3.2.3 shows a network where probabilities of failure are known. p is the probability that a unit fails, and q is the probability that a switch fails. The branching from node 2 corresponds to whether the unit fails or not. All arcs in the network have zero duration. If the unit does not fail, (the probability for this is 1-p) then the branch (2,6) is selected and node 6 is released, indicating that the system performed successfully. If the unit fails, (this occurs with probability p) then arc (2,3) is selected. Node 3 has a deterministic output and it generates arcs (3,4) and (3,8). Node 8 is a sink node that indicates a system failure due to the failure of all units. Its release counter is decreased by one every time the system experiences a failure of one unit. The initial value of its release counter is n+1 because there is a total of n+1 units in the system. Arc (3,4) corresponds to the instantaneous switching mechanism. The probabilistic output at node 4 corresponds to the possibility of failure in the switching mechanism. If the switch fails, arc (4,5) is selected; then node 5 is released and activates arcs (5,3) and (5,7). The purpose of arc (5,3) is to trigger another switch, and the purpose of arc (3,7) is to stop the simulation run when the last switch fails. One is subtracted from the release counter at node 7, which is initially n every time a switch fails, so that the release of node 7 indicates a system failure due to the failure of the first unit and all the switches.

It is easy to see how Figure 3.2.3 can be modified in order to accommodate other assumptions about the system. Let us assume that when a switch fails, the corresponding unit can be activated by another switch. Everything else remains the same. We obtain the network shown in Figure 3.2.4. The difference between the two networks is in the way the release counter at node 8 decreases. In Figure 3.2.3 one is subtracted from the release counter every time a unit or a switch fails, while in Figure 3.2.4 this is done only when a unit fails.

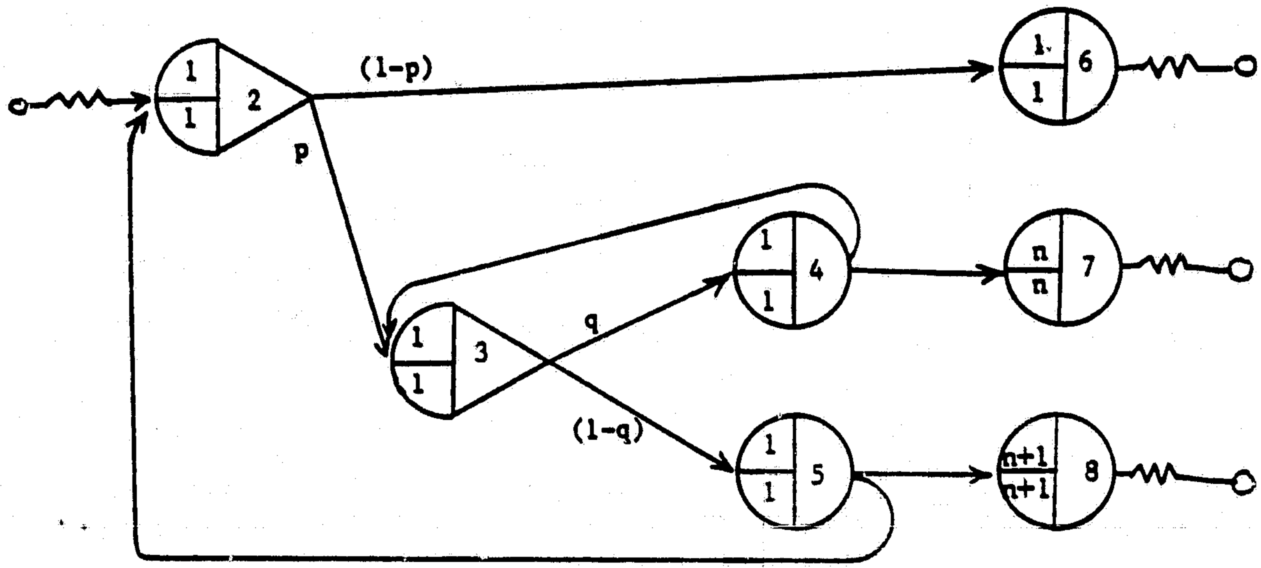


Figure 3.2.4 One Unit, n Spares, Switching Subject to Failure, No Shut Off

Before looking at other examples, let us see how the network would be formed if instead of having the probabilities of failure of the units explicitly specified, we only know their lifetime distribution. This is assumed the same for all units since they are identical. As above, we first construct the network corresponding to the assumption that a switch failure shuts the corresponding unit off. Referring to Figure 3.2.5, arc (2,3) represents the life time of a unit and is the only arc with a non-zero time duration. Upon failure of a unit, automatic replacement is triggered by a switch which has a probability of failure as shown by the probabilistic output of node 3. Notice that one is subtracted from the release counter of node 6 whenever a switch fails, and from the release counter of node 7 whenever a switch <u>or</u> a unit fails. If we want to model the situation where a switch failure does not shut the corresponding unit off, we only need to remove arc (4,7) from the network in Figure 3.2.5 so that the release counter of node 7 is decreased by one only when a unit fails.

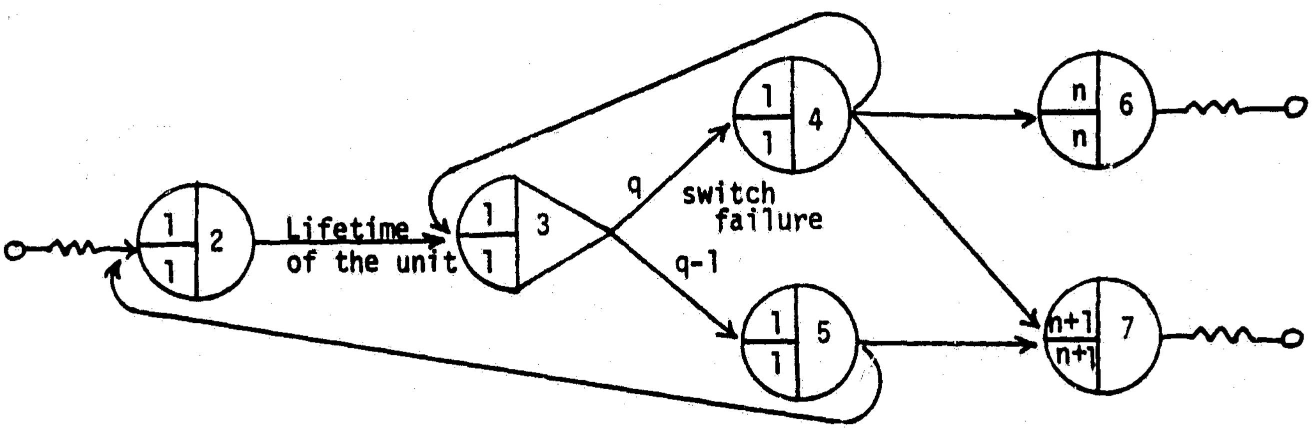


Figure 3.2.5 One Unit, n Spares, Switching is Subject to Failure, Life Time Distribution is Known

Case 3: One Unit, n Identical Spares, Switching with Delay

As before, we first look at the situation where the probabilities of failure are known. The switches are not subject to failure and they are identical (i.e., the distribution of the time delay is the same for all switches). Let p be the probability of failure of a unit. It is easy to see that the GRASP network in Figure 3.2.6 is exactly the same as in the first case, Figure 3.2.1 (b), with arc (2,3) now having a duration which is the time delay of the switch. We can collect statistics on system down time by collecting delay statistics at node 3. The type of statistics to be collected is not shown in the network. The description of the rest of the network is the same as in case 1, and does not need to be repeated. If the failure probabilities are not explicitly given and the life time distribution of the units are known, little needs to be changed from the model in case 1 in order to include the switch delay requirements. The model is shown in Figure 3.2.7

and we can notice that it is basically the same as in Figure 3.2.2 (a).

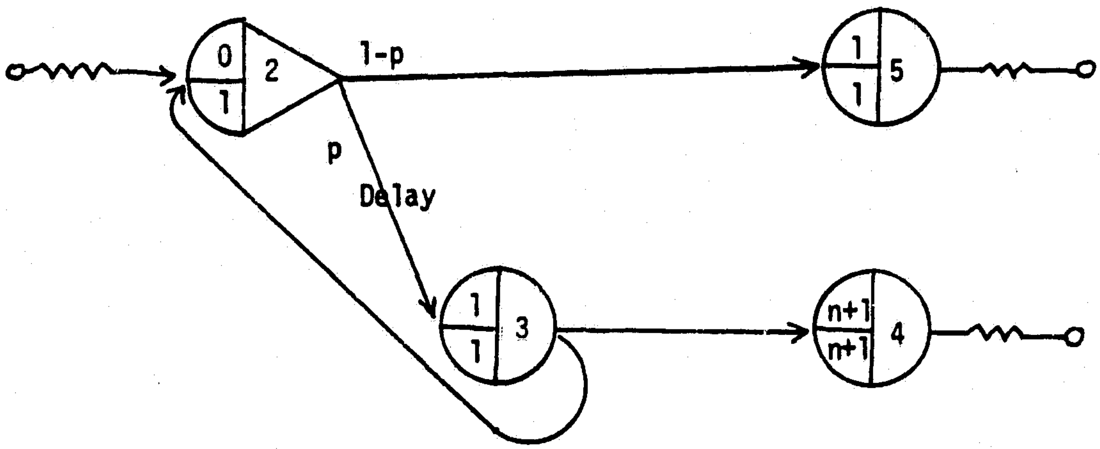


Figure 3.2.6 One Unit, n Spares, Switching with Delay, Known Failure Probabilities

The only difference is that the duration of arc (3,2) is now the time delay of the switching mechanism.

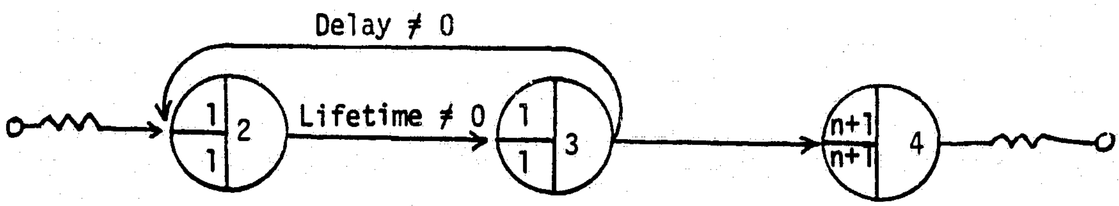


Figure 3.2.7 One Unit, n Spares, Switching with Delay, Known Lifetime Distribution

Notice that in the network of Figure 3.2.7 we can collect statistics on failure times of the units at node 3, but we cannot directly collect statistics on delays at node 2 because this node is already a source node. It is easy to modify the network by adding one node to collect this data if such statistics are needed.

Case 4: One Unit, n Spares, Switching with Delay and Possibly Failure

This case combines the previous two cases. Depending upon the assumptions made, we can identify different models.

If we assume that the failure probabilities are known and that a switch shuts a unit off upon failure, then we obtain the network in Figure 3.2.8, which is the same as that of Figure 3.2.3. The switching delay is specified as the duration of arc (3,4). There are alternate ways to model the delay. For instance we can specify it as the duration of arc (2,3) and (5,3).

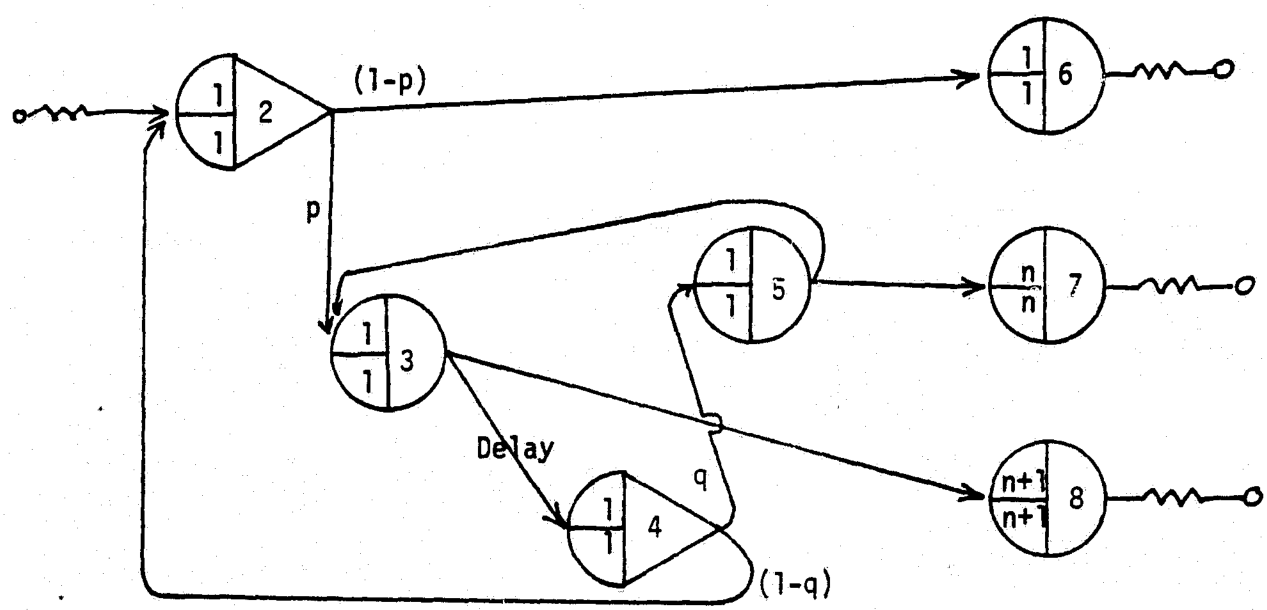


Figure 3.2.8 One Unit, n Spares, Switching Subject to Delay, Possibly a Failure and With Shut-off

If we assume that a switch failure does not shut a unit off, and everything else is as in the previous model, we obtain the network shown in Figure 3.2.11. Notice that it is an extension of the network given in Figure 3.2.4. Arc (3,4) and (3,5) have now non-zero durations and they represent the switching delay. There are also other ways to represent this delay. The interpretation of other nodes and arcs in this network is the same as in the network in Figure 3.2.4.

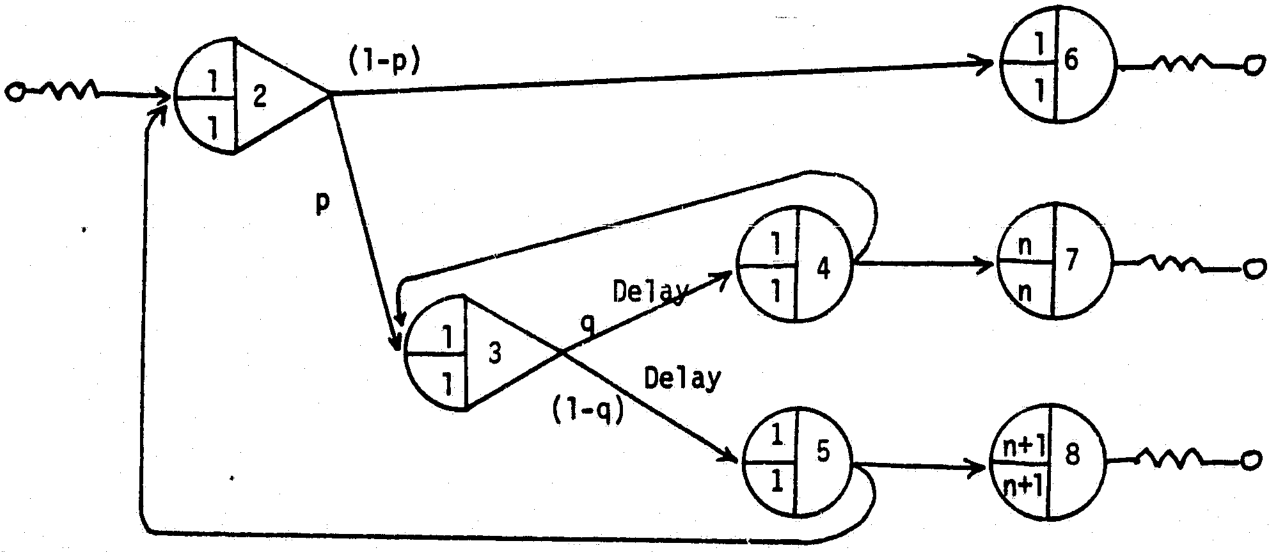


Figure 3.2.11 One Unit, n Spares, Switching Subject to Delay and Failure, No Shut Off

If we are given the lifetime distribution instead of the probabilities of failure, the GRASP network is exactly the same as in Figure 3.2.5. We only need to specify delay density functions for arcs (3,4) and (3,5).

Case 5: Non-identical units

To end this section on simple models with replacement, we now see how to handle the case where the system is composed of non-identical units.

We first look at the situation where the probabilities of failure are known. We will only extend the case 1 example for illustrative purposes. Requirements from cases 2, 3 and 4 can be added to the new model without any problem.

Let p_i be the probability of failure of unit i, i=1, ..., n+1. Since the units are assumed non-identical, the p_i's are not all equal. The laternatives, failure or non-failure of the unit, were represented previously by a single node with a probabilistic output. Since the probabilities of failure now differ, we need one such node for each unit. One way to model such a system is shown in Figure 3.2.12.

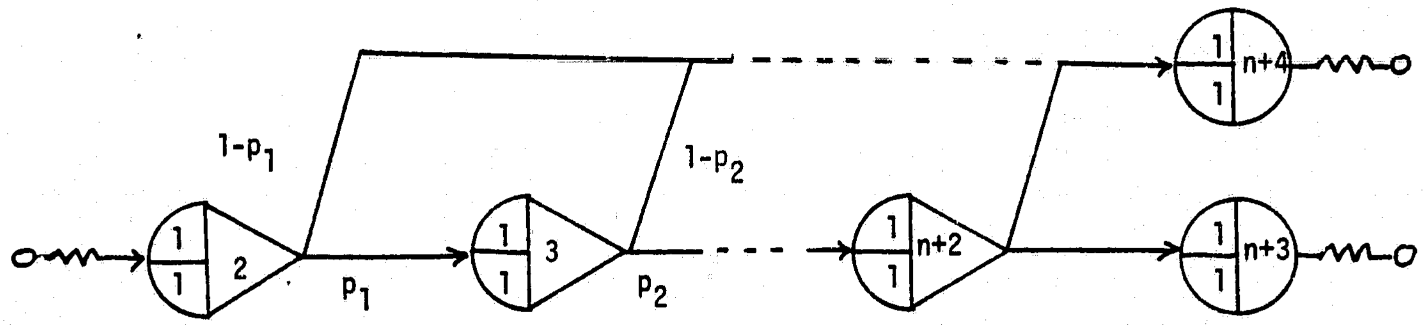


Figure 3.2.12 One Unit, n Non-identical Spares, Perfect Switching, Known Probabilities of Failure

Notice that Figure 3.2.12 is exactly the same as Figure 3.2.4 (b), apart from the fact that we have an extra unit. This is due to the fact that the assumptions made are such that the system behaves exactly like a system where the units are in parallel and are independent. However, for the purpose of eventually including additional requirements in our model, it should be kept in mind that the systems are different. However, they behave exactly in the same manner and that is what the GRASP network shows. The following situation will show this is not true if instead of the probabilities of failure

C-2

ORIGINAL PAGE IS
OF POOR QUALITY

the life time distributions are specified. In a parallel system, all items might fail at the same time and this cannot occur in this model. The GRASP network is straightforward and is shown in Figure 3.2.13. The duration of each arc corresponds to the life time of one unit. If two distributions happen to be the same, the number of nodes in the network can be reduced by replacing one arc by a self-loop. This will cause the node with a loop to be released several times during the simultion. This does not modify the model because the subsequent release of a node will not affect the release of the sink node. As mentioned earlier, the network in Figure 3.2.13 can be extended in a variety of ways to include additional assumptions and requirements about the system.

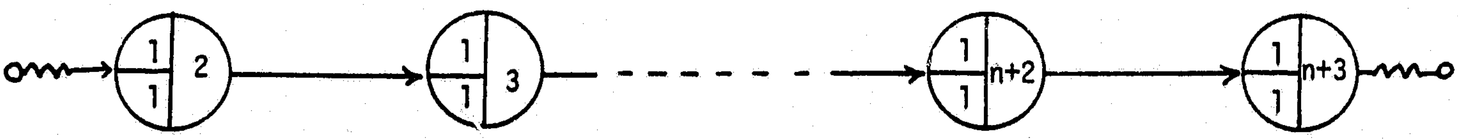


Figure 3.2.13 One Unit, n Non-identical Spares, Perfect Switching, Different Lifetime Distributions

The analysis of systems with more than one unit with replacement is very similar to the analysis of repairable systems with 2 or more units. The GRASP model depends mainly upon the configuration of the system; that is, whether the units are in series, parallel, or standby. Many different models can be constructed because one can make a lot of different assumptions about the same physical configuration, and this parallels the large variety of systems one can encounter in real life situations. In the next section, we present a selection of several such models that are fairly representative.

3.3 Parallel Systems with Repair

We need only look at models of systems with 2 units, since extensions to models of systems with more than 2 units can easily be obtained. From the block diagram shown in Figure 3.3.1, we can construct several GRASP networks depending on the assumptions made. The state transition diagram analysis presented in the previous chapter will be used to validate the models.

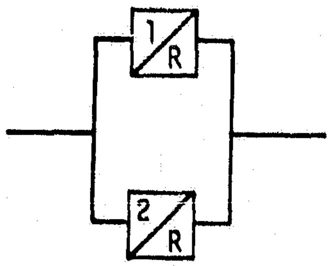


Figure 3.3.1 Block Diagram for Any System With Two Units in Parallel

We assume the following:

- components fail completely independently of each other
- a component goes into repair as soon as it fails, and each component has its own repair facility
- a component starts operating as soon as it is repaired
- a component does not stop operating until it fails.

Before drawing the GRASP network of such a system, it would be helpful to draw the state transition diagram which is an accurate picture of all stated assumptions. Let A and B be the components. Each has only two states; it is either operating or under repair. Let A1, B1 be the operating states and A2, B2 be the repair states. The system can then be in one of the following states:

(A1,B1) = both units are operating

(A1,B2)
(A2,B1) = one unit is operating, the other under repair

System is up

(A2,B2) = both units are under repair; the system is down.

This procedure of representing the state of a system in terms of the states of its components is used in many reliability analysis, and the order of the elements in each couple is irrelevant. For instance, (A1,B2) is the same as (B2,A1). Recall that GRASP assumes that the probability of an occurrence of two or more simultaneous events is negligible. Hence, we can reasonably assume that transitions from (A1,B1) to (A2,B2) or from (A1,B2) to (A2,B1) are not admissible. Figure 3.3.2 shows the state transition diagram corresponding to these assumptions. Notice, also, that transitions from (A2,B2) to (A1,B1), and from (A2,B1) to (A1,B2), are also inadmissible. The equivalent GRASP network is obtained from previous networks corresponding to each unit.

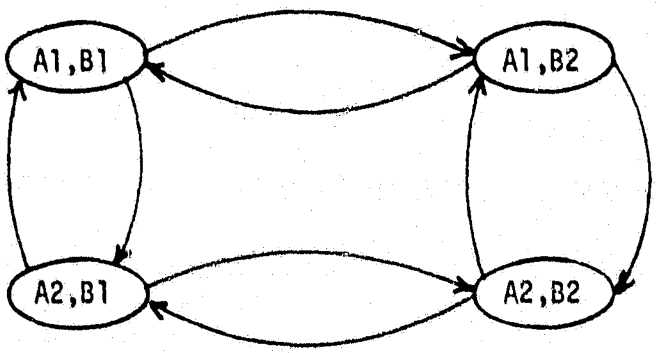


Figure 3.3.2 State Transition Diagram for Two Units in Parallel with Repair

Assume that both units are operating when we start the simulation. Figure 3.3.3 (a) represents unit A. The arcs (2,3) and (3,2) represent the time between failures, and the time between repairs, respectively. Figure 3.3.3 (b) represents Unit B. Arcs (4,5) and (5,4) correspond to the time between failures, and the time between repairs for the second unit. To be consistent with the symbolism introduced earlier, arcs (2,3) and (4,5) are failure arcs and (3,2) and (5,4) are repair arcs with negative arc types (number).

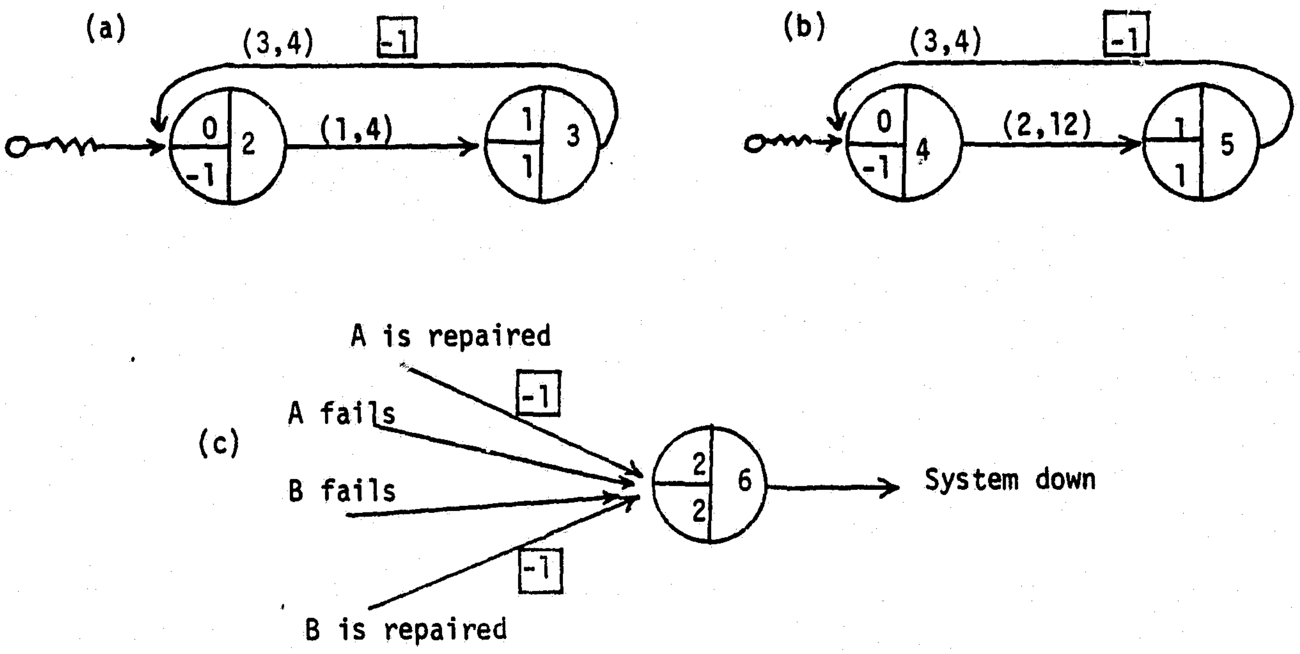


Figure 3.3.3 Decomposition of a GRASP Network for 2 Units in Parallel with Repair

Nodes 3 and 5 are failure nodes and have a <u>positive</u> release counter; nodes 2 and 4 are repair nodes with N2 = -1. Since nodes 2 and 4 are also source nodes, the initial value of their release counter is N1 = 0. The specification table would include the following information for the time to failure and repair time distributions. The TBF for unit one is Erlang (distribution type 4) with parameters in parameter set 1. The TBF for unit 2 is Weibull (type 12) with parameters in parameter set 2. Both repair distributions are Erlang with parameters in set 3. We need to put parts (a) and (b) in Figure 3.3.3 together, and add one more node to indicate when the system is down. This node is shown in Figure 3.3.3 (c). Its release counter indicates how many components must fail in order to define a system failure. Putting all the pieces together, we obtain the complete network of the system as shown in Figure 3.3.4.

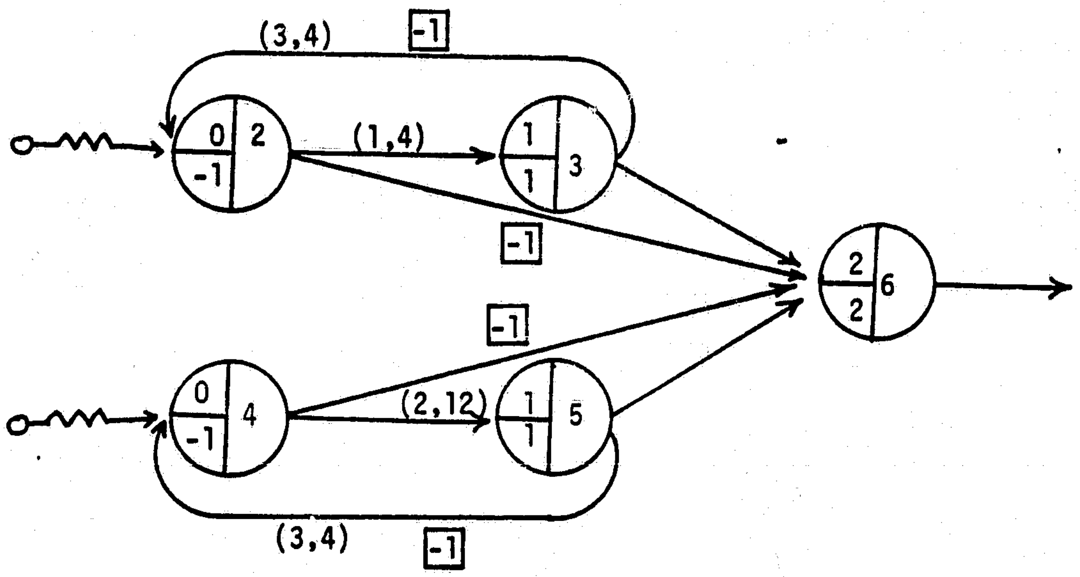


Figure 3.3.4 Two Units in Parallel
with Repair

It is not indicated what type of statistics are to be collected at node 6, and how the simulation is to be terminated. If repair statistics are also needed, we have to add a repair node and extend the network of Figure 3.3.4. The resulting network is shown in Figure 3.3.5. Six nodes are necessary to model the whole system:

- two nodes indicating whether unit A is up or down
- two nodes indicating whether unit B is up or down
- two nodes indicating whether the system is up or down

It may appear that Figure 3.3.5 has too many arcs. We can usually reduce the number of arcs by adding more nodes to the network, but we have to keep in mind that the size of the network is mainly determined by the number of nodes. To validate this model we have to make sure that nodes 6 and 7 are released only when necessary. For this goal we need to refer to the state transition diagram in Figure 3.3.2 and look at the changes in the release counters for each admissible transition. The starting state is obviously (A1,B1). By referring to the state transition diagram, we have two possible transitions from this state; to the state (A1,B2) or the state (A2,B1).

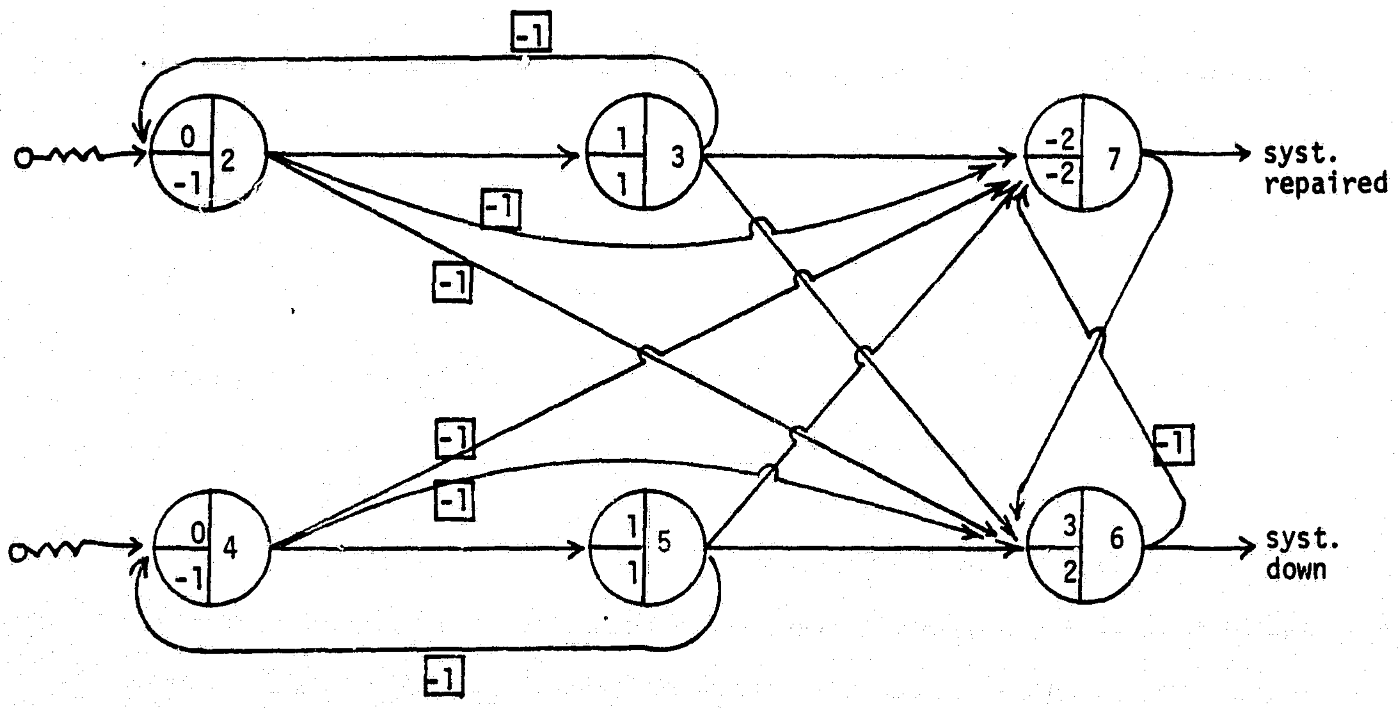


Figure 3.3.5 Two Units in Parallel
with Repair

Since the release counters at nodes 2, 3, 4 and 5 are initialized to (+1) or (-1), these nodes are released every time an incoming arc is completed. Hence, we only need to look at the release of nodes 6 and 7. When the simulation is started, nodes 2 and 4 are released and all arcs emanating from these nodes are scheduled. Arcs (2,3), (4,5), (3,2) and (5,4) are the only arcs with a non-zero time duration. When node 2 is released, N1 at node 7 changes from -2 to -1, and at node 6, N1 remains equal to 3. Node 4 is released at the same time, but the program schedules the arcs out of node 4 after the arcs out of node 2 if the input data for node 2 is read before node 4 (see data input section). This, however, does not make any difference when node 4 is released, arc (4,6) has no effect on N1 at node 6, which remains equal to 3, and arc (4,7) changes N1 from -1 to 0, causing node 7 to be released. Its release counter is then reset to N2 = -2, and arc (7,6) causes N1 to decrease from 3 to 2 at node 6. The simulation starts with the system being in the state (A1,B1) and the release counters at nodes 6 and 7 being equal to 2 and -2, respectively. N1 is different from N2 at node 6 because we want to start the simulation by releasing node 7. The release of node 7 at time 0 means that we start the simulation with the system just being repaired. This does not violate any of the assumptions of the system, and the same network can be used if the simulation starts with both items down or in any other state. Only N1 needs to be initialized differently for nodes 6 and 7. Once the simulation is started, release counters change only at the disturbing events of the system (i.e., an item fails or gets repaired). Referring to each transition in the state transition diagram (Figure 3.3.2), we can summarize the changes in the release counters of nodes 6 and 7. In each of the following tables the first column indicates the node numbers, and the second column the changes in the release counters. Each change is represented by an arrow corresponding to the arc that caused the change, with the origin of the are reported on the arrow. A letter R in the arrow indicates a reset of the release counter.

$(A1,B1) \longrightarrow (A1,B2)$

6	$2 \xrightarrow{5} 1$
7	$-2 \xrightarrow{5} -2$

$(A1,B1) \longrightarrow (A2,B1)$

6	$2 \xrightarrow{3} 1$
7	$-2 \xrightarrow{3} -2$

$(A1,B2) \longrightarrow (A1,B1)$

6	$1 \xrightarrow{4} 2$
7	$-2 \xrightarrow{4} -1$

$(A1,B2) \longrightarrow (A2,B2)$

6	$1 \xrightarrow{3} 0 \xrightarrow{R} 2$
7	$-2 \xrightarrow{3} -2 \xrightarrow{6} -1$

$(A2,B1) \longrightarrow (A1,B1)$

6	$1 \xrightarrow{2} 2$
7	$-2 \xrightarrow{2} -1$

$(A2,B1) \longrightarrow (A2,B2)$

6	$1 \xrightarrow{5} 0 \xrightarrow{R} 2$
7	$-2 \xrightarrow{5} -2 \xrightarrow{6} -1$

$(A2,B2) \longrightarrow (A1,B2)$

6	$2 \xrightarrow{2} 2 \xrightarrow{7} 1$
7	$-1 \xrightarrow{2} 0 \xrightarrow{R} -2$

$(A2,B2) \longrightarrow (A2,B1)$

6	$2 \xrightarrow{4} 2 \xrightarrow{7} 1$
7	$-1 \xrightarrow{4} 0 \xrightarrow{R} -2$

Notice that the states (A1,B2) and (A2,B1) are equivalent. This implies that the corresponding values of the release counters at nodes 6 and 7 are the same. However, identical values of the release counters at two different states does not imply that these states are equivalent. For instance, the release counter values are the same for the states (A1,B1) and (A2,B2), although these states are obviously different. This is due to the fact that the nature of the next disturbing event is different depending upon whether we are in state (A1,B1) or (A2,B2). The network shown in Figure 3.3.5 is obviously not unique. It is not difficult to verify that the network given in Figure 3.3.6 is also valid. It uses the network modification capacility and it may be easier to follow because it is less dense than the network in Figure 3.3.5. We can also find other equivalent verisions of networks for the system under the same assumptions.

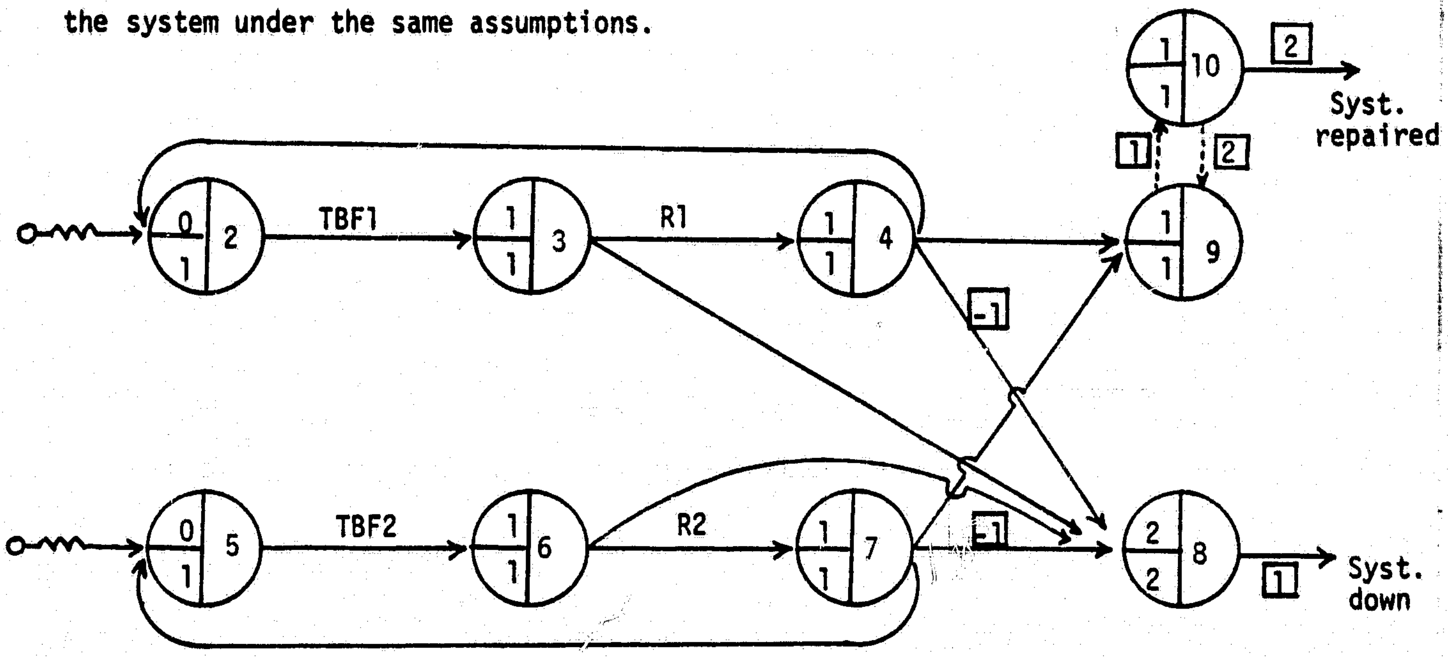


Figure 3.3.6 GRASP Network for 2 Units in Parallel

It is also possible to have other models by relaxing model assumptions. The network in Figure 3.3.5 can easily be modified to include the following situations:

- the failure of one item causes the other item to stop functioning without being failed. The system starts functioning again as soon as the failed item is repaired.
- The failure of one item causes the other item to fail with a known probability. However, only one needs to be repaired in order for the system to function again.
- only one repair facility is available for both items
- any combination of the assumptions above.

3.4 Series System with Repair

As in the previous case, depending upon the modeling assumptions, several equivalent GRASP models can be built from the same block diagram. It is also sufficient to study systems with only two components. The block diagram is shown in Figure 3.4.1 and since we have only two units, it does not convey a great deal of information about the system operation characteristics. Hence, we need to state the assumptions under which the GRASP model will be constructed. These are:

- components fail independently of each other
- two repair facilities are available so that a component goes into repair as soon as it fails
- a component starts operating as soon as it is repaired even if the system is still down
- a component does not stop operating until it individually fails. In other words, a component keeps operating even if the system is down. This is a common assumption for systems such as aircraft operational system.

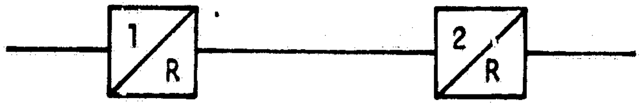


Figure 3.4.1 Block Diagram for Any System with 2 Units in Series

Notice that these assumptions are exactly the same as the ones made for a system with two units in parallel. In the parallel system, both units must be down in order for the system to be down and once the system has failed, only one unit needs to be repaired in order for the system to be repaired. In a series system, if both components are down, both units need to be repaired in order for the system to be repaired. Similarly, if the system is operating it goes down if one unit goes down. This dichotomy between a parallel system with repair and a series system with repair makes them very similar. This will become apparent in the representation of the transition diagrams and their corresponding GRASP networks. States are defined as before, but their interpretation is slightly different.

(A1,B1) = both units are operating; system is up

(A1,B2), (A2,B1) = one unit is operating; the other unit is under repair

(A2,B2) = both units are under repair

(A1,B2), (A2,B1), (A2,B2): system is down.

The state transition diagram is exactly the same as for the parallel system, and does not need to be presented. Hence, we will also refer to Figure 3.3.2 as the state transition diagram for a series system. Proceeding as in the parallel system, we obtain the network in Figure 3.4.2. We assume that the simulation starts with both items operating. We have as before:

2 nodes representing the states of item A
2 nodes representing the states of item B
2 nodes representing the states of the whole system

The network in Figure 3.4.2 differs from the network for a parallel system (Figure 3.3.5) only in the way the release counters for first release are initialized at nodes 6 and 7.

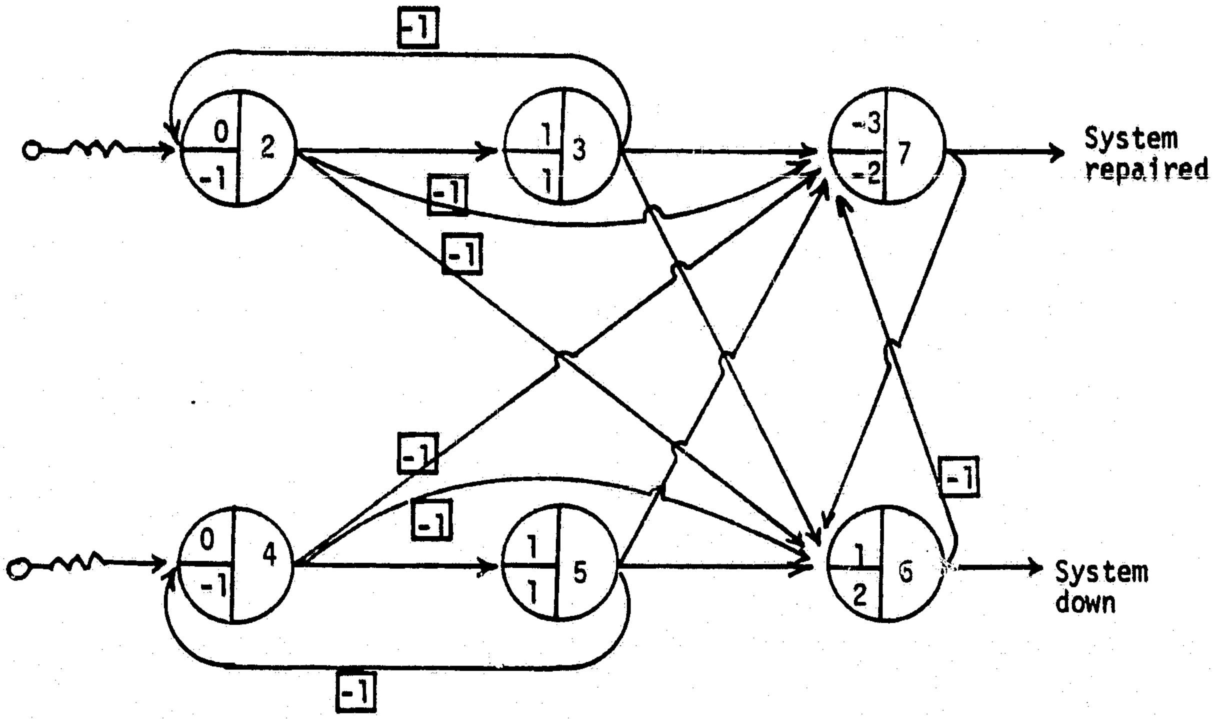


Figure 3.4.2 Two Units in Series with Repair

The simulation starts with the release of nodes 2 and 4. At the time, the release counter of node 7 is -1 and the one for node 6 remains unchanged. The starting state is (A1,B1). To validate the model we will proceed as before; examine the changes in the release counters for each transition.

$(A1,B1) \longrightarrow (A1,B2)$

6	$1 \xrightarrow{5} 0 \xrightarrow{R} 2$
7	$-1 \xrightarrow{5} -2 \xrightarrow{6} -1$

$(A1,B1) \longrightarrow (A2,B1)$

6	$1 \xrightarrow{3} 0 \xrightarrow{R} 2$
7	$-1 \xrightarrow{3} -2 \xrightarrow{6} -1$

$(A1,B2) \longrightarrow (A1,B1)$

6	$2 \xrightarrow{4} 2 \xrightarrow{7} 1$
7	$-1 \xrightarrow{4} 0 \xrightarrow{R} -2$

$(A1,B2) \longrightarrow (A2,B2)$

6	$2 \xrightarrow{3} 1$
7	$-1 \xrightarrow{3} -2$

$(A2,B1) \longrightarrow (A1,B1)$

6	$2 \xrightarrow{2} 2 \xrightarrow{7} 1$
7	$-1 \xrightarrow{2} 0 \xrightarrow{R} -2$

$(A2,B1) \longrightarrow (A2,B2)$

6	$2 \xrightarrow{5} 1$
7	$-1 \xrightarrow{5} -2$

$(A2,B2) \longrightarrow (A1,B2)$

6	$1 \xrightarrow{2} 2$
7	$-2 \xrightarrow{2} -1$

$(A2,B2) \longrightarrow (A2,B1)$

6	$1 \xrightarrow{4} 2$
7	$-2 \xrightarrow{4} -1$

After the first release of node 7, the state (A1,B1) corresponds to the values of 1 and -2 in the release counter of nodes 6 and 7, respectively. Starting from this state a second time, it is easy to verify that at the next states (A1,B2) and (A2,B1) the release counters contain the same values as above. Hence, there is no need to go through the remaining transitions.

(A1,B1) $\longrightarrow$ (A1,B2)

6	$1 \xrightarrow{5} 0 \xrightarrow{R} 2$
7	$-2 \xrightarrow{5} -2 \xrightarrow{6} -1$

(A1,B1) $\longrightarrow$ (A2,B1)

6	$1 \xrightarrow{3} 0 \xrightarrow{R} 2$
7	$-2 \xrightarrow{3} -2 \xrightarrow{6} -1$

Through this analysis we see that there is no way for node 6 to be released unless the system goes down; and there is no way for node 7 to be released unless the system gets repaired. As in the parallel case, states (A1,B2) and (A2,B1) have the same effect on the system. It is possible to have other equivalent networks. Figure 3.4.3 shows one such network that uses a network modification scheme for the repair node instead of a node with a negative release count. Each network has its own merits, and it is up to the user to choose the one that appears more understandable.

The previous assumptions can be relaxed as in the parallel system's case, and there are also different ways to extend the model.

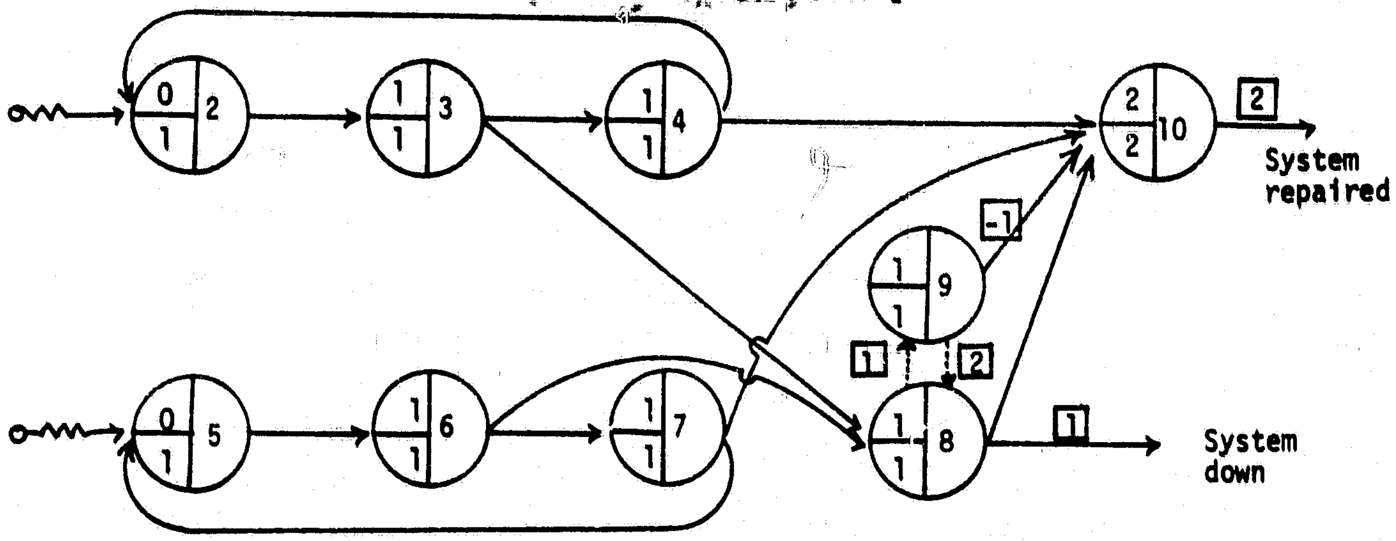


Figure 3.4.3 Two Units in Series
With Repair

3.5 Standby Redundancy

This model is depicted by the block diagram in Figure 3.5.1. The system operates with only one unit. As soon as this unit fails, the redundant unit takes over. Hence, the system goes down only if both units are in a failed state.

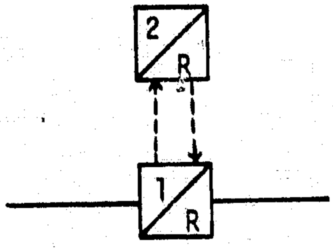


Figure 3.5.1 A One Unit Standby
Redundant System

We make the following assumptions:

- a unit cannot fail when it is in a stand-by redundant position, and it remains inactive until the other unit fails

ORIGINAL PAGE IS
OF POOR QUALITY

- the switch-over is perfect (no failure, no delay)
- when a unit fails, it goes immediately into repair
- when a unit is repaired, it stays in a stand-by position if the system is operating, and starts operating if the "up" component fails
- the performance of the system is the same whichever unit is operating.

Such a system is often called a "cold" stand-by system in contrast to the system where the redundant unit can fail, which is called a "warm" or active system.

Given the above assumption, we can identify the following states for each unit:

A1 (B1)	the unit A (B) is operating
A2 (B2)	the unit A (B) is under repair
A3 (B3)	the unit is in a redundant position.

It may help to look at the behavior of an inidivdual component before examining the combined system. Figure 3.5.2 shows the state transition diagram for component A and Figure 3.5.3 shows the corresponding GRASP model.

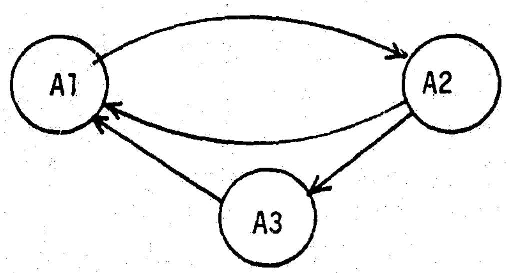


Figure 3.5.2 State Transition Diagram for One Component in a 2-component Stand-by System

It is easy to see in Figure 3.5.2 why the transitions from A1 to A3 and from A3 to A2 are not admissible. The network in Figure 3.5.3 has 3 nodes corresponding to the 3 states of the copmonent, and it will be included in the network representing the entire system. Taken by itself, we see that only 3 arcs are needed to represent how long a component stays in one state.

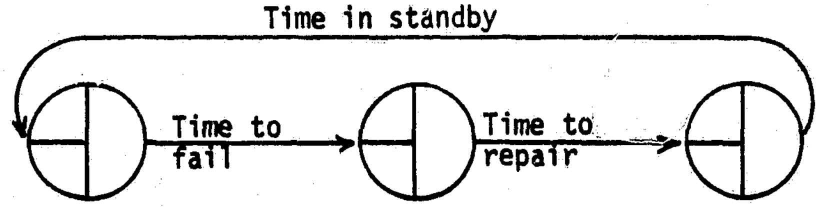


Figure 3.5.3 GRASP Network Corresponding to Figure 3.5.2

From the states of the individual components, we can identify the following admissible states for the entire system.

(A1,B3)	unit A is operating B is redundant	System is up
(A3,B1)	A is redundant B is operating	
(A2,B1)	A is under repair B is operating	
(A1,B2)	A is operating B is under repair	
(A2,B2)	Both A and B are under repair	System is down

Because of the configuration of the system, it is easy to visualize why the other states, such as (A1,B1), are not admissible.

There is no real problem identifying all admissible transitions, and the state transition diagram of the whole system is given by Figure 3.5.4 Only 8 transitions (excluding transitions for which 2 events occur at the same time) are admissible.

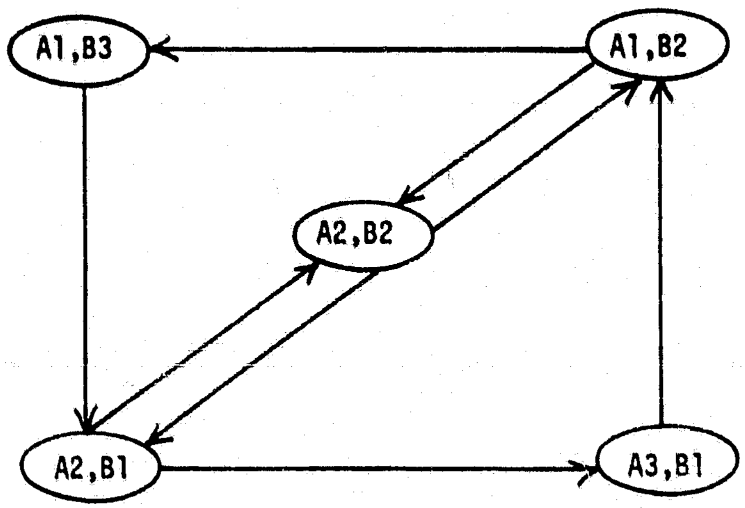


Figure 3.5.4 State Transition Diagram of a Cold Stand-by 2-unit System

Assume that the simulation starts in state (A1,B3). The GRASP network is represented by Figure 3.5.5. It has the following components.

- 3 nodes whose releases correspond to the states of component A
- 3 nodes whose releases correspond to the states of component B
- 2 nodes whose releases correspond to whether the system is down or repaired.

Only arcs representing the time to fail and the time to repair of components A and B, that is a total of 4 arcs, have non-zero duration. All the others are for the control of the node release counters.

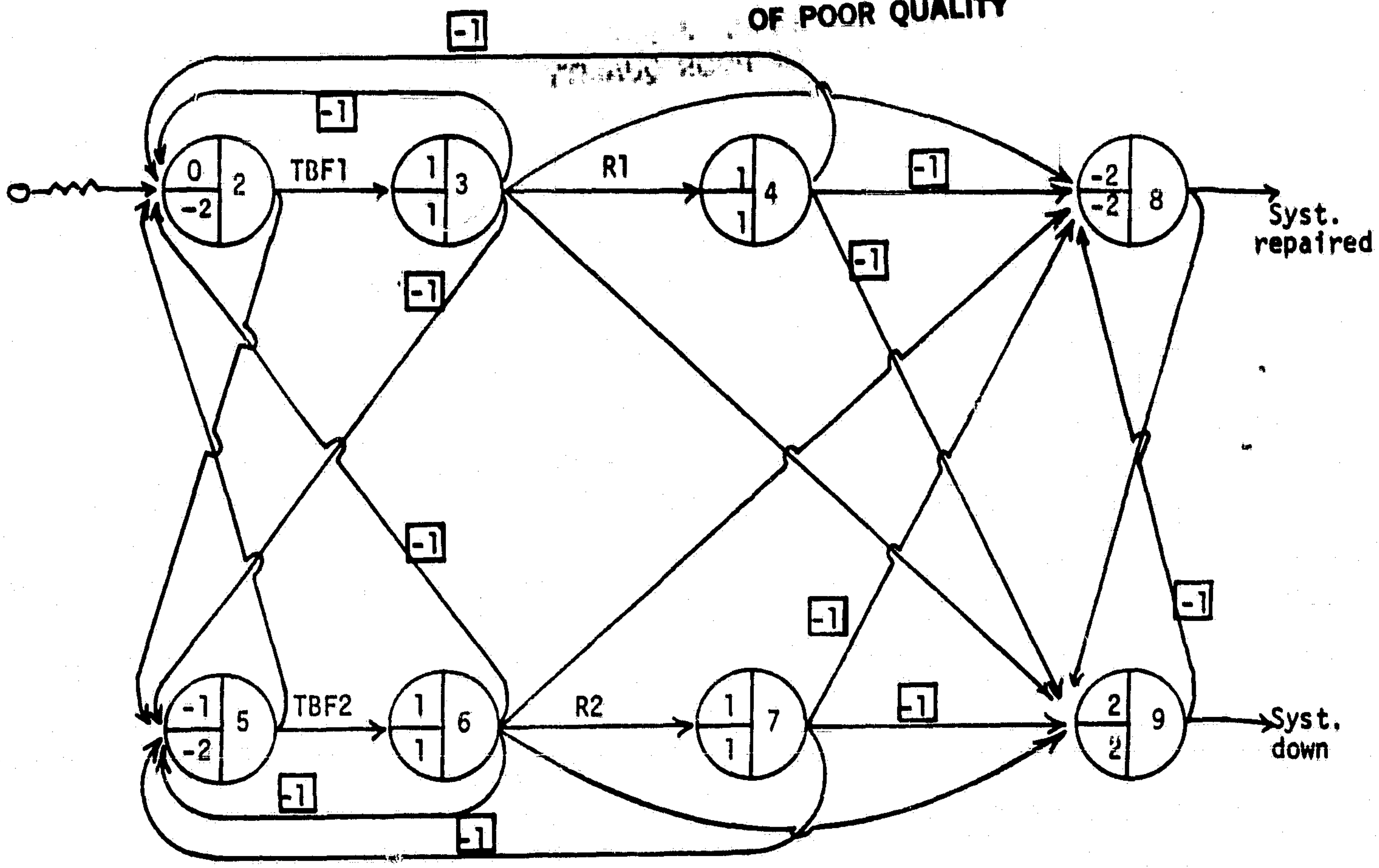


Figure 3.5.5 GRASP Network for a One-Unit System with a Cold Stand-by Redundant Unit

The validation of this model proceeds as in the previous examples, and is briefly summarized as follows when the simulation begins, node 2 is released. Since arc (2,5) has a nonnegative type, N1 at node 5 remains equal to -1. Examining the changes of the release counters for nodes 2, 5, 8 and 9, the following tables can be constructed.

It is evident from these tables that the GRASP network in Figure 3.5.5 is an exact model of the system. The symmetry that exists in the system is also shown in these tables. For instance, notice that the behavior of the system as represented by the release counters of nodes 2, 5, 8 and 9 is the same as that for states (A2,B1) and A1,B2), and for (A1,B3) and (A3,B1).

(A1,B3) ⟶ (A2,B1)

2	$-2 \xrightarrow{3} -1 \xrightarrow{5} -2$
5	$-1 \xrightarrow{3} 0 \xrightarrow{R} -2$
8	$-2 \xrightarrow{3} -2$
9	$2 \xrightarrow{3} 1$

(A2,B1) ⟶ (A2,B2)

2	$-2 \xrightarrow{6} -1$
5	$-2 \xrightarrow{6} -1$
8	$-2 \xrightarrow{6} -2 \xrightarrow{9} -1$
9	$1 \xrightarrow{6} 0 \xrightarrow{R} 2$

(A2,B1) ⟶ (A2,B1)

2	$-2 \xrightarrow{4} -1$
5	-2
8	$-2 \xrightarrow{4} -1$
9	$1 \xrightarrow{4} 2$

(A2,B2) ⟶ (A2,B1)

2	$-1 \xrightarrow{5} -2$
5	$-1 \xrightarrow{7} 0 \xrightarrow{R} -2$
8	$-1 \xrightarrow{7} 0 \xrightarrow{R} -2$
9	$2 \xrightarrow{7} 2 \xrightarrow{8} 1$

(A3,B1) ⟶ (A1,B2)

2	$-1 \xrightarrow{6} 0 \xrightarrow{R} -2$
5	$-2 \xrightarrow{6} -1 \xrightarrow{2} -2$
8	$-1 \xrightarrow{6} -2$
9	$2 \xrightarrow{6} 1$

(A1,B2) ⟶ (A2,B2)

2	$-2 \xrightarrow{3} -1$
5	$-2 \xrightarrow{3} -1$
8	$-2 \xrightarrow{3} -2 \xrightarrow{9} -1$
9	$1 \xrightarrow{3} 0 \xrightarrow{R} 2$

(A1,B2) ⟶ (A1,B3)

2	-2
5	$-2 \xrightarrow{7} -1$
8	$-2 \xrightarrow{7} -1$
9	$1 \xrightarrow{7} 2$

(A2,B2) ⟶ (A1,B2)

2	$-1 \xrightarrow{4} 0 \xrightarrow{R} -2$
5	$-1 \xrightarrow{2} -2$
8	$-1 \xrightarrow{4} 0 \xrightarrow{R} -2$
9	$2 \xrightarrow{4} 2 \xrightarrow{8} 1$

3.6 Other Common Configurations

Another common configuration in reliability is the k-out-of-n structure. As an example, we model a 2-out-of-3 system under the same assumptions as for a parallel or series system, that is:

- components fail independently of each other
- a failed component goes immediately into repair
- a component starts operating as soon as it is repaired
- a component does not stop operating until it fails.

A 2-out-of-3 system can be a system of 3 electric generators where at least 2 must be working in order for the system to function, or a three-engine air plane that can survive with only two engines. Let A, B, and C be the elements in the system. Each component has only two states; A1, B1, C1 are the operating states and A2, B2, C2 are the down states. Every combination of the states of A, B, C forms an admissible state of the system.

(A1,B1,C1) = system is up

(A1,B1,C2) = system is up

(A1,B2,C1) = system is up

(A1,B2,C2) = system is down

(A2,B1,C1) = system is up

(A2,B1,C2) = system is down

(A2,B2,C1) = system is down

(A2,B2,C2) = system is down

The state transition diagram in Figure 3.6.1 does not show the transitions corresponding to the occurrence of two or more simultaneous changes in the individual components, since the probability of such an occurrence is assumed negligible. The GRASP network is shown in Figure 3.6.2.

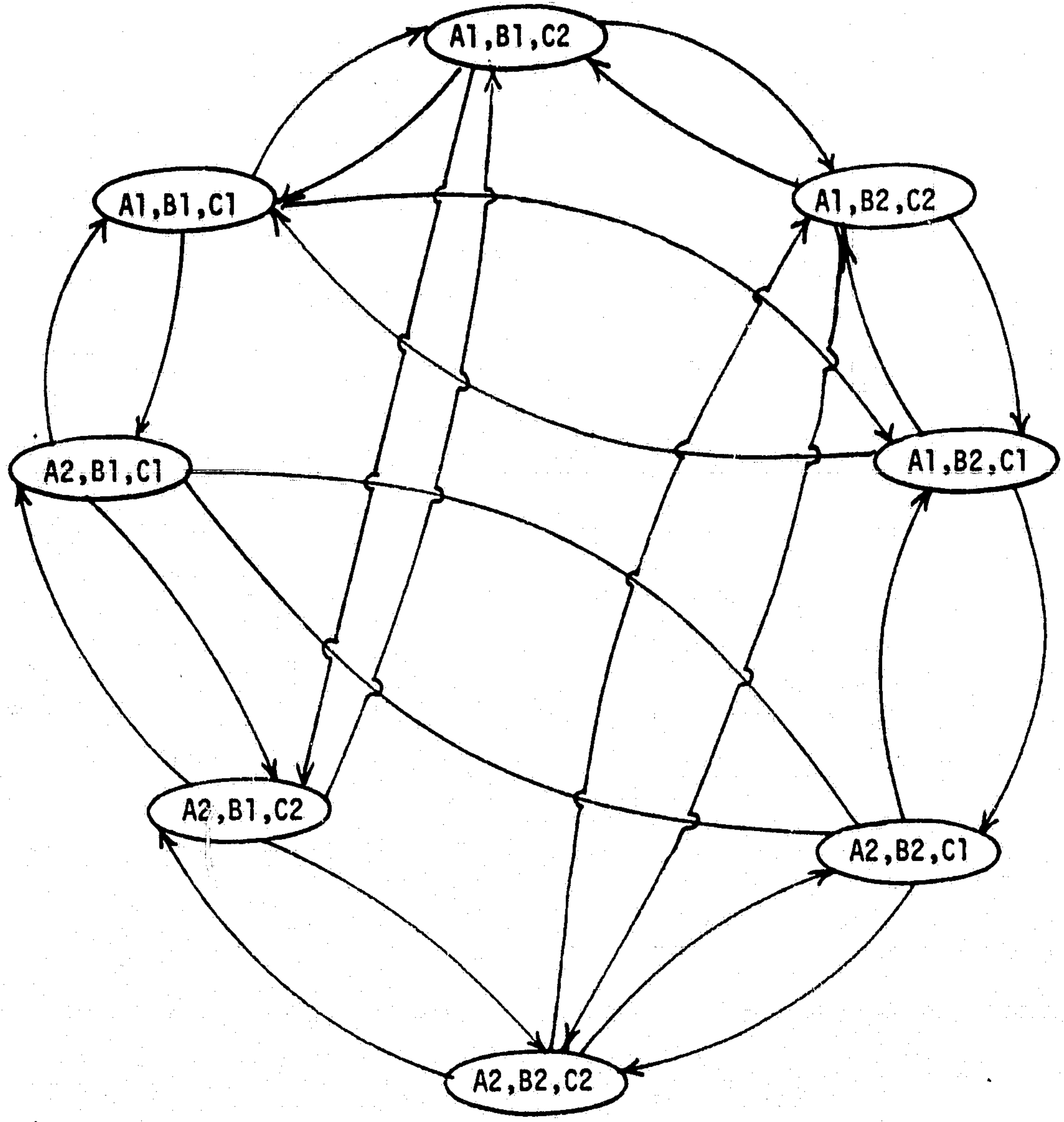


Figure 3.6.1 State Transition Diagram for a 2-out-of-3 System

ORIGINAL PAGE IS
OF POOR QUALITY

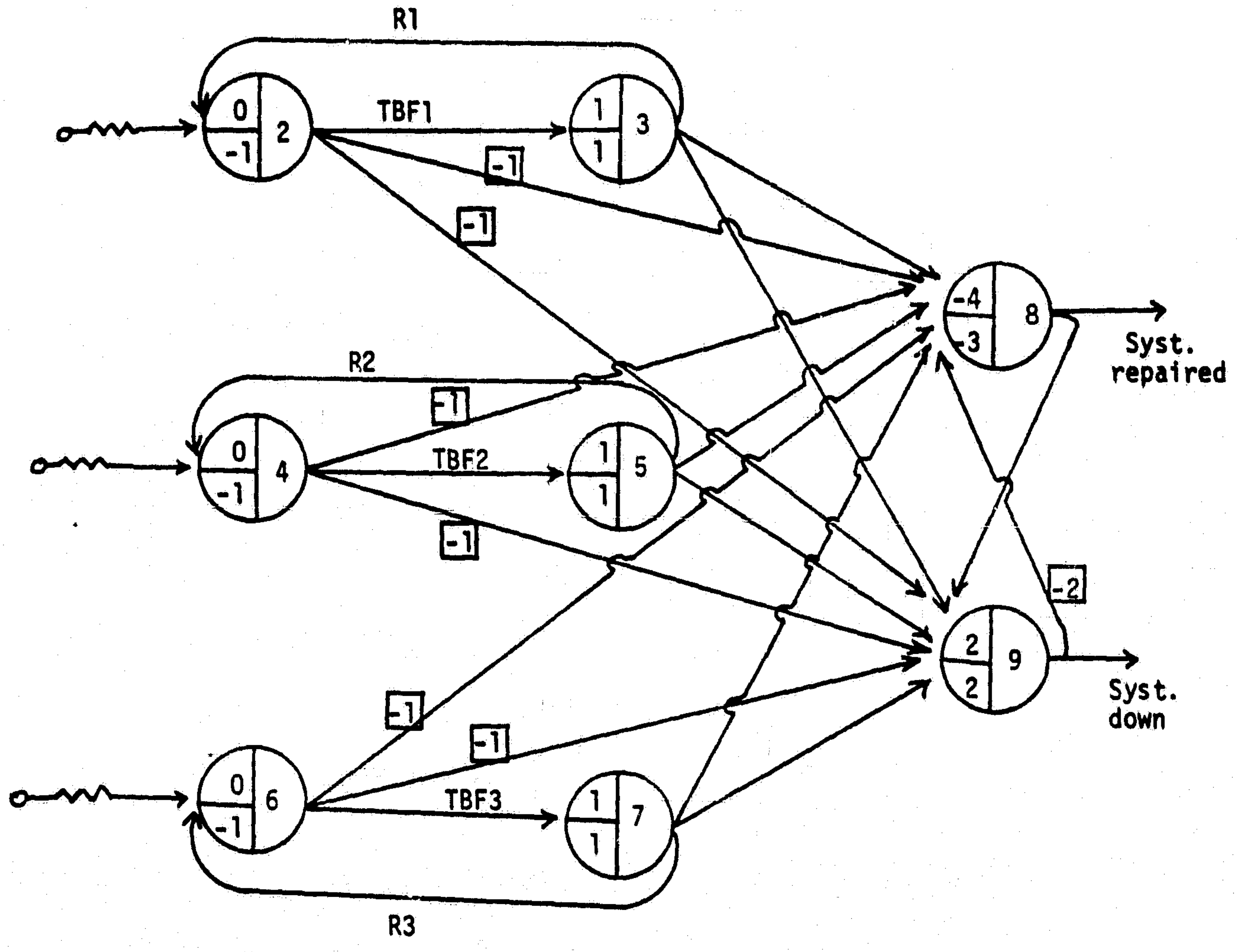


Figure 3.6.2 GRASP Network for a 2-out-of-3 System

To validate the model we do not need to look at all of the transitions, because the following classes of equivalent states can be identified.

{(A1,B1,C1)} = class 1

{(A1,B1,C2), (A1,B2,C1), (A2,B1,C1)} = class 2

{(A1,B2,C2), (A2,B2,C1), (A2,B1,C2)} = class 3

{(A2,B2,C2)} = class 4

We need to look at the transitions for only one element in each class. Let us examine the following representative from each class:

(A1,B1,C1), (A1,B1,C2), (A1,B2,C2), (A2,B2,C2)

Note that there is a total of six transitions to examine. When the simulation starts, N1 becomes equal to -1 at node 8 and remains unchanged at node 9.

$(A1,B1,C1) \longrightarrow (A1,B1,C2)$

8	$-1 \xrightarrow{7} -2$
9	$2 \xrightarrow{7} 1$

$(A1,B1,C2) \longrightarrow (A1,B1,C1)$

8	$-2 \xrightarrow{6} -3$
9	$1 \xrightarrow{6} 2$

$(A1,B1,C2) \longrightarrow (A1,B2,C2)$

8	$-2 \xrightarrow{5} -3 \xrightarrow{9} -1$
9	$1 \xrightarrow{5} 0 \xrightarrow{R} 2$

$(A1,B2,C2) \longrightarrow (A1,B1,C2)$

8	$-1 \xrightarrow{4} 0 \xrightarrow{R} -3$
9	$2 \xrightarrow{4} 2 \xrightarrow{8} 1$

$(A1,B2,C2) \longrightarrow (A2,B2,C2)$

8	$-1 \xrightarrow{3} -2$
9	$2 \xrightarrow{3} 1$

$(A2,B2,C2) \longrightarrow (A1,B2,C2)$

8	$-2 \xrightarrow{2} -1$
9	$1 \xrightarrow{2} 2$

3.7 Example With Cost

The average lifetime of a non-repairable fuel pump is 1700 hours. When it fails, it takes one to twelve hours to buy a new one at a cost of $500.00, and one to three hours of labor at $30.00/hour to replace it. Preventive maintenance could double the lifetime of the pump. It consists of inspecting every month, the rotation speed, the output and the electric power consumption. If any of these characteristics falls below an acceptable level of performance, then this is considered a failure and the pump is replaced. The acceptable level of performance is defined to be that level below which the pump will not survive another month. In case a replacement is necessary during maintenance, a new pump is immediately available at a cost of $450.00. In case the old pump is still good, then it is just cleaned and lubricated and put back into service. The inspection operation takes a half to one hour, the replacement, if any, one to two hours, and cleaning and lubricating, one to one and a half hours. The labor cost during maintenance is $20.00/hour. After each maintenance operation, the pump is assumed to be as good as new, whether it has been replaced or not. If the pump fails before the scheduled maintenance time, then it is replaced as in the case without preventive maintenance. It is assumed that the lifetime of the pump is exponentially distributed, the procurement time is uniformly distributed and the replacement and maintenance times are normally distributed. For the normally distributed times, it can safely be assumed that the minimum and maximum are three standard deviations away from the mean. We want to estimate the average down time and the total cost under each of the policies, with or without preventive maintenance, over a time span of three years.

Case without Preventive Maintenance:

The GRASP network is shown in Figure 3.7.1. Arc (2, 3), represents the lifetime of the pump. Every time node 2 is released, the duration of arc (2, 3) is sampled from the exponential distribution with parameter $\lambda = 1/1700$. Node 3 represents the event when the pump fails. It is also a mark node so that the time at which it is released is recorded for later use. Arc (3, 4) represents the procurement time which is sampled from the uniform distribution between 1 and 12, and its setup cost is \$500.00. Arc (4, 5) represents the replacement time which is sampled from the normal distribution with a mean μ equal to 2 (middle of the interval [1, 3]) and a standard deviation σ such that $6\sigma = 3 - 1 = 2$ (assuming that minimum $= \mu - 3\sigma$ and maximum $= \mu + 3\sigma$). The variable cost for arc (4, 5) is \$30.00. Node 5 collects interval statistics representing how long the pump was down. The purpose of arc (5, 2) is to start a new cycle; its duration is zero, so that node 2 is released immediately after the pump has been replaced. Arc (6, 7) represents the time span of three years or 26,280 hours. Node 7 is a sink node and represents the end of the simulation.

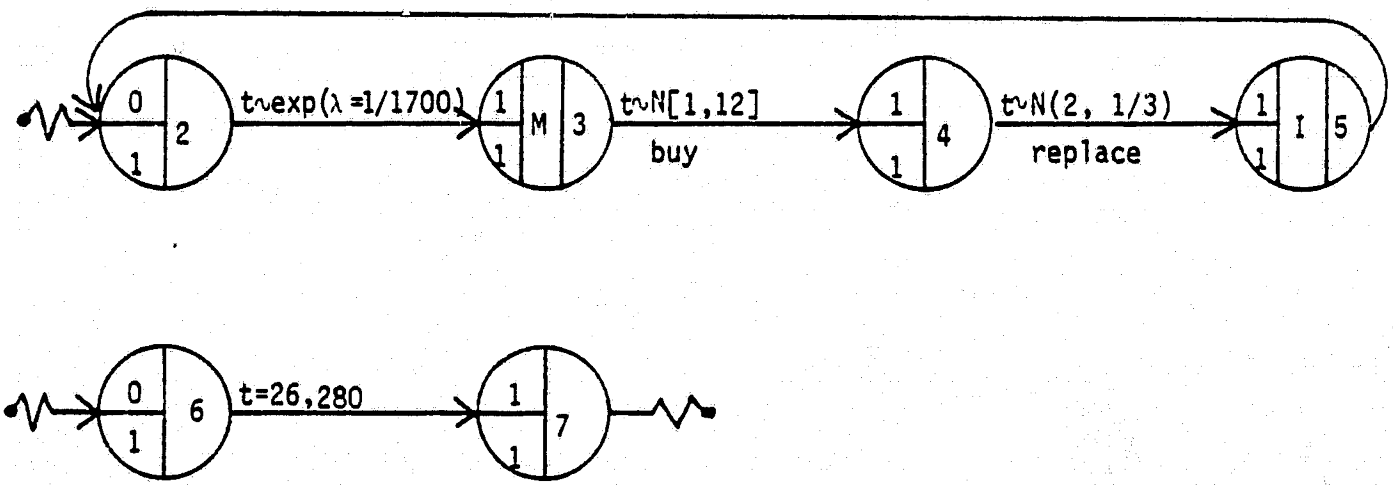


Figure 3.7.1 GRASP Network of a Fuel Pump without Preventive Maintenance

Case with Preventive Maintenance:

Before describing the model, it is first needed to estimate a few parameters. The average lifetime is 2x1700 = 3400 hours. Let T be the lifetime of the pump. The probability to replace the pump at the end of the month is estimated below.

$$
\begin{aligned}
p &= \text{Pr. } [1 \text{ month} < T \leq 2 \text{ months}] \\
&= \text{Pr. } [720 \text{ hours} < T \leq 1440 \text{ hours}] \\
&= \text{Pr. } [T \leq 1440 \text{ and } T > 720] \\
&= \text{Pr. } [T \leq 1440] \times \text{Pr. } [T > 720] \\
&= \text{Pr. } [T \leq 1440] \times (1 - \text{Pr.}[T \leq 720]) \\
&= [1 - \exp(-1440/3400)] \times [1 - (1 - \exp(-720/3400))] \\
&= [1 - \exp(-1440/3400)] \times [\exp(-720/3400)] \\
&= 0.2794
\end{aligned}
$$

The inspection time is normally distributed with minimum = .5, maximum = 1, μ = .75 and σ = (1 - .5)/6 = .0833. The replacement time is normally distributed with minimum = 1, maximum = 2, μ = 1.5 and σ = (2 - 1)/6 = .1667. The time for cleaning and lubricating is normally distributed with minimum = 1, maximum = 1.5, μ = 1.25 and σ = (1.5 - 1)/6 = .0833. Minnimum and maximum values are not needed in defining a normal distribution, but they are used here to estimate the standard deviation. The GRASP model is represented in Figure 3.7.2

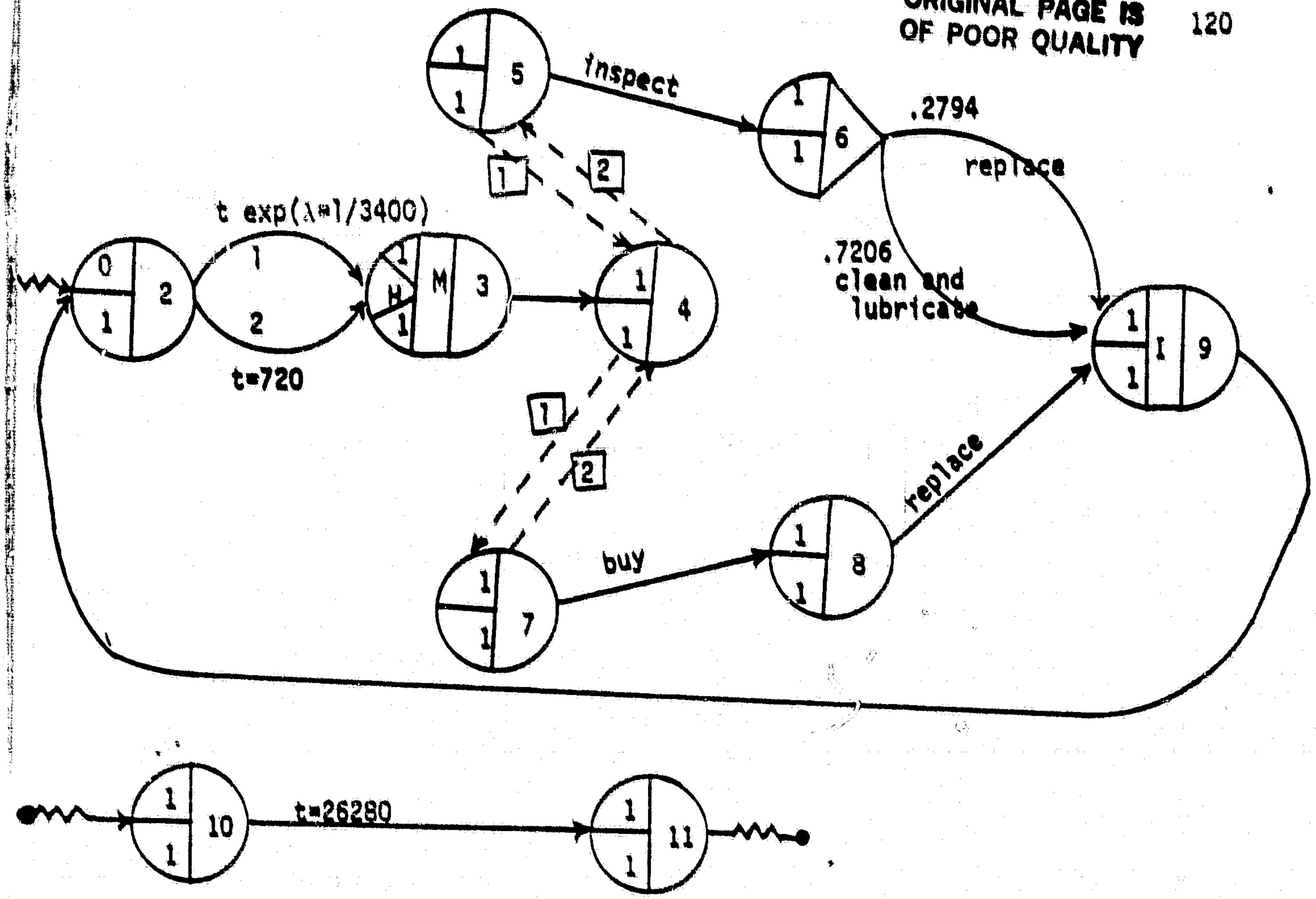


Figure 3.7.2 GRASP Network of a Fuel Pump with Preventive Maintenance

Nodes 2 and 10 are source nodes. Node 2 represents the time at which the pump starts operating and node 10 starts the time span of three years which is represented by arc (10, 11). Arc $(2, 3)_1$ represents the lifetime of the pump which is an exponentially distributed random variable with $\lambda = 1/3400$. Arc $(2, 3)_1$ has an identification number equal to 1 which is used to replace the output of node 4 by the output of node 7. Arc $(2, 3)_2$ represents the time between scheduled maintenances. Its identification number is 2 so that when node 4 is released, its output will be replaced by the output of node 5. Node 3 represents either the failure of the pump or the end of the time period between maintenances. It is a mark node with a halt capability so that one and only one arc, $(2, 3)_1$ or $(2, 3)_2$ will be completed. Arc (3, 4) is a dummy arc. Node 4 represents the same event as node 3 and it has no output of its own because it will be replaced either by node 5 or by node 7, depending on which arc, $(2, 3)_1$ or $(2, 3)_2$, is completed. Arc (5, 6) represents the inspection tiem. Arc (6, 9) represents the replacement, if any, during maintenance and is scheduled with probability .2794. Arc (6, 9) represents the cleaning and lubricating operations and is scheduled with probability .7206. Arcs (7, 8) and (8, 9) are similar to arcs (3, 4) and (4, 5) respectively, of the model for the case without preventive maintenance. Node 9 collects type I statistics and node 11 represents the end of a simulation run.

This is as far as we shall carry the analysis of simple systems, since it provides a sample of the kinds of repairable systems that can be analyzed using GRASP. We hope that the purpose of this chapter has been accomplished, and that the reader previously unacquainted with GRASP methodology has become sufficiently interested to more closely examine the GRASP methodology. The next example illustrates approaches that can be followed for modeling complex systems.

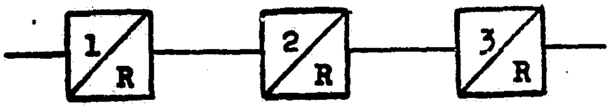
3.8 A More Complex Example

In this section a complete reliability system is presented and analyzed. It is more complex than previous examples and a network decomposition technique is presented that allows the system to be analyzed in parts.

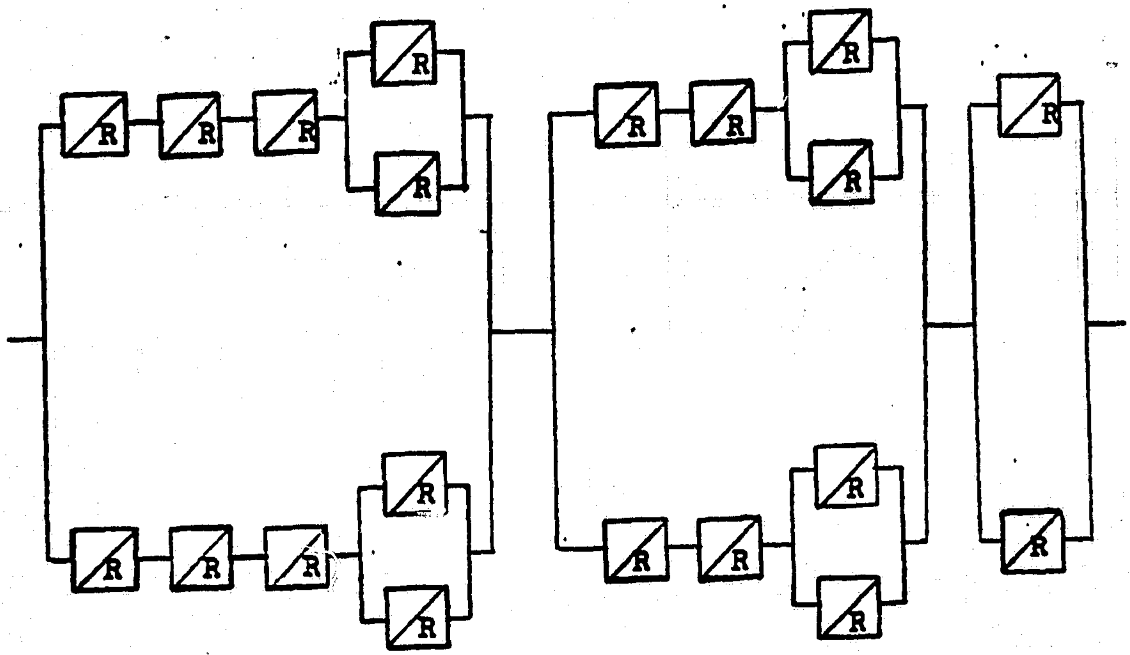
Figure 3.8.1 presents the GRASP Block Diagram for the system. The three units labeled one through three are control centers that work in concert to control a process. Each of the control centers is itself a complex, parallel redundant system. The internal structure of all of the control centers is the same. The reliability of the components within the centers may vary, of course. For the purposes of this example, the control centers will be assumed identical in structure and content. This is not a stringent assumption if the centers perform similar functions and consist of similar types of equipment. The simulation approach to reliability provides estimates of performance. It is not unreasonable, then, to estimate the reliability of a class of components and assume that each member of the class performs according to that estimate.

Now, it is obvious that the GRASP network of Figure 3.8.1 is very complex and involves a large number of nodes and arcs. Also, since reliability is concerned with rare events, an enormous number of events may need to be simulated to obtain a single failure of the overall system. To obtain the repeated failures needed to estimate the failure time distribution, an excessive amount of computer time may be required.

The solution to this dilemma is to isolate independent subgroups in the Block Diagram and simulate them separately to obtain TBF and repair distributions for the subgroups. These subgroups are then substituted into the Block Diagram as single components and the simplified system simulated. For example,



Three Control Centers in Series



A Typical Control Center

Figure 3.8.1 A Complex Reliability System in Figure 3.8.1 suppose that TBF and repair distributions are known for the components labeled 1, 2, and 3. This situation is equivalent to Case 1 of section 3.1.8 In order to obtain this desirable situation it is necessary to analyze the control centers that make up units 1 through 3. The control center itself is somewhat complex, but simulating one of them is much simpler than simulating 3 of them together.

Further, having decomposed the system once, it is reasonable to decompose the control centers into still lower level subgroups. Figure 3.8.2 shows the three level decomposition that will be used to study this system. Comparing Figure 3.8.2 to Figure 3.8.1 will indicate how levels 3(a) and 3(b) fit into the Level 2 decomposition shown in Figure 3.8.1.

Briefly, the procedure will be as follows. Levels 3(a) and 3(b) contain the individual components of the system. Estimates for the reliability characteristics of these components are assumed known. The GRASP Networks for Level 3(a) and 3(b) are prepared and simulated. The GRASP output for 3(a), say, will consist of histograms of time to failure and time to repair for the 3(a) system. Using statistical goodness-of-fit techniques, a probability distribution can be fit to these histograms [20] (alternatively, the histograms produced by GRASP can be used instead of goodness-of-fit tests). The result is that the 3(a) subgroup can be represented by a single block in the GRASP Block Diagram. This procedure is repeated for the 3(b) subgroup. At level 2 the results of levels 3(a) and 3(b) are used as the failure and repair characteristics for the blocks 12, 13, and 14, 15 in Level 2, respectively. Units 10 and 11 are single units and are simulated directly. Then, Level 2 is simulated, and the resulting histograms fit to distributions which are used in the simulation

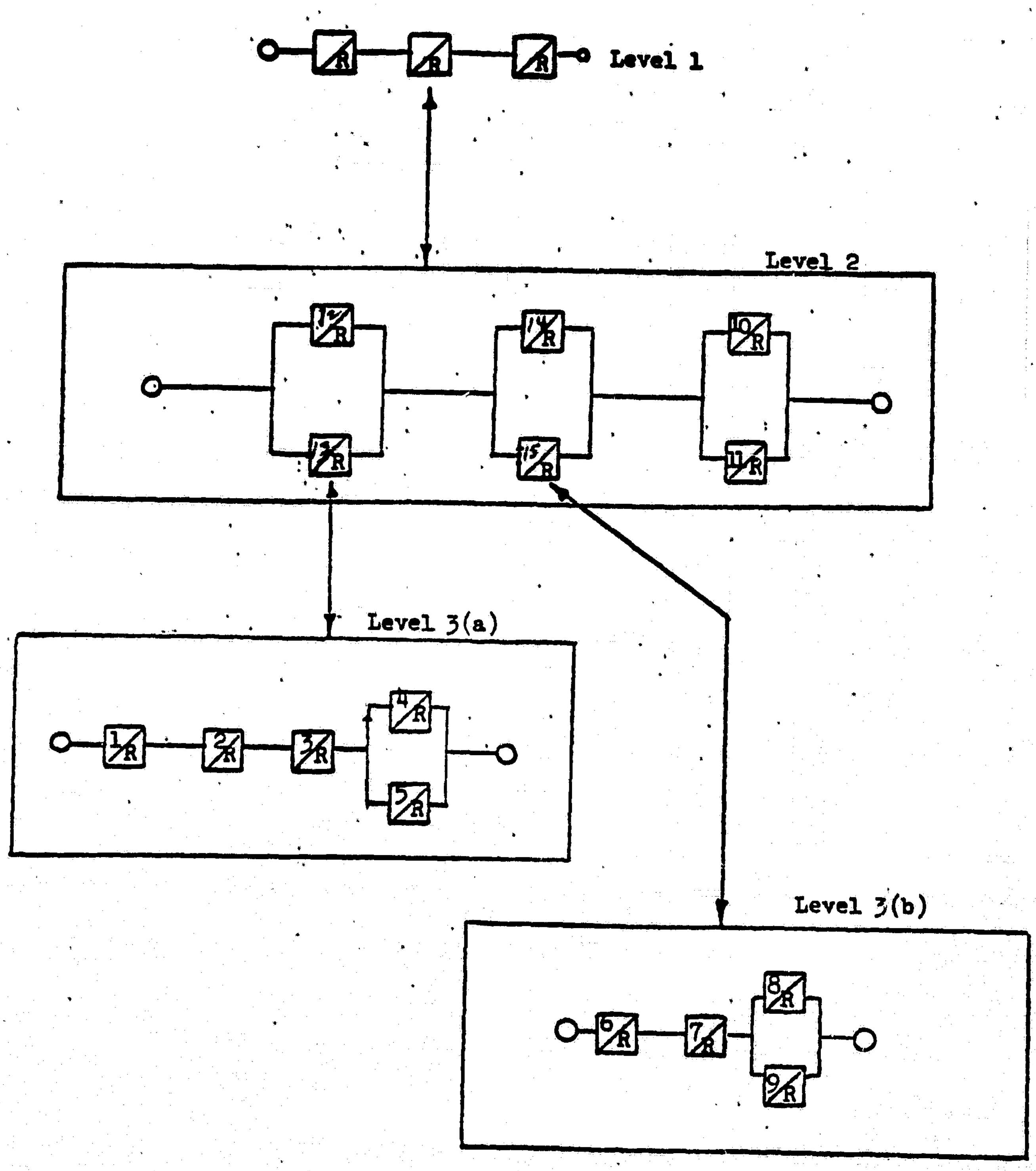


Figure 3.8.2 Three Level Decomposition

of Level 1. The Level 1 simulation gives estimates of the reliability characteristics of the overall system shown in Figure 3.8.1.

Figure 3.8.3 contains the GRASP Block Diagram and specification table for Level 3(a). Figure 3.8.4 is the GRASP network for Level 3(a). Figures 3.8.5 and 3.8.6 are analogous for Level 3(b). The GRASP networks for Level 2 and Level 1 are shown in Figure 3.8.7 and 3.8.8, respectively. Detailed information is given in Figure 3.8.3 for Level 3(a). For example, from the specification table it can be seen that unit 1 has a TBF distribution that is gamma with a mean of 25 time units and a standard deviation of 18. The analyst has determined that 500 is a reasonable maximum to use. Also, unit 1 has an exponential repair distribution with mean 0.1 time units.

Figure 3.8.4 reveals how the standard GRASP Networks seen previously can be used to build up more complex situations. Units 4 and 5 in Level 3(a) form two units in parallel with repair (see 3.3.6). Nodes 11 through 19 in Figure 3.8.4 are taken directly from Figure 3.3.6. The time information for the arcs is taken from the Specification Table in Figure 3.8.3 and the list of distribution numbers in Section 2. The parameter sets specified are those that will be used on the data cards describing the network.

Now, if we consider units 4 and 5 as a single component, Level 3(a) is just four units in series with repair as in Section 3.4. Nodes 2 through 10 of Figure 3.8.4 are connected to nodes 20 and 22 and they represent units 1, 2, and 3. Note that each group of nodes that represent a unit (say nodes 2, 3, and 4) are connected to nodes 20 and 22 by two arcs. One arc gives an indication of failure (arc 3-20), and the other

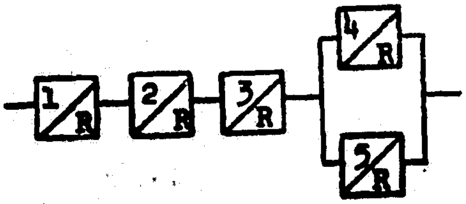
SPECIFICATION TABLE - LEVEL 3(a)

Unit	TBF Distribution	Mean	St.Dev.	Min	Max	Other Information
1	Gamma	25	18	0	500	
2	Exp	25	-	0	500	
3	Erlang-2	30	-	0	500	
4	Gamma	5	2	0	100	Same as 5
5	Gamma	5	2	0	100	Same as 4
Unit	**Repair Distribution**	**Mean**	**St.Dev.**	**Min**	**Max**	**Other Information**
1	Exp	0.1	-	0	10	
2	Exp	0.1	-	0	10	
3	Gamma	0.2	0.4	0	20	
4	Normal	0.1	0.04	0	.33	Same as 5
5	Normal	0.1	0.04	0	.33	Same as 4

Figure 3.8.3 Block Diagram and Specification Table for Level 3(a)

ORIGINAL PAGE IS
OF POOR QUALITY

Figure 3.8.3 (Continued)



Level 3(a)

indicates completion of repair (arc 4-22).

The arcs leaving nodes 17 (failure) and 19 (repair) provide this information for the parallel combination of units 4 and 5. Hence, they may be connected to nodes 20 and 22 exactly as are the fail-repair arcs of the other units.

All that remains is to collect statistics from the arcs leaving nodes 20 and 22. Nodes 23 through 27 do this. Node 26 will collect the histogram of failure time. Since node 25 is a source node it is released at the start of the simulation. Arc 25-26 will be the first activity to complete at node 26 which collects Delay statistics. Recall that D statistics are the time delay from first activity completion to node release. Since the system starts in an operating condition, the next activity completed at node 26 will be from arc 23-26 which indicates a system failure. Node 26 will be released with the completion of the second activity since N1=2. The information recorded is the time between first activity completion and node release. Note that this is the time to system failure. The release count will then be set to N2=2.

The next activity to node 26 must be arc 23-26 (end of repair) which will reduce the release count from 2 to 1 at node 26. The node is again released when the system fails and arc 24-26 is taken. Node 26, then, will continue to collect statistics on the time during which the system was operating: in other words, the time between failures.

Node 27 will collect statistics on the repair times. The first activity to complete at node 27 will be arc 24-27. Hence, the first completion at node 27 is induced by system failure and the second activity required for node release is induced by the end of repair. The interval, as desired, corresponds to the repair time. Sink node 28 will be released when 999 system

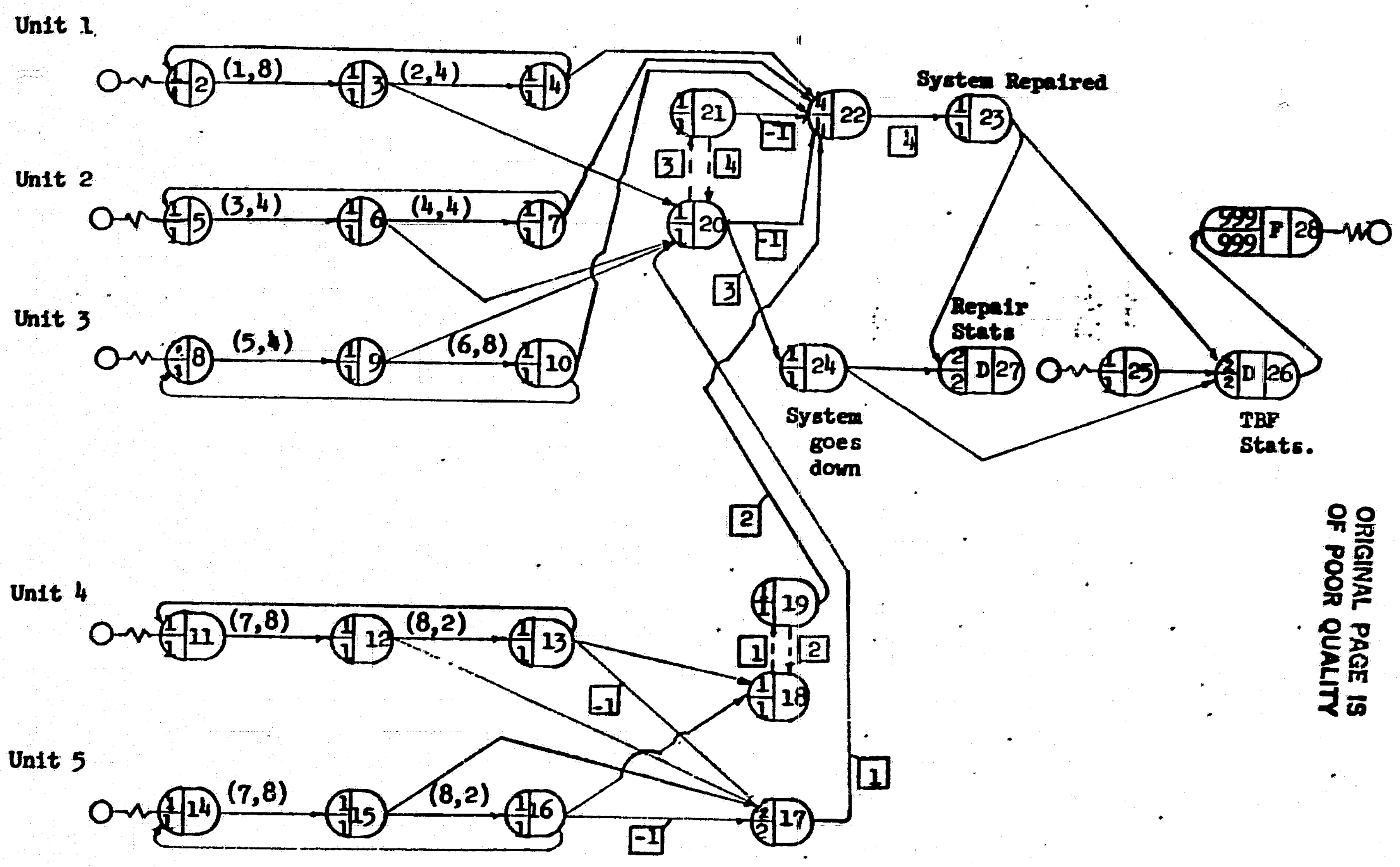


Figure 3.8.4 GRASP Network for Level 3(a)

failures have occurred. Even though statistics at node 28 are not of interest in this case, first statistics will be collected. This occurs because, even if we leave the desired statistics type blank, the default type at all statistics nodes is F.

Figures 3.8.5 and 3.8.6 are the Specification Table and GRASP Network for Level 3(b). Note that the network in 3.8.6 is identical in structure to Figure 3.8.4 for Level 3(a). The only major difference is that there are only two single units in series (units 6 and 7) instead of three. To be consistent with this, N1 = N2 = 3 for node 46.

The unit and node numbers in Figure 3.8.3 and 3.8.4 are different from those in Figure 3.8.5 and 3.8.6. This is not necessary, of course, since they will be simulated separately. However, to promote clarity for this example, node and unit numbers will not be repeated between networks.

Note that in Figure 3.8.6, arcs 35-36 and 38-39 have the same parameter set and distribution type. Since units 8 and 9 are identical, they have the same TBF distribution. It is not necessary that the parameters of that distribution be stored in different parameter sets. All arcs with the same distribution can reference the same parameter set.

The Level 2 GRASP network is shown in Figure 3.8.7. Distribution information for units 12 through 15 is left blank because the distributions will not be known until Levels 3(a) and 3(b) are simulated. Units 10 and 11 are single components and their specifications are shown in Table 3.8.1.

Each of the paralleled groups at Level 2 (see Figure 3.8.2) is represented by nine nodes (nodes 53 through 61 for example) which are arranged exactly as in Figure 3.3.6. These three parallel groupings are connected in series by nodes 80 through 82 just as nodes 20 through 22 connect the units in series in Figure 3.8.4. Finally, nodes 83 through 88 collect statistics in exactly the same manner as nodes 23 through 28.

Table 3.8.1 Specification Table for Units 10 and 11

SPECIFICATION TABLE - UNITS 10 AND 11				
Unit	TBF Distribution	Mean	Min	Other Information
		St.Dev.	Max	
10	Erlang-2	7	0	Same as 11
		-	50	
11	Erlang-2	7	0	Same as 10
		-	50	
Unit	Repair Distribution	Mean	Max	Other Information
		St.Dev.	Min	
10	Exp	0.1	0	Same as 11
		-	10	
11	Exp	0.1	0	Same as 10
		-	10	

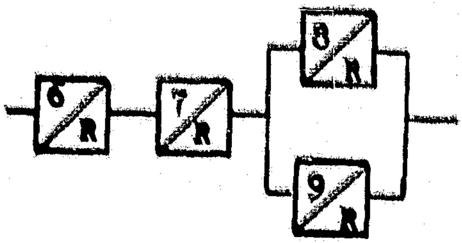
Figure 3.8.8 is the GRASP Network for Level 1. It is a direct application Statistics are maintained in exactly the same manner as for the other levels. One difference is included, however. A C-node is added to keep track of the accumulated down time over the specified mission time of 1000 time units.

C-node 106 is activated and deactivated exactly as in node 107 which collects the repair time histogram. The difference is that node 106 maintains a running sum of the time that is activated so C-node 106 records one datum each time a run is completed. The run is terminated by the release of node 110. Multiple runs will yield a histogram of cumulative down time at node 106.

SPECIFICATION TABLE - LEVEL 3(b)				
Unit	TBF Distribution	Mean / St.Dev.	Min / Max	Other Information
6	Gamma	20 / 15	0 / 300	
7	Erlang-2	22 / -	0 / 300	
8	Exp	5 / -	0 / 100	Same as 9
9	Exp	5 / -	0 / 100	Same as 8
Unit	Reapir Distribution	Mean / St.Dev.	Min / Max	Other Information
6	Exp	0.2 / -	0 / 10	
7	Exp	0.2 / -	0 / 10	
8	Triangular		0 / 1	Mode=0.1 (same as 9)
9	Triangular		0 / 1	Mode=0.1(same as 8)

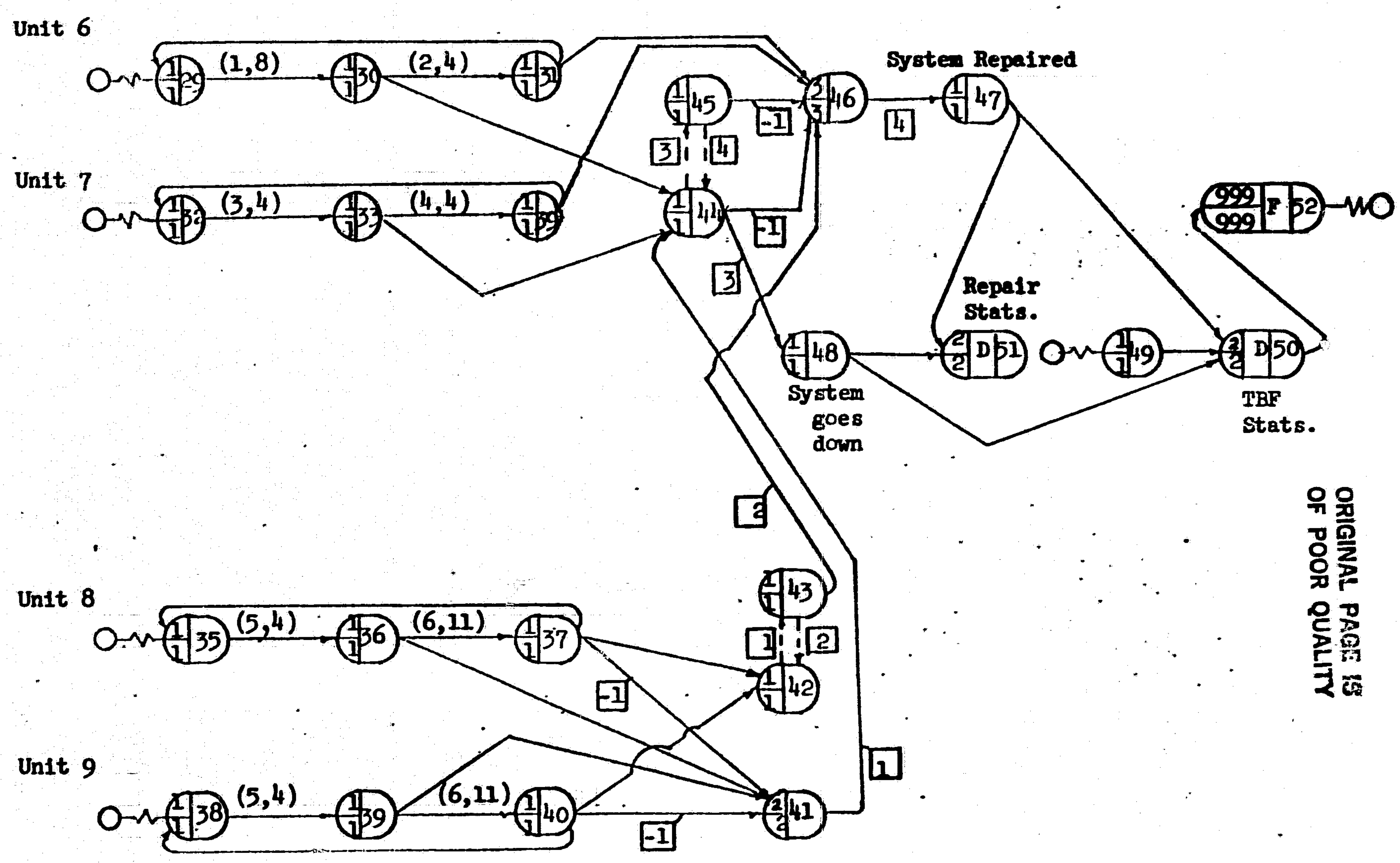
Figure 3.8.5 Block Diagram and Specification Table for Level 3(b)

Figure 3.8.5 (Continued)



Level 3(b)

Once again, the parameter sets and distribution type for the units are left blank. The failure and repair time distributions will be determined from the histograms of Level 2. Note that units 16 through 18 are all identical, so all the failure distributions in Figure 3.8.8 will be the same. Similarly, the repair distributions will also be the same.



ORIGINAL PAGE IS
OF POOR QUALITY

Figure 3.8.6 GRASP Network for Level 3(b)

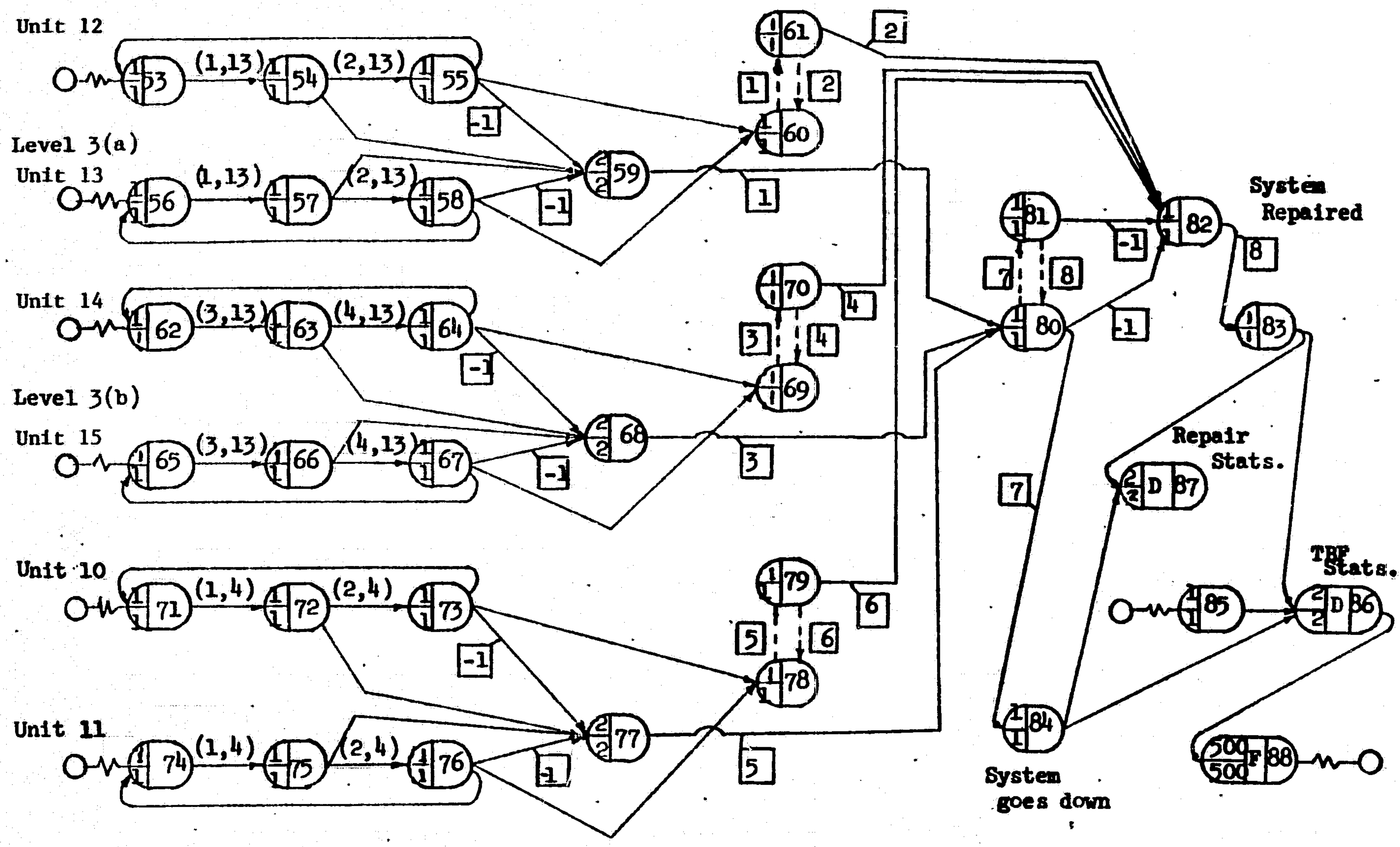


Figure 3.8.7 GRASP Network for Level 2

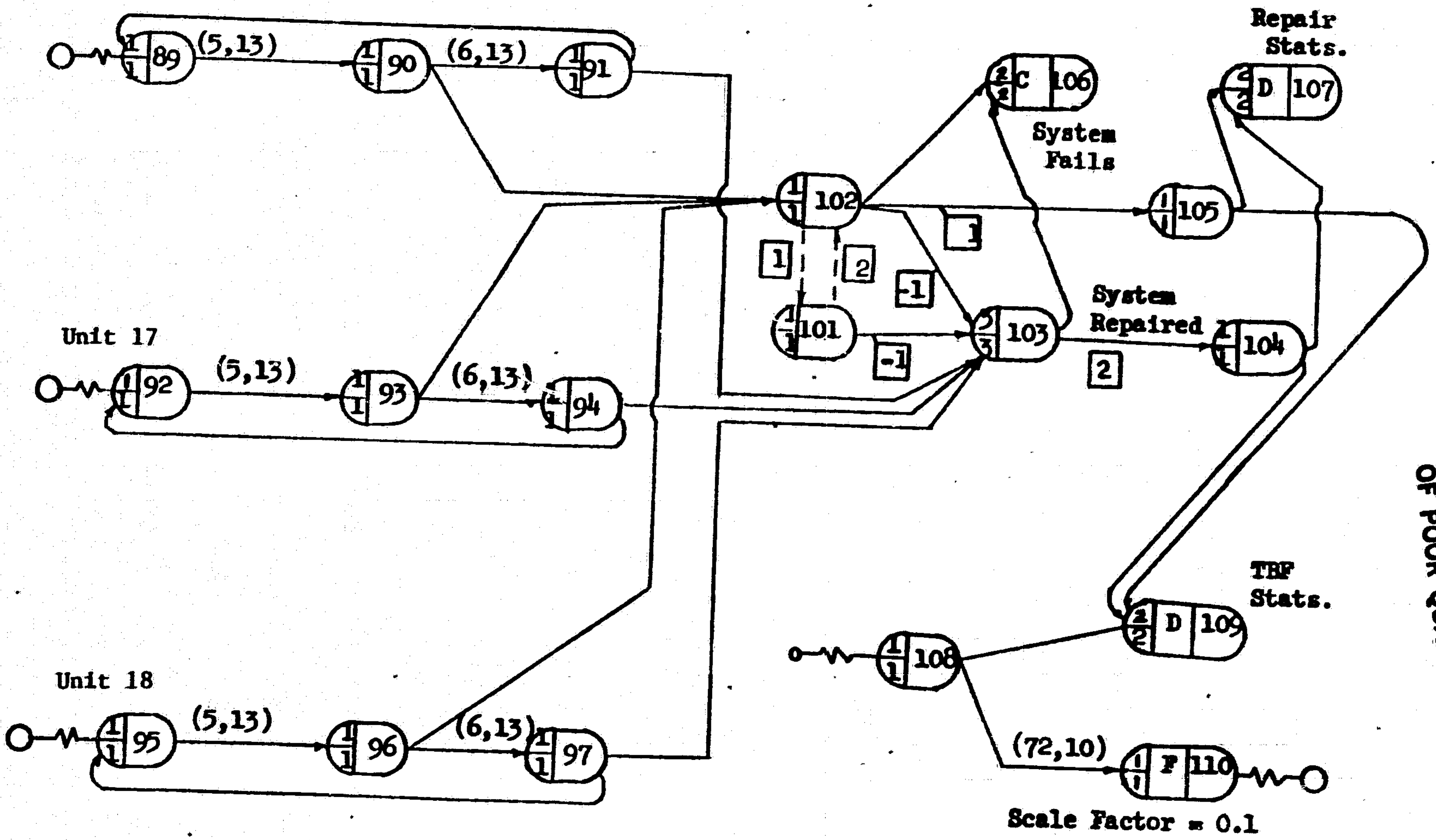


Figure 3.8.8 GRASP Network for Level 1

SECTION 4

PREPARATION OF GRASP INPUT DATA

This section will present complete information on the preparation of input data cards for GRASP. In addition, input data decks for the examples in Section 3 will be presented. Output for these examples will also be presented.

Input to GRASP is very flexible and has a number of options and special features that make it attractive for reliability analysis. Before the input card descriptions can be readily understood, however, it is essential that the user have knowledge of the control language and understand the concept of a subsystem.

4.1 The GRASP Program Control Language

GRASP input is segmented by the control language into independent functions. The control language consists of words called "Keywords" that are five letters or less in length (only the first four characters are significant, however). Keywords are punched on data cards left-justified in columns one through five and are used to command GRASP to take a particular action. Allowable Keywords are given below with a brief explanation of each.

NEW - read a complete GRASP network

RUN - run the simulation of the network currently residing in memory and produce all appropraite output

STOP - terminate program execution

HIST - read an empirical distribution from cards

PUNCH - punch a histogram from the just ended simulation in a format suitable to be read by the HIST keyword

SAVE - save a histogram from the just ended simulation as an empirical distribution to be used in a subsequent simulation

EDIT - enter the EDIT mode, where chnages can be made to the existing network

The NEW and HIST Keywords direct GRASP to read data cards. These data cards are placed immediately after the Keyword card and are identified by the Keyword which causes them to be read. For example, there are six data card types which are read after a NEW Keyword is encountered. These card types are identified as NEW-1, NEW-2, and so on to NEW-6. Similarly, the three HIST type data cards are HIST-1, HIST-2, HIST-3.

When in the EDIT mode, another group of command words called "Type Words" is available to specify the information to be edited. Type words are punched exactly like Keywords. The allowable Type words are given below:

ONE - edit the information on data card NEW-1

NODE - edit nodes in the existing network (data card type NEW-2)

PARM - edit Parameter Sets (data card type NEW-3)

ARCS - edit arcs in the existing network (data card type NEW-4)

CNODE - edit master C-node parameters (data card type NEW-6)

detailed descriptions of all data card types will be given in Section 4.4. First, however, a preliminary example of a deck set up will be useful. Figure 4.1.1 contains two examples of typical data decks. Figure 4.1.1 (a) is a typical example of a deck set up which will read one network, simulate it, and stop. The NEW Keyword causes the immediate following NEW data cards to be read. These data cards completely describe a GRASP network. The RUN command causes the previously read network to be simulated and all output to be produced. STOP terminates execution.

Figure 4.1.1 (b) presents a more complex example. As before, the NEW card causes a complete network to be read. Presumably, the network requires discrete distributions, because an HIST Keyword immediately follows the NEW data cards. The HIST Keyword causes discrete distributions to be read from the HIST data cards. At this point the network is simulated by the RUN card.

After the simulation is complete, the user wants to modify the network, perhaps to test the effect of changing some of the distribution types and parameter sets. The EDIT Keyword puts GRASP into the Edit mode so it is ready to accept changes.

The PARM Type word instructs GRASP to read data cards of type NEW-3 which describe parameter sets. In this way, new parameter sets may be created or old ones changed by storing new information in them. When this operation is completed, the ARCS card causes type NEW-4 data cards to be read. These describe arcs in the network. Finally, the RUN card causes the modified network to be simulated. "PUNCH 6 AS 1" causes the time histogram at node 6 to be punched as discrete distribution number 1.

The next NEW card causes a completely different network to be read. RUN and STOP simulate it and terminate.

These simple examples serve to illustrate the use and power of the control language. The language obviously permits multiple simulations to test the sensitivity of the solution to different parameters in the system. As will be seen in Section C, the language also permits the analysis of a decomposed network (such as that given in Section 3.7) in a single program execution.

```
(a)                              (b)

NEW                              NEW
     NEW-1 thru NEW-6                 NEW-1 thru NEW-6
RUN                              HIST
STOP                                  HIST-1 thru HIST-3
                                 RUN
                                 EDIT
                                 PARM
                                      NEW-3
                                 ARCS
                                      NEW-4
                                 RUN
                                 PUNCH 6 AS 1
                                 NEW
                                      NEW-1 thru NEW-6
                                 RUN
                                 STOP
```

Figure 4.1.1 Examples of Data Deck Set-ups Using the GRASP Control Language

4.2 Subsystem Duplication

Reliability networks often have groups of components which are repeated a number of times. For example, in Figure 3.7.2, the Level 3(a) Block Diagram is such a grouping. It is repeated twice in the level 2 diagram as units

12 and 13. It is reasonable to consider such groupings as subsystems.

For the purpose of GRASP input, however, the term "subsystem" has a more specific meaning. A subsystem in GRASP refers to the structure of the GRASP Network and not to the GRASP Block Diagram. Consider Figure 3.8.7, the GRASP Network for level 2. Notice the groups of nodes 53-61 and their associated arcs (not including arcs that leave the group -- in this case, arcs 61-82 and 59-80). The structure of this entire group is repeated twice more in nodes 62-70 and 71-79. For the purpose of GRASP, these arcs and nodes are called a "subsystem". By identifying this group as a subsystem, it is possible to input only one copy of the group and let GRASP internally generate all the other copies.

The proper terminology is as follows. A subsystem is a collection of nodes and arcs in a GRASP Network whose structure is repeated more than once. A subsystem node is a node in a subsystem. A subsystem arc is an arc which begins and ends in the same subsystem. Other arcs and nodes are non-subsystem arcs (even though one end of the arc may be in a subsystem) and non-subsystem nodes. There may be more than one subsystem type in a network. For example, nodes 53-61, 62-70, and 71-79 all belong to the same subsystem type. If there were other repeated groupings in the network, they would be other subsystem types.

Node numbering must be handled in a special way when using subsystems. Normally there are no restrictions on node numbering (except that each node number be unique and be greater than one and less than or equal to the maximum established by array sizes internal to GRASP). However, when generating subsystems, the nodes within and between copies of the subsystem must be consecutive. For example, in Figure 3.8.7 the nodes in the subsystem are numbered 53 through 61. These nodes and arcs will be input on data cards. GRASP will generate the other two copies, and it will generate node numbers consecutively starting with 62.

The implication for the user is that when the GRASP Network is drawn the subsystems must be identified before the nodes are numbered. Otherwise there is a possibility of having GRASP generate a node number that is also used elsewhere.

Finally, note that all copies of a subsystem need not be identical in their parameters. Only the structure (arc and node configuration) need be the same. Again looking at Figure 3.8.7, the failure distributions on arcs 53-54 and 62-63 will be different. Procedurally, GRASP reads the first copy of the subsystem. Then it generates the required number of copies. Finally, GRASP will accept changes to individual nodes (N1 and N2 for example) and arcs in the generated copies.

Complete information on subsystem generation is contained in Section 4.4.

4.3 Parameter Sets, Distribution Types and Random Deviates

The available probability distributions in GRASP were given in Section 2. The sampling procedure used by GRASP is a two step process. GRASP first examines the arc to determine the distribution type. Then it uses the parameters stored in the parameter set to generate a random variate from the specified Distribution Type that has those parameters. For convenience, the Distribution Types and their associated code numbers are listed again in Table 4.3.1. The information needed in the parameter sets is given in Table 4.3.2.

Table 4.3.1 Distribution Types

Code	Distribution Type	Modify Parameter Set
1	Constant	
2	Normal	
3	Uniform	
4	Erlang-K	
5	Lognormal	YES
6	Poisson	
7	Beta	YES
8	Gamma	YES
9	Beta fitted to three values as in PERT	YES
10	Constant equal to parameter set number divided by input constant	
11	Triangular	YES
12	Weibull	YES
13	Empirical	

Table 4.3.2 Parameter Set Specification

Distribution Type and Number	Field 2	Field 3	Field 4	Field 5
1. Constant	Constant	NU	NU	NU
2. Normal	mean	min	max	st. dev.
3. Uniform	NU	min	mix	NU
4. Erlang-K	mean/K	min	max	K
5. Lognormal*	mean	min	max	st. dev.
6. Poisson	λ	min	max	NU
7. Beta*	mean	min	mix	st. dev.
8. Gamma*	mean	min	max	st. dev.
9. Beta (PERT)*	most likely	optimistic	pessimistic	NU
10. Constant		Not applicable		
11. Triangular*	mode	min	max	NU
12. Weibull*	A	min	max	B
13. Empirical		Not applicable		

* - indicates parameter set is modified

NU - Not Used

Parameter Set information is specified and used as follows. A Parameter Set is a vector of at most four numbers that are entered in fields 2 through 5 on a data card (field 1 is the Parameter Set number). Table 4.3.2 specifies the contents of the Parameter Set for each Distribution Type. For example, from Table 4.3.2, it can be seen that the parameter set for normal distributions consists of the mean value, user determined minimum and maximum values, and the standard deviation of the distribution. As another example, the parameters for the Erlang-k distribution are the mean divided by the parameter k, the min and max, and k. Note with $k = 1$, a negative exponential distribution results.

Most distributions have a minimum and maximum associated with them. When these distributions are sampled, if the variate obtained is less than the min, the min is used. If it is greater than the max, the max is used; and if it is between the min and max, then the variate itself is used. Hence, the distributions actually sampled have an increased probability mass at their minimum and maximum values.

A few comments need to be made before leaving this subject. As explained in Section 2, Distribution Types 10 (constant) and 13 (empirical) do not use Parameter Sets.

The parameters for the Poisson distribution consist of the parameter, λ, a minimum value, n_0, and a maximum value, n_1. No values will be sampled outside the min-max bounds. However, the distribution sampled is not the usual Poisson unless $n_0 = 0$. It is a "shifted" Poisson. If n is an integer such that $n_0 \leq n \leq n_1$, then the probability of n is given by

$$p(n) = \frac{\lambda^{(n-n_0)} e^{-\lambda}}{(n - n_0)!} .$$

In other words, the distribution has been shifted n_0 units to the right.

Finally, the parameters for the Weibull distribution are taken from the equation for the density function.

$$f(t) = A^{B}t^{B-1} \exp(-At^{B})$$

In both Tables 4.3.1 and 4.3.2 some distributions are specified as modifying their parameter sets. This inidicates that GRASP initially changes the values that are input by the user to a form that permits faster generation of variates. The immediate implication is that the same Parameter Set cannot be used to generate variates from, say, both a Gamma and a Normal distribution even though both distributions have identical parameters. This is so because the Gamma will cause the input values to be changed. When variates are called for from the Normal distribution, incorrect parameters will be used. <u>It is necessary to use a different parameter set for each different distribution, even if the parameters have the same values.</u>

It is <u>not</u> necessary, however, to use a different Parameter Set for each arc. If two arcs have the same distribution, they may use the same parameter set even if that Distribution Type modifies its parameters.

4.4 Network Data Cards

4.4.1 Main Cards: NEW-1 to NEW-6

The NEW Data Cards are used to completely describe a GRASP Network. A NEW Keyword card must immediately precede these cards. The NEW data cards must be presented to the program in the order in which they are described (Table 4.4.1). It will be noted that type NEW-2 cards have A, B, and C subtypes and type NEW-4 cards have A and B subtypes which refer to subsystems. NEW Data Cards with suffixes A, B, and C are used <u>only</u> during subsystem generation, never during EDIT operations.

On first reading of Table 4.4.1, it is recommended that all sections pertaining to data cards with A, B, or C suffixes be skipped. These cards are concerned only with subsystems generation. Once the data cards for inputting a basic network without subsystems is understood, the Table can be reread to learn subsystem procedures. In this second reading, it will be helpful to refer to Table 4.4.2. An example data deck with subsystems will be presented when the Level 2 (Figure 3.8.7) data input is described.

Table 4.4.1 Type NEW Data Cards

The symbol * next to a field number indicates that the corresponding variable may not be changed during the EDIT mode.

I. Data Cards NEW-1: Network Description

	Field	Col.	Format	Variable	Description
Card 1:					
	1	1-80	20A4	NAME	Any comment or information about the simulation model.
Card 2:					
	1	1-3	I3	NPRT	= 1 for printing the input, 0 otherwise.
	* 2	4-6	I3	NSNK	Number of sink nodes.
	* 3	7-9	I3	NSTS	Number of statistics nodes (including all stat., sink and accum. types).
	* 4	10-12	I3	MODI	= 1 if modifications exist, 0 otherwise.
	5	13-15	I3	IGRF	= 1 if histograms are to be plotted, 0 otherwise.
	* 6	16-18	I3	NCND	Number of C-nodes.
	* 7	19-21	I3	NCTS	Number of count types.
	* 8	22-24	I3	NSBS	Number of subsystem types.
	9	25-27	I3	IDMP	0 - no dump 1 - dump files before execution 2 - dump files after execution 3 - dump files before and after execution
Card 3:					
	1	1-10	I10	ISED	Starting seed for random number generation.
	2	11-20	I10	NRNS	Number of simulation runs.
	3	21-30	I10	NSTR	Run number for beginning of tracing.
	4	31-40	I10	NETR	Run number for end of tracing.
	5	41-50	F10.0	TSTR	Time for beginning of tracing.
	6	51-60	F10.0	TETR	Time for end of tracing.
	7	61-70	F10.0	SCAL	Scale factor for distribution type 10, (default is 1.0)

Table 4.4.1 (continued)

II. Data Card NEW-2: Description of Nodes (nonsubsystem nodes)

One card is required for each nonsubsystem node of the network.

Field	Column	Format	Variable	Description
1*	1-3	I3	NODE	The unique node number of this NODE (#1 is not allowed)
2*	4-6	I3	INOUT	Special characteristics of the node. Codes for these special characteristics are: 1 Source node 2 Sink node 3 Node for which statistics are collected 4 Mark node activities
3	7-9	I3	MREL1	The number of activities to release the node for the first time. Default is 1(one).
4	10-12	I3	MREL2	The number of activities required to release the node after the first time. Default = 1.
5*	13	A1	PTOPT	Output characteristics of the nodes. P for probabilistic; D for deterministic, if left blank, D will be assumed
6*	14	A1	PRMV	If the events that have been scheduled to end on this node are to be halted (cancelled) when this node is released, an "H" should be put in this field. If all different activities are required to release this node, an "A" is put in this field. A "U" indicates that both of the above conditions are desired. Otherwise, leave blank

Field 7-11 are required for statistic or sink nodes only (2 or 3 in Field 2)

Table 4.4.1 (continued)

Field	Column	Format	Variable	Description
7	15-20	F6.2	XLL	The lower limit of the second cell for the histograms to be obtained for this node. The first cell of the histogram will contain the number of times the activity was completed in less than the value given in this field.
8	21-26	F6.2	WIDH	The width of each cell of the histogram. Each histogram contains 32 cells. If WIDH = 0, no histogram will be maintained. If WIDH is negative, GRASP will automatically scale the histogram (XLL will be ignored).
9*	27	A1	MSINK	Statistical quantities to be collected: F - Time of <u>first</u> release of the node A - Time of <u>all</u> releases of the node B - Time <u>between</u> releases of the node I - Time <u>interval</u> required between 2 node releases D - Time delay from first activity completion until the node is released C - C-node, like delay, but keeps an accumulated total of all time activated. MRELP and MREL2 (Fields 3 and 4) must be equal to 2; INOUT (Field 2) must be 3
10	28-37	F10.4	XLC	Same as Field 7 but for cost histograms
11	38-47	F10.4	WIDC	Same as Field 8 but for cost histograms

Table 4.4.1 (continued)

Field	Column	Format	Variable	Description
12*	48-50	I3	NUNOD	Not used for nonsubsystem nodes (should be left blank for Card NEW-2)

The last card of this type should be blank or must have a zero in Field 1.

IIA. Data Card New -2A: Subsystem Type Information

Required only if there are subsystems (Field 17, Data Card NEW-1 number of subsystem types). In order to input the node information for a subsystem type, one data card of type NEW-2A is required. This is followed by a stream of NEW-2B cards, one for each node in the subsystem. Immediately after the NEW-2B cards is a stream of NEW-2C cards, one for each node in any copy of the subsystem type which has characteristics different from the original copy. If there are more than one subsystem type, the sequence of NEW-2A, 2B, and 2C cards is repeated for each subsystem type. See Figure 4.1.1.

Field	Column	Format	Variable	Description
1	1-5	I5	NNSUB	Number of nodes in first subsystem type
2	6-10	I5	NSSP	Number of times this particular subsystem type is to appear in network (include the original read-in plus generated ones)
3	11-15	I5	KACTS	Number of arcs in this particular subsystem type

No 'blank' cards follow, just NEW - 2B cards.

IIB. Data Card NEW-2B: Subsystem Node Description

Not required unless preceded by a NEW-2A card. The user reads in cards describing the nodes of the subsystem type. The program generates the other copies of the subsystem type. One card is read in for every node of the subsystem. Format is very similar to Data Card NEW-2.

Table 4.4.1 (continued)

Field	Column	Format	Variable	Description
1-11	1-47	same as for Data Card Type NEW-2		
12	48-50	I3	NUNOD	Number of the next copy of this subsystem type for which this node will have different characteristics than are contained on this card.

The last card of this type will have a zero in Field 1 or be a blank (signifies end of particular subsystem's nodes characterization).

IIC. Data Card NEW-2C: Variant Subsystem Node Characterization

Not required unless some NUNOD (Field 12, NEW-2B) is $\geq$ 2. This card contains the different node characteristics for a subsystem node.

Field	Column	Format	Variable	Description
1	1-3	I3	-	Not used
2-12	4-50	same as NEW-2B		

Note: These give changes in node characteristics of a designated node in a designated subsystem type. One NEW-2C card must be present for each node in any copy of the subsystem which has different characteristics than those described on the NEW-2B cards. Care must be taken to place them in the correct order. All changes to a particular copy are read before any changes to a subsequent copy. The order of different nodes is the same as the order of NEW-2B cards. No extra cards follow these except for another of NEW-2A or NEW-3 (which follow immediately).

MULTIPLE SUBSYSTEMS

In the case where multiple subsystem types occur, these may be input by simply repeating a new series of NEW-2A, 2B and 2C cards for each desired subsystem type.

III. Data Card NEW-3: Parameter Sets

The parameters associated with the distribution of the time to perform each activity.

Table 4.4.1 (continued)

Field	Column	Format	Variable	Description
1	1-3	I3	IPARS	Parameter set number
2	4-13	F10.4	PRAM(1)	
3	14-23	F10.4	PRAM(2)	Parameter values (as defined by distribution type) See Table 4.3.2 p. 112
4	24-33	F10.4	PRAM(3)	
5	34-43	F10.4	PRAM(4)	

The last card of this type must be blank or have a zero in Field 1.

IV. <u>Data Card NEW-4: Arc Descriptions (non-subsystem activities)</u>

One card for each non-subsystem arc

Field	Column	Format	Variable	Description
1*	1-8	F8.3	ATRIB(1)	Probability this arc is taken. Default is 1.0.
2*	9-11	I3	JQ	Start node
3*	12-14	I3	JTRIB(1)	End node
4	15-17	I3	JTRIB(2)	Parameter set number
5	18-20	I3	JTRIB(3)	Distribution type (default is 10)
6	21-23	I3	JTRIB(4)	Count type
7	24-26	I3	JTRIB(5)	Activity number
8	27-35	F9.2	ATRIB(3)	Set up cost
9	36-44	F9.2	ATRIB(4)	Variable cost
10	45-47	I3	NUACT	Not used in non-subsystem activities
11*	48-50	I3	JTRIB(6)	C-node associated with accumulating this activity's cost.

The last card of this type must be a blank or have a zero in Field 2.

Table 4.4.1 (continued)

IVA. Data Card NEW-4A: Subsystem Arc Description

Required only if there are subsystems (Field 17, Data Card NEW-1-1). One NEW-4A card is required for each subsystem activity in the first copy of the subsystem. The start and end node numbers must correspond to the node numbers on the previously read NEW-2B cards. Activity information for the subsystem types is read in the same order as the node information was. In other words, the first group of NEW-2A, 2B, and 2C cards must correspond to the first group of NEW-4A and 4B cards, and so on.

Field	Column	Format	Variable	Description
1-9	1-44	same as type NEW-4		
10	45-47	I3	NUACT	Number of the next copy of this subsystem type for which this arc will have different parameters than are contained on this card.

The last card of this set will have a zero in Field 2 or be a blank.

IVB. Data Card NEW-4B: Variant Subsystem Arc Characterization

Required only if same NUACT (Field 10, Data Card NEW-4A) is ≥ 2. This card will give the different activity characteristics of a subsystem arc.

Field	Column	Format	Variable	Description
1	same as 4 and 4A			
2-3	not used			
4-10	same as 4 and 4A			

Note: These give changes in activity characterization between two subsystem copies. Only the characterization, not the end or start node can be changed. One NEW-4B card must be presented for each arc in any copy of the subsystem which has different characteristics than those described on the NEW-4A cards. Care must be taken to get them in correct order. All changes to a particular subsystem copy are read before any changes to another copy. The order that arcs are read is the same as the order of NEW-4A cards (See Figure 4.4.1). No blank cards are inserted. Additional Data Card NEW-4A's or NEW-5's follow immediately.

Table 4.4.1 (continued)

V. Data Card NEW-5: Node Modification

Required only if modifications exist in the network (i.e., MODI greater than zero - Field 4, Data Card NEW-1-2). One card for each activity number that triggers modification(s). If many different modifications of nodes result from completing a given activity, a continuation card may be required. Up to 12 replacements can fit in one card.

Field	Column	Format	Variable	Description
1*	1-3	I3	NACTN	An activity number that triggers a replacement (a negative number will indicate continuation of previous cards information).
2*	4-6	I3	NABYA(-)	The number of a node to be replaced if the activity number given in Field 1 is completed.
3*	7-9	I3	NABYA(-)	The number of a node to be inserted in the network in place of the node in Field 2 when the activity in Field 1 is completed.
4-25	10-75	2I3	NABYA(-)	Fields 2 and 3 are repeated if the activity in Field 1 affects multiple replacements.

Fields 2 and 3 represent a replacement, 4 and 5 represent the next replacement and so on. The last card of this type must have a zero in Field 1 or be a blank card.

VI. Data Card NEW-6: Master C-Node Information

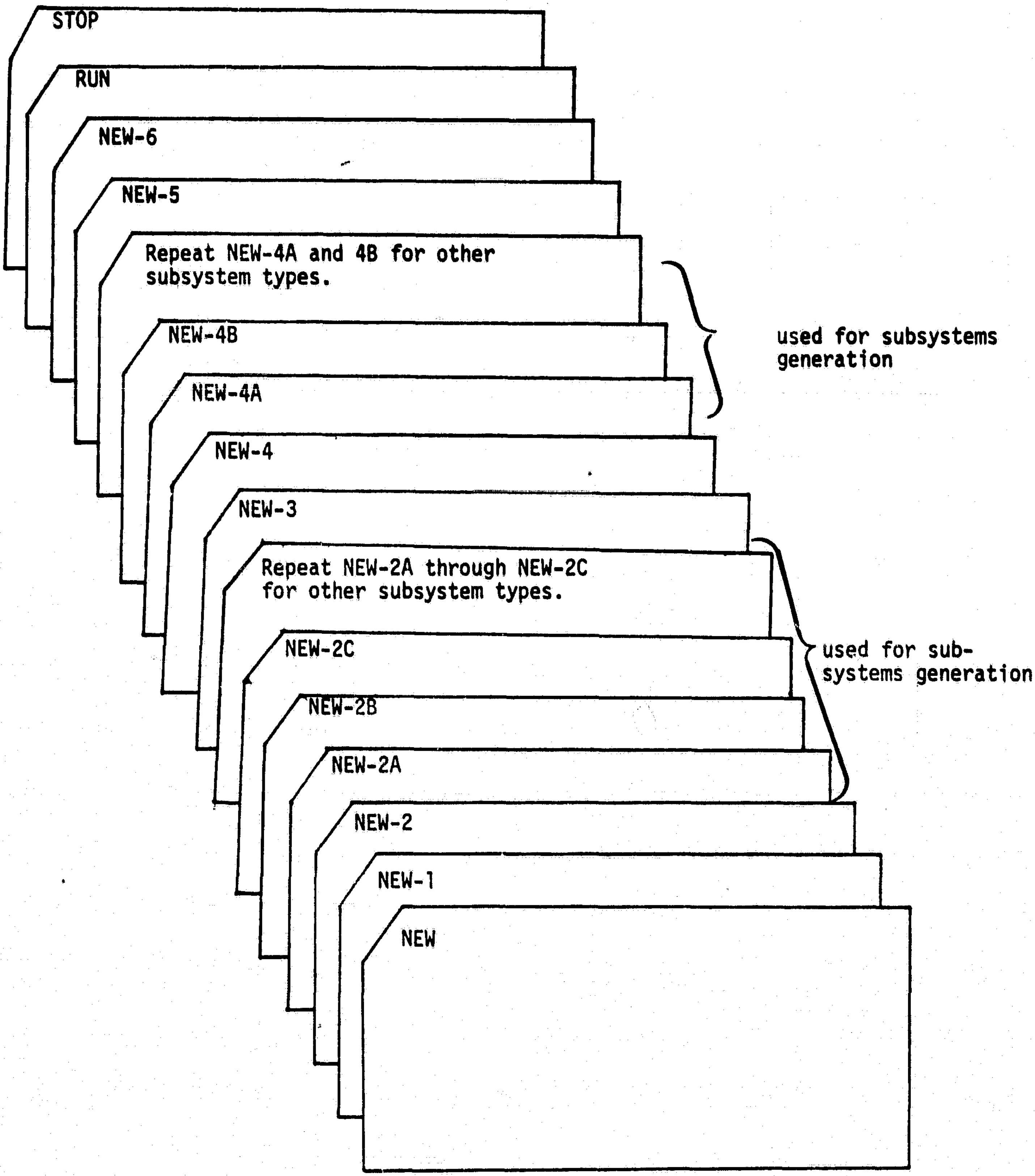
Only needed if the number of C-nodes (NCND, Field 6, Data Card NEW-1-2) is greater than zero. Only one card of this type is included, and must be present if C-nodes exist in system even if T2 and C2 criteria will not be active. (A discussion of T2 and C2 criteria is given in Section III D).

Field	Column	Format	Variable	Description
1	1-10	F10.4	T2	T2 value for C-node totals. Zero or blank indicates no T2 values. -1.0 indicates T2 = 0.0
2	11-20	F10.4	C2	C2 value for C-node totals. Zero or blank indicates no C2 value. -1.0 indicates C2 = 0.0

Table 4.4.1 (continued)

Field	Column	Format	Variable	Description
3	21-26	F6.2	XLT1	Lower limit of second cell of Master C-node time histogram
4	27-32	F6.2	WTT1	Width of cell for Master C-node time histograms
5	33-42	F10.4	XLC1	Lower limit of second cell of Master C-node cost histograms
6	43-52	F10.4	WTC1	Width of cell for Master C-node cost histograms
7	53-55	I2	ITFLG	0-T2 criterion will not terminate simulation run 1-T2 criterion will terminate simulation run
8	56-57	I2	ICFLG	0-C2 criterion will not terminate simulation run 1-C2 criterion will terminate simulation run

Table 4.4.2 Arrangement of Input Data Cards



4.4.2 Histogram Data Cards (HIST-1 to HIST-3)

A separate set of data cards is required when reading empirical distributions. These cards immediately follow the HIST Keyword and are designated as types HIST-1 to HIST-3. Details are contained in Table 4.4.3.

GRASP uses the input of empirical data, via histograms, in two ways. The usual application is to define a discrete distribution when the random variable may have up to 32 discrete values, each having a probability associated with it such that the sum of the probabilities is one.

The second application is to define a continuous distribution where the probability density function is described in a stair-step fashion with up to 32 steps (or cells) comprising a conventional histogram. In this case, the area under the density function must equal one. The cells must be of uniform width, and the variable values read in to define the histogram apply to the lower limit of each cell. GRASP will sample uniformly within each step, so the continuous random variable can have any value within the range of the histogram. The usefulness of this application is that it permits a continuous random variable that results from a lower level simulation to be approximated with a histogram and then used as input to a succeeding, higher level simulation. Without this capability it would always be necessary to determine or estimate the analytical expression for the distribution, that is, identify the distribution.

Table 4.4.3 HIST Data Cards

I. Data Card HIST-1 Parameters

Field	Column	Format	Variable	Description
1	1-5	I5	KHIS	Number of the empirical distribution. KHIS $\leq$ 10.
2	6-10	I5	NCELS	Number of cells in the distributions. NCELS $\leq$ 32
3	11-20	F10.0		Width of Cells. 0-one value per cell. >0-uniform sampling within cells

II. Data Card HIST 2 Probabilities

Field	Column	Format	Variable	Description
1	1-5	F5.4	PROB(·,JQ)	Probability (not cumulative) of first cell. Probabilities must sum to one.
2-16	6-80	F5.4	Same	Repeat of Field 1 for each cell. There must be NCELS entries. Use as many data cards of this Type as necessary.

III. Data Card HIST-3 Cell Values

Field	Column	Format	Variable	Description
1	1-10	E10.4	VAL(·,JQ)	Value for lower limit of first cell.
2-8	11-80	E10.4	Same	Repeat of Field 1 for each cell. There must be NCELS entries. Use as many data cards of this Type as necessary.

Repeat HIST-1 through HIST-3 for additional empirical distributions. The last card in this group must be blank (i.e., have a zero in Field 2 of HIST-1).

4.4.3 SAVE and PUNCH Options

The SAVE and PUNCH keywords provide a method by which the histogram produced by one GRASP simulation can be converted to an empirical distribution and used in another network. For example, in Figure 3.8.2, the outputs for Level 3(a) are the means, variances, and histograms for failure times and for repair times. The most rigorous procedure is to do goodness-of-fit tests on these histograms to identify the distributions before using the results in Level 2. However, as an alternative, or whenever there is not a good fit to a recognizable distribution function, the histograms can be used as approximations to the exact distributions. Examples of this usage will be shown for the example in Section 3.8.

The PUNCH keyword will punch the specified output histogram in a format suitable for reading by the HIST Keyword in a subsequent simulation. If it is desired to simulate both networks successively in the same computer run, the SAVE Keyword may be used. SAVE will cause the specified histogram to be converted to a GRASP empirical distribution and stored in the appropriate arrays for later sampling. Neither SAVE nor PUNCH destroy the histogram, so it can be both SAVEed and PUNCHed. However, whenever a RUN card is encountered all statistical arrays are cleared, so any histograms that have not been SAVEed or PUNCHed will be destroyed.

When histograms are collected by GRASP, the first and last cells theoretically have infinite width. In other words, the first cell contains all observations from $-\infty$ to the upper value of the first cell. Similarly, the last cell contains all values from the lower limit of the last cell to $+\infty$. When SAVE or PUNCH is invoked, these cells are truncated to the same width as all other cells in the histogram. This obviously causes a limitation on the range of the random variable which can be sampled from the resulting empirical distribution. Where possible, histograms should be scaled (by appropriate selection of cell widths) so all observations fall within the 30 inside cells. Self scaling of the histograms will insure this situation.

Finally, note that time histograms may be SAVEed or PUNCHed. No cost histograms can be processed by these commands.

Table 4.4.4 has the format for SAVE and PUNCH.

Table 4.4.4 Format for SAVE and PUNCH Keywords

Field	Column	Format	Variable	Description
1	1-4	A4	KWORD	SAVE or PUNC
2	5-8	I4	NOD	Node number for time histogram to be saved or punched
3	9-12	I4	KHIS	Number to be assigned to the empirical distribution saved or punched. JQ < 10 (as on HIST-1, Field 1).

4.4.4 EDIT Option

Editing Type words will be discussed briefly. More complete examples and discussion will be included in the next section. GRASP must be put in the Edit mode by an EDIT Keyword before any Type words are used. Any of the fields not marked with an asterisk (*) in Table 4.4.1 may be changed. All fields that are not changed, and in particular those marked with "*" must have the same information as the most recently read card of that type. This is true of all of the Type words. Information which is not changed must nevertheless be punched on the card. Note: Editing of the asterisked fields will <u>not</u> generate an immediate error message but will cause erroneous results.

If the asterisked fields are examined, it will be seen that only parametric editing is permitted. In other words, the strucutre (i.e., node and arc configuration) may not be changed.

Also, no data card types with suffixes A, B, or C are applicable during EDITS. Arcs and nodes in subsystems are referred to by their node numbers just as are non-subsystem arcs and nodes.

<u>ONE</u>- The type word ONE causes GRASP to read a type NEW-1 data card.

<u>NODE</u> - Causes a stream of Data Card Type NEW-2 to be read. Cards of this Type will be read until a blank of zero is encountered in Field 1.

<u>PARM</u> - Causes a stream of NEW-3 data cards to be read (parameter sets). Existing parameter sets can be modified and additional parameter sets can be created. NEW-3 cards are read until a zero or blank is encountered in Field 1.

<u>ARCS</u> - Causes reading of Type NEW-4 cards. At this point, recall that some parameter sets are modified by GRASP. After reading a network, GRASP "forgets" which parameter sets have been modified. Therefore, reference must be made to Table 4.3.1 when editing distribution types or parameter sets.

The following considerations are important:

1. If a PARM card has generated or changed a parameter set that applies to a distribution of a type that modifies its parameters, then an ARCS card <u>MUST</u> appear, and one card with that distribution type must be read (even though no changes are made on it). This will cause the proper modification of that parameter set.

2. If a distribution type on an edited NEW-4 card is changed to

one which modifies its parameter set, the parameter set will be modified when the card is read. If it is not desired that the parameter set be modified, then the parameter set number must be read as a negative number. (This would be desired if the parameter set had been modified in an earlier simulation of the same computer run, by virtue of the distribution appearing elsewhere in the network).

An additional consideration in modifying ARCS is that the NEW-4 card may not uniquely describe an arc in the network. See Figures 3.1.3, 3.1.6 and 3.1.8 for example.

The arcs between Nodes 2 and 3 can all be identical, so the NEW-4 Data Cards for these arcs are also identical. GRASP ranks the parallel arcs between any two nodes, however. Therefore, it is possible to uniquely specify which arc is to be changed.

Given a start and an end node, the arcs between these nodes are ranked in decreasing order of their probabilities (Field 1, NEW-4 cards). If there are ties, then they are ranked by the order in which they were originally read. Hence, the arc of highest probability is ranked 1, the next highest 2, etc. If there are ties, the first read is ranked before the second read, etc. To uniquely specify one of several parallel arcs, an additional variable must be included on NEW-4 cards when Editing ARCS. Table 4.4.5 describes the format.

CNOD - Permits changing of all fields on the NEW-6 data card.

Table 4.4.5 Addition to Data Card Type NEW-4 for Editing Arcs

Filed	Column	Format	Variable	Description
1-11	1-50	-	-	Same as NEW-4
12	79-80	I2	IRNK	Rank of the arc (Zero will be taken as one)

4.5 Examples of Data Input

4.5.1 Two Units in Parallel: The Example in Figure 3.3.5

Figure 4.5.1 shows the input data for a system with two units in parallel, previously discussed as the network in Figure 3.3.5. The first and the last lines are artificial, numbered 0.1 and 36.1. They are added in order to easily locate the column numbers of the different fields.

Line number 1 shows the keyword NEW, starting from column 1. This work begins every GRASP network. Several GRASP networks can be simulated in only one execution of the program. Refer to Figure 4.1.1 for the sequencing of the data cards. Lines 2, 3, and 4 contain the NEW-1 cards. Line 2 contains CARD-1 of NEW-1 and has the title of the GRASP model. Line 3 contains CARD-2 and has the following information:

Field 1	Col. 1-3	NPRT = 1	print option
Field 2	Col. 4-6	NSNK = 1	number of sink nodes
Field 3	Col. 7-9	NSTS = 3	number of statistic nodes
Field 4	Col. 10-12	MODI = 0	no modifications
Field 5	Col. 13-15	IGRF = 1	plot histograms

Fields 6 - 9 are left blank, so the variables NCND (number of C-nodes), NCTS (number of count types), NSBS (number of subsystem types), and IDME (dumping of the files option) are equal to zero.

Line 4 contains card 3 of NEW-1 and has the following information:

Field 1	Col. 1-10	ISED = 1113497	starting seed for random number generation.
Field 2	Col. 11-20	NRNS = 1	number of simulation runs.

Fields 3 - 7 are left blank so the remaining variables NSTR (run number for beginning of tracing), NETR (run number for end of tracing), TSTR (time for beginning of tracing), TETR (time for end of tracing) and SCAL (scale factor for distribution type 10) are equal to their default values.

```
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
NEW
TWO UNITS IN PARALLEL
  1  1  3  0  1
   1113497          1
  2  1  1 -1
  3     1  1
  4  1  1 -1
  5     1  1
  6  3  3  2      0.    -1.8
  7  3 -2 -2      0.    -1.8
  8  1  1  1
  9  2  1  1               F
  0
  1        5.        0.       100.        2.
  2        .1        0.       100.        1.4
  3        .5        0.       100.        1.
  4      500.
  0
          2  3  1  4
          2  6  0 10    -1
          2  7  0 10    -1
          3  2  3  4    -1
          3  6  0 10
          3  7  0 10
          4  5  2 12
          4  6  0 10    -1
          4  7  0 10    -1
          5  4  3  4    -1
          5  6  0 10
          5  7  0 10
          6  7  0 10    -1
          7  6  0 10
          8  9  4  1
          0
RUN
STOP
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
```

Figure 4.5.1 Input data for a two units in parallel system (see Figure 3.3.5)

Lines numbered 5 to 12 describe the nodes of the network. B indicates "Between" statistics will be collected at nodes 6 and 7, and the histograms will be automatically scaled by the program.

Line number 13 indicates the end of the node description cards. Lines numbered 14 to 17 give the parameters of the distributions. Line number 14 gives parameter set 1 and corresponds to an Erlang distribution with mean/K = 5., minimum = 0, maximum = 100 and K = 2. Line 15 has the parameters A = .1, minimum = 0, maximum = 0 and B = 1.4, and this corresponds to distribution number 12 which is the Weibull distribution (see Table 4.3.2). Line 16 gives the parameter set for the repair distribution which is Erland-1 (exponential) with a mean equal to 0.5, a minimum equal to 0 and a maximum equal to 100. Line 17 gives the parameter set corresponding to distribution type 1 which is just a constant. Hence, only field 2 is used and it gives the value of that constant. This constant is used for stopping the simulation.

ORIGINAL PAGE IS
OF POOR QUALITY

```
NEW
GRASP SIMULATION PROJECT : TWO UNITS IN PARALLEL

                              **NETWORK DESCRIPTION**

        NUMBER OF SINK NODES: NSNK =  1
        STATISTICS COLLECTED ON: NSTS =  3 NODES.
        ARE THERE MODIFICATIONS IN THE NETWORK?: MODI =  0
        TYPE OF HISTOGRAMS TO BE COLLECTED: IGRF =  1
        NUMBER OF ACCUM. NODE TYPES: NCNO =  0
        NUMBER OF COUNT TYPES: NCTS =  0
        NUMBER OF SUBSYSTEM TYPES: NSBS =  0
        DUMPING OF FILES OPTION: IDMP =  0
        SEED FOR INITIAL RANDOM NUMBER: ISED =    1113497
        NUMBER OF RUNS TO END THE SIMULATION: NRNS =          1
        RUN NUMBER TO BEGIN TRACING: NSTR =          0
        RUN NUMBER TO END TRACING: NETR =          0
        TIME TO BEGIN TRACING: TSTR =          0.
        TIME TO END TRACING: TETR =          0.
        SCALE FOR CONSTANT TIMES: SCAL =          1.

                              **NODE DESCRIPTION**

NODE   NUMBER     NUMBER OF RELEASES    A=AND                OUTPUT    STATISTICS TYPE
       RELEASES       FOR REPEAT     H= HALT . U=A U H        TYPE

  2       1              -1                                    D
  3       1               1                                    D
  4       1              -1                                    D
  5       1               1                                    D
  6       3               2                                    D              B
  7      -2              -2                                    D              B
  8       1               1                                    D
  9       1               1                                    D              F

        SOURCE NODE NUMBERS
            2    4    8

        SINK NODE NUMBERS
            9

        STATISTICS COLLECTED ALSO ON NODES
            7    6
```

Figure 4.5.2 (a) Echo for NEW-1 and NEW-2 cards

** ARC PARAMETERS **

PARAMETER SET	PARAMETERS 1	2	3	4
1	5.0000	0.0	100.0000	2.0000
2	0.1000	0.0	100.0000	1.4000
3	0.5000	0.0	100.0000	1.0000
4	500.0000	0.0	0.0	0.0

** ARC DESCRIPTION **

START NODE	END NODE	PARAMETER SET	DISTRIBUTION TYPE	COUNT TYPE	ARC NUMBER	PROBABILITY	SETUP COST	VARIABLE COST	ASSOCIATED COST C-NODE
2	3	1	4	0	0	1.0000	0.0	0.0	0
2	6	0	10	0	-1	1.0000	0.0	0.0	0
2	7	0	10	0	-1	1.0000	0.0	0.0	0
3	2	3	4	0	-1	1.0000	0.0	0.0	0
3	6	0	10	0	0	1.0000	0.0	0.0	0
3	7	0	10	0	0	1.0000	0.0	0.0	0
4	5	2	12	0	0	1.0000	0.0	0.0	0
4	6	0	10	0	-1	1.0000	0.0	0.0	0
4	7	0	10	0	-1	1.0000	0.0	0.0	0
5	4	3	4	0	-1	1.0000	0.0	0.0	0
5	6	0	10	0	0	1.0000	0.0	0.0	0
5	7	0	10	0	0	1.0000	0.0	0.0	0
6	7	0	10	0	-1	1.0000	0.0	0.0	0
7	6	0	10	0	0	1.0000	0.0	0.0	0
8	9	4	1	0	0	1.0000	0.0	0.0	0

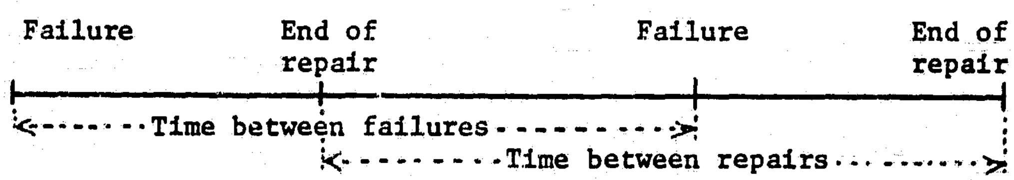
Figure 4.5.2 (continued)

(b) Echo for NEW-3 cards

(c) Echo for NEW-4 cards

A description of the arcs starts at line 19. Each arc is completely described by one card. Since there is no probabilistic nodes in the network, the probability associated with each arc is 1. No cost data was requested. Line 34 indicates the end of the arc descrition cards. The keywords RUN and STOP are self-explanatory.

A complete output of this example is given in Figure 4.5.2. The output starts with an echo check of the data to help locate any data input errors. Part (a) of Figure 4.5.2 shows the output corresponding to the basic run information and the node description cards. Notice that limited information diagrams are also given. Figures describing the output are presented sequentially as they appear in an actual run. Parts (b) and (c) of Figure 4.5.2 describe the parameters of the distributions, and each activity, respectively. The column headings are self-explanatory. The node summary in Figure 4.5.3 reveals that the mean time between repairs is 73.2 and the mean time between failures is 81.67. These quantities estimate the operating characteristics pictured below.



Statistically, they are expected to be equal. Their difference is due to the fact that we have unequal numbers of observations for nodes 6 and 7, and the simulation time (500) was not long enough to reach exact solutions. This is reflected in a large standard deviation for both results. The histograms in Figures 4.5.4 and 4.5.5 are very similar, but since only a few observations are generated, they don't give a complete picture of the actual distribution.

RUN

GRASP SIMULATION PROJECT : TWO UNITS IN PARALLEL

FINAL RESULTS FOR 1 SIMULATION RUN(S)

NODE	PROB./ COUNT	MEAN	STAND. DEV.	NO OF OBS.	MIN	MAX	NODE TYPE
9	1.0000	0.500000E+03	0.0	1.	0.500000E+03	0.500000E+03	F
7	1.0000	0.731997E+02	0.924509E+02	6.	0.188977E+02	0.260673E+03	B
6	1.0000	0.816739E+02	0.100799E+03	5.	0.187415E+02	0.260780E+03	B

Figure 4.5.3

Node summary for the network in Figure 3.3.5

STAT HISTOGRAM FOR NODE 7

OBSV FREQ	RELA FREQ	CUML FREQ	LOWER BOUND OF CELL
1	0.167	0.167	-INF
0	0.0	0.167	0.2265E+02
2	0.333	0.500	0.3020E+02
1	0.167	0.667	0.3775E+02
1	0.167	0.833	0.4530E+02
0	0.0	0.833	0.5285E+02
0	0.0	0.833	0.6040E+02
0	0.0	0.833	0.6795E+02
0	0.0	0.833	0.7550E+02
0	0.0	0.833	0.8305E+02
0	0.0	0.833	0.9060E+02
0	0.0	0.833	0.9815E+02
0	0.0	0.833	0.1057E+03
0	0.0	0.833	0.1132E+03
0	0.0	0.833	0.1208E+03
0	0.0	0.833	0.1283E+03
0	0.0	0.833	0.1359E+03
0	0.0	0.833	0.1434E+03
0	0.0	0.833	0.1510E+03
0	0.0	0.833	0.1585E+03
0	0.0	0.833	0.1661E+03
0	0.0	0.833	0.1736E+03
0	0.0	0.833	0.1812E+03
0	0.0	0.833	0.1887E+03
0	0.0	0.833	0.1963E+03
0	0.0	0.833	0.2038E+03
0	0.0	0.833	0.2114E+03
0	0.0	0.833	0.2189E+03
0	0.0	0.833	0.2265E+03
0	0.0	0.833	0.2340E+03
0	0.0	0.833	0.2416E+03
1	0.167	1.000	0.2491E+03
6			

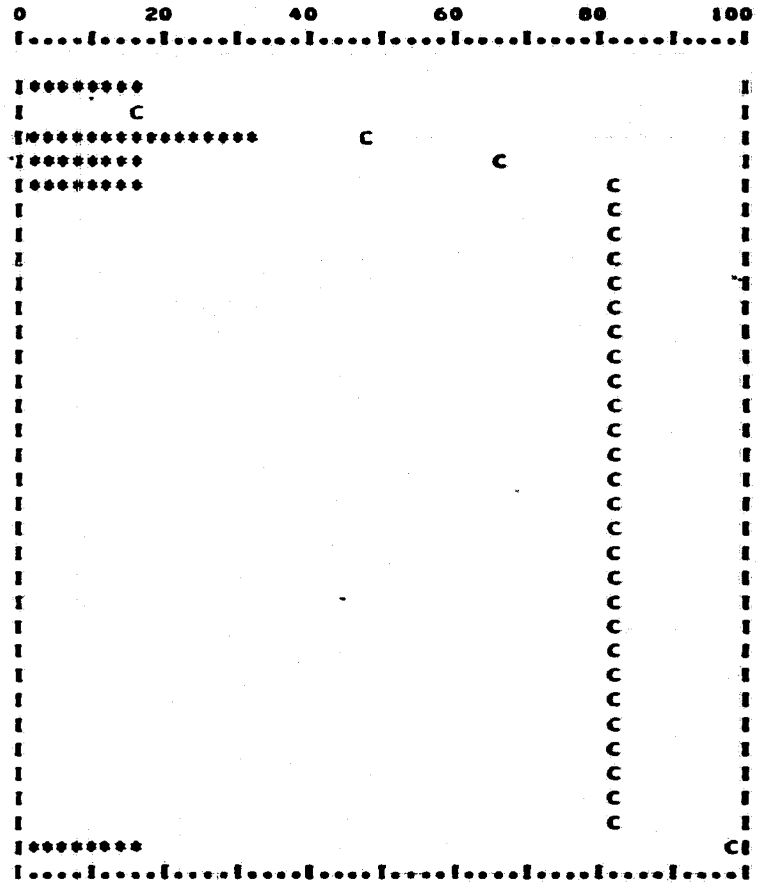


Figure 4.5.4

STAT HISTOGRAM FOR NODE 6

```
OBSV   RELA    CUML    LOWER BOUND     0        20        40        60        80       100
FREQ   FREQ    FREQ      OF CELL       I....I....I....I....I....I....I....I....I....I....I

   1   0.200   0.200       -INF        I**********                                       I
   0   0.0     0.200    0.2280E+02     I         C                                       I
   1   0.200   0.400    0.3040E+02     I**********         C                             I
   1   0.200   0.600    0.3800E+02     I**********                   C                   I
   1   0.200   0.800    0.4560E+02     I**********                             C         I
   0   0.0     0.800    0.5320E+02     I                                       C         I
   0   0.0     0.800    0.6080E+02     I                                       C         I
   0   0.0     0.800    0.6840E+02     I                                       C         I
   0   0.0     0.800    0.7600E+02     I                                       C         I
   0   0.0     0.800    0.8360E+02     I                                       C         I
   0   0.0     0.800    0.9120E+02     I                                       C         I
   0   0.0     0.800    0.9880E+02     I                                       C         I
   0   0.0     0.800    0.1064E+03     I                                       C         I
   0   0.0     0.800    0.1140E+03     I                                       C         I
   0   0.0     0.800    0.1216E+03     I                                       C         I
   0   0.0     0.800    0.1292E+03     I                                       C         I
   0   0.0     0.800    0.1368E+03     I                                       C         I
   0   0.0     0.800    0.1444E+03     I                                       C         I
   0   0.0     0.800    0.1520E+03     I                                       C         I
   0   0.0     0.800    0.1596E+03     I                                       C         I
   0   0.0     0.800    0.1672E+03     I                                       C         I
   0   0.0     0.800    0.1748E+03     I                                       C         I
   0   0.0     0.800    0.1824E+03     I                                       C         I
   0   0.0     0.800    0.1900E+03     I                                       C         I
   0   0.0     0.800    0.1976E+03     I                                       C         I
   0   0.0     0.800    0.2052E+03     I                                       C         I
   0   0.0     0.800    0.2128E+03     I                                       C         I
   0   0.0     0.800    0.2204E+03     I                                       C         I
   0   0.0     0.800    0.2280E+03     I                                       C         I
   0   0.0     0.800    0.2356E+03     I                                       C         I
   0   0.0     0.800    0.2432E+03     I                                       C         I
   1   0.200   1.000    0.2508E+03     I**********                                      CI
 ---                                   I....I....I....I....I....I....I....I....I....I....I
   5
```

Figure 4.5.5

4.5.2 Two Units in Series Example in Figure 3.4.2

Data input for Figure 3.4.2 is shown in Figure 4.5.6. A new feature of GRASP is demonstrated in this example; that is, the EDIT key. The example represents two identical units in series. Their life distribution is a log-normal with a mean equal to 50, a minimum equal to 0, a maximum equal to 300 and a standard deviation equal to 20. These parameters are specified in parameter set number 1. Their repair distribution is exponential with mean equal to 2 as specified in parameter set 2. After the network is simulated, the program will read the keyword EDIT and cause the network to be modified. The next line read is line number 36. It has the keywork ONE and directs the program to redefine cards NEW-1. Line numbers 37-39 are the revised cards. They are identical to lines 2-4 in most of the fields; only the random number seed, the trace and dump fields have been changed. The event trace is now requested and the file dump is suppressed. The output for this example includes the usual echo check of the input data; node summary; the histograms for nodes 6 and 7; and a file status at the end of the run. The node summary appears in Figure 4.5.7. The mean time between the releases of node 7 is 22.64, and the mean time between the releases of node 6 is 23.45. These quantities estimate the same random variable, which is the sum of the mean life time and the mean repair time. Other parts of the GRASP output are not included in this analysis. The RUN command at line 43 causes the revised network to be simulated. The output from the edited network is not included here. The requested file dump after execution will be presented later in section 5. A network can be edited more than once, and GRASP counts the number of times that a network is edited. The editing operation is cumulative. In other words, every time a part of the network is replaced by an edit operation, such a replacement is final, and subsequent edits operate on the revised network.

ORIGINAL PAGE IS
OF POOR QUALITY

```
 0.1   123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
 1.    NEW
 2.     TWO UNITS  IN  SERIES.
 3.      1   1   3   0   1   0   0   0  2
 4.       6277189          1
 5.      2   1   1 -1D
 6.      3       1  1D
 7.      4   1   1 -1D
 8.      5       1  1D
 9.      6   3  -2 -2D      0.    -1.8
10.      7   3   2  2D      0.    -1.8
11.      8   1   1  1D
12.      9   2   1  1D
13.      0
14.      1        50.          0        300.        20.
15.      2         2.         0.         50.         1.
16.      3       500.
17.      0
18.          1.   2   3   1   5
19.          1.   2   6   0  10     -1
20.          1.   2   7   0  10     -1
21.          1.   3   2   2   4     -1
22.          1.   3   6   0  10
23.          1.   3   7   0  10
24.          1.   4   5   1   5
25.          1.   4   6   0  10     -1
26.          1.   4   7   0  10     -1
27.          1.   5   4   2   4     -1
28.          1.   5   6   0  10
29.          1.   5   7   0  10
30.          1.   6   7   0  10
31.          1.   7   6   0  10     -1
32.          1.   8   9   3   1
33.               0
34.    RUN
35.    EDIT
36.    ONE
37.     TWO UNITS  IN  SERIES  (AFTER EDIT).
38.      1   1   3   0   1   0   0   0
39.      87654321          1           1           1
40.    PARM
41.      3        31.
42.      0
43.    RUN
44.    STOP
44.1   123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
```

Figure 4.5.6 Data Input for Figure 3.4.2

RUN

GRASP SIMULATION PROJECT : TWO UNITS IN SERIES.

FINAL RESULTS FOR 1 SIMULATION RUN(S)

NODE	PROB./ COUNT	MEAN	STAND. DEV.	NO OF OBS.	MIN	MAX	NODE TYPE
9	1.0000	0.500000E+03	0.0	1.	0.500000E+03	0.500000E+03	F
7	1.0000	0.226359E+02	0.104692E+02	20.	0.612207E+01	0.450352E+02	B
6	1.0000	0.234460E+02	0.113941E+02	21.	0.707178E+01	0.474114E+02	B

Figure 4.5.7 Node Summary for the Network in Figure 3.4.2

4.5.3 Data Input for the Standby System in Figure 3.5.5

The data input for this example is shown in Figure 4.5.8, and is self-explanatory. Both units have exponential lifetimes. The first unit has its parameters in parameter set 1 in line 16. It has a mean lifetime equal to 4. The second unit has its parameters in parameter set 3, for both units are identically distributed (exponential) and share parameter set 3, with a mean time to repair equal to 2. Parameter set 4 is used in arc (10,11) for stopping the simulation. Nodes 8 and 9 collect statistics on the cycle time of the system (sum of the lifetime and repair time). As expected, these node summaries are nearly the same, as shown in Figure 4.5.9. The plotted histograms are practically identical. These are shown in Figures 4.5.10 and 4.5.11. This output shows that the distribution of the cycle time has a relatively long tail with perhaps multiple modes.

```
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
NEW
 AN EXAMPLE OF A STANDBY SYSTEM.
  1  1  3  0  1  3  0  0
  87654321          1
  2  1  1 -20
  3     1  10
  4     1  10
  5    -1 -20
  6     1  10
  7     1  10
  8  3 -2 -20     0.    -1.8
  9  3  2  20     0.    -1.8
 10  1  1  10
 11  2  1  10              F
  0
  1         4.        0.      1000.       1.
  2         3.        0.      1000.       1.
  3         2.        0.      1000.       1.
  4       500.
  0
        1.  2  3  1  4
        1.  2  5  0 10
        1.  3  2  0 10    -1
        1.  3  4  2  4
        1.  3  5  0 10    -1
        1.  3  8  0 10
        1.  3  9  0 10
        1.  4  8  0 10    -1
        1.  4  9  0 10    -1
        1.  4  2  0 10    -1
        1.  5  6  3  4
        1.  5  2  0 10
        1.  6  5  0 10    -1
        1.  6  7  2  4
        1.  6  2  0 10    -1
        1.  6  8  0 10
        1.  6  9  0 10
        1.  7  8  0 10    -1
        1.  7  9  0 10    -1
        1.  7  5  0 10    -1
        1.  8  9  0 10
        1.  9  8  0 10    -1
        1. 10 11  4  1
            0
RUN
STOP
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
```

Figure 4.5.8 Data input for Figure 3.5.5

RUN

GRASP SIMULATION PROJECT : AN EXAMPLE OF A STANDBY SYSTEM.

FINAL RESULTS FOR 1 SIMULATION RUN(S)

NODE	PROB./ COUNT	MEAN	STAND. DEV.	NO OF OBS.	MIN	MAX	NODE TYPE
11	1.0000	0.500000E+03	0.0	1.	0.500000E+03	0.500000E+03	F
9	1.0000	0.724428E+01	0.729185E+01	67.	0.200928E+00	0.311179E+02	B
8	1.0000	0.725010E+01	0.755454E+01	67.	0.384033E+00	0.315017E+02	B

Figure 4.5.9 Node Summary for the Network Example in Figure 3.5.5

STAT HISTOGRAM FOR NODE 8

OBSV FREQ	RELA FREQ	CUML FREQ	LOWER BOUND OF CELL
5	0.075	0.075	-INF
6	0.090	0.164	0.9750E+00
12	0.179	0.343	0.1950E+01
7	0.104	0.448	0.2925E+01
4	0.060	0.507	0.3900E+01
9	0.134	0.642	0.4875E+01
3	0.045	0.687	0.5850E+01
3	0.045	0.731	0.6825E+01
2	0.030	0.761	0.7800E+01
0	0.0	0.761	0.8775E+01
2	0.030	0.791	0.9750E+01
0	0.0	0.791	0.1072E+02
2	0.030	0.821	0.1170E+02
3	0.045	0.866	0.1267E+02
1	0.015	0.881	0.1365E+02
0	0.0	0.881	0.1462E+02
0	0.0	0.881	0.1560E+02
0	0.0	0.881	0.1657E+02
1	0.015	0.896	0.1755E+02
0	0.0	0.896	0.1852E+02
0	0.0	0.896	0.1950E+02
0	0.0	0.896	0.2047E+02
2	0.030	0.925	0.2145E+02
1	0.015	0.940	0.2242E+02
0	0.0	0.940	0.2340E+02
0	0.0	0.940	0.2437E+02
1	0.015	0.955	0.2535E+02
1	0.015	0.970	0.2632E+02
0	0.0	0.970	0.2730E+02
0	0.0	0.970	0.2827E+02
0	0.0	0.970	0.2925E+02
2	0.030	1.000	0.3022E+02

67			

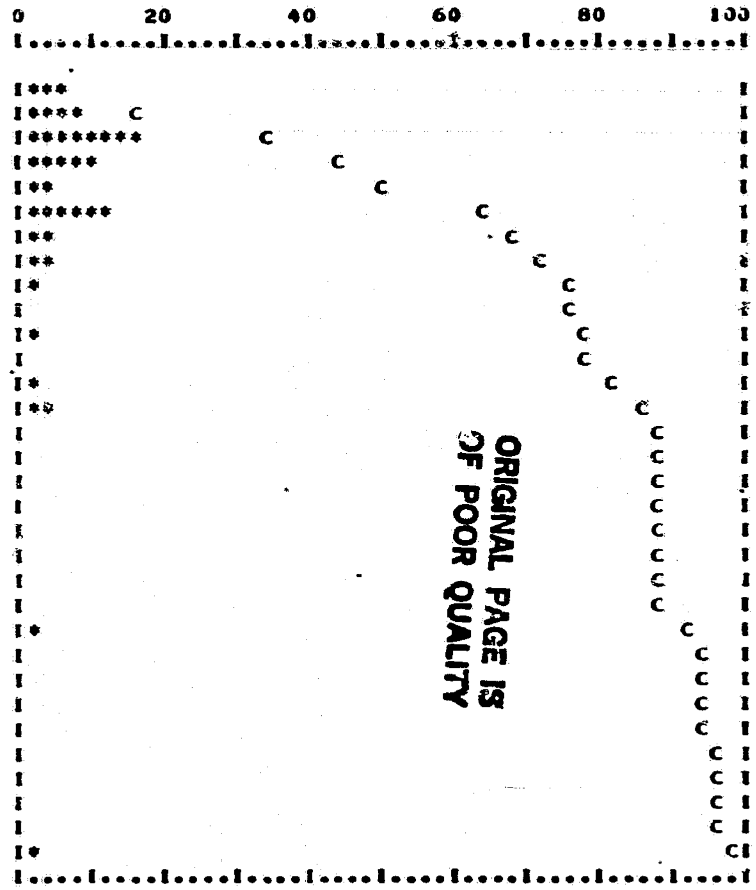


Figure 4.5.10

STAT HISTOGRAM FOR NODE 9

OBSV FREQ	RELA FREQ	CUML FREQ	LOWER BOUND OF CELL
3	0.045	0.045	-INF
13	0.194	0.239	0.9700E+00
6	0.090	0.328	0.1940E+01
9	0.134	0.463	0.2910E+01
3	0.045	0.507	0.3880E+01
6	0.090	0.597	0.4850E+01
5	0.075	0.672	0.5820E+01
2	0.030	0.701	0.6790E+01
3	0.045	0.746	0.7760E+01
1	0.015	0.761	0.8730E+01
1	0.015	0.776	0.9700E+01
1	0.015	0.791	0.1067E+02
3	0.045	0.836	0.1164E+02
1	0.015	0.851	0.1261E+02
2	0.030	0.881	0.1358E+02
0	0.0	0.881	0.1455E+02
0	0.0	0.881	0.1552E+02
1	0.015	0.896	0.1649E+02
0	0.0	0.896	0.1746E+02
0	0.0	0.896	0.1843E+02
1	0.015	0.910	0.1940E+02
0	0.0	0.910	0.2037E+02
1	0.015	0.925	0.2134E+02
1	0.015	0.940	0.2231E+02
0	0.0	0.940	0.2328E+02
1	0.015	0.955	0.2425E+02
0	0.0	0.955	0.2522E+02
0	0.0	0.955	0.2619E+02
2	0.030	0.985	0.2716E+02
0	0.0	0.985	0.2813E+02
0	0.0	0.985	0.2910E+02
1	0.015	1.000	0.3007E+02

67			

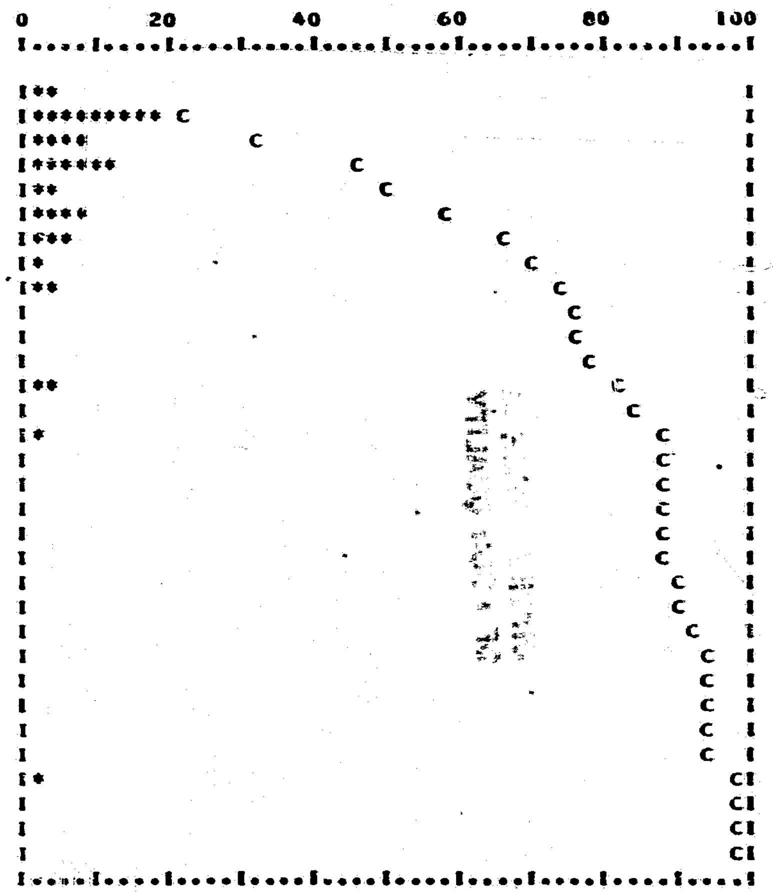


Figure 4.5.11

4.5.4 Data Input for the Example with Cost

The data input for the case without maintenance is shown in Figure 4.5.12. The only thing new in this example is the cost analysis. This is indicated in lines 18 and 19. The computer results are shown in Figure 4.5.13. They show that the average downtime is equal to 8 hours and the corresponding average cost is \$545.34. From this, the average cost per hour of downtime can be computed as 545.34/8 = \$68.17. The cost indicated for node 7 is the average cost for three years and is equal to \$8,430.89. The histogram for node 5 (Figure 4.5.14) shows how the downtimes are distributed. What probability distribution does that suggest?

The data input for the case with preventive maintenance is shown in Figure 4.5.15. Notice that the source seed for random number generation is used. This means that the pump operates under the same conditions as in the previous case and this will allow us to make meaningful comparisons. The computer results are shown in Figure 4.5.16. The mean down time in this case is estimated to be 3.17 hours - the corresponding average cost if \$238.82. The average cost per hour is then 238.82/3.17 = \$75.33.

```
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
NEW
FUEL PUMP WITHOUT MAINTENANCE.
  1  1  2  0  1  0  0  0  0
  87654321        100
  2  1  0  1
  3  4  1  1
  4     1  1
  5  3  1  1            -1.0 1
  6  1  0  1
  7  2  1  1
  0
  1    1700.00          .0 99999999.         1.
  2                     1.        12.
  3         1.5         1.         3.    .333333
  4 26280.0
  0
           2  3  1  4
           3  4  2  3                 500.0      0.0
           4  5  3  2                   0.0      3000
           5  2
           6  7  4  1
           0
RUN
STOP
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
```

Figure 4.5.12

Data input for the case

without maintenance.

ORIGINAL PAGE IS OF POOR QUALITY

GRASP SIMULATION PROJECT : FUEL PUMP WITHOUT MAINTENANCE.

FINAL RESULTS FOR 100 SIMULATION RUN(S)

NODE	PROB./ COUNT	MEAN	STAND. DEV.	NO OF OBS.	MIN	MAX	NODE TYPE
7	1.0000	0.262800E+05	0.0	100.	0.262800E+05	0.262800E+05	F
COST		0.843089E+04	0.199823E+04	100.	0.484804E+04	0.136156E+05	
5	1.0000	0.800778E+01	0.323156E+01	1546.	0.203125E+01	0.142734E+02	I
COST		0.545336E+03	0.960725E+01	1546.	0.529998E+03	0.581930E+03	

Figure 4.5.13 Computer results for the case without maintenance.

STAT HISTOGRAM FOR NODE 5

OBSV FREQ	RELA FREQ	CUML FREQ	LOWER BOUND OF CELL
5	0.003	0.003	-INF
30	0.019	0.023	0.2310E+01
46	0.030	0.052	0.2695E+01
50	0.032	0.085	0.3080E+01
65	0.042	0.127	0.3465E+01
60	0.039	0.166	0.3850E+01
53	0.034	0.200	0.4235E+01
54	0.035	0.235	0.4620E+01
50	0.032	0.267	0.5005E+01
50	0.032	0.299	0.5390E+01
51	0.033	0.332	0.5775E+01
51	0.033	0.365	0.6160E+01
61	0.039	0.405	0.6545E+01
50	0.032	0.437	0.6930E+01
57	0.037	0.474	0.7315E+01
53	0.034	0.508	0.7700E+01
61	0.039	0.548	0.8085E+01
53	0.034	0.532	0.8470E+01
50	0.032	0.614	0.8855E+01
59	0.038	0.653	0.9240E+01
41	0.027	0.679	0.9625E+01
56	0.036	0.715	0.1001E+02
50	0.032	0.748	0.1039E+02
53	0.034	0.782	0.1078E+02
52	0.034	0.816	0.1116E+02
69	0.045	0.860	0.1155E+02
39	0.025	0.886	0.1193E+02
45	0.029	0.915	0.1232E+02
56	0.036	0.951	0.1270E+02
47	0.030	0.981	0.1309E+02
24	0.016	0.997	0.1347E+02
5	0.003	1.000	0.1386E+02
1546			

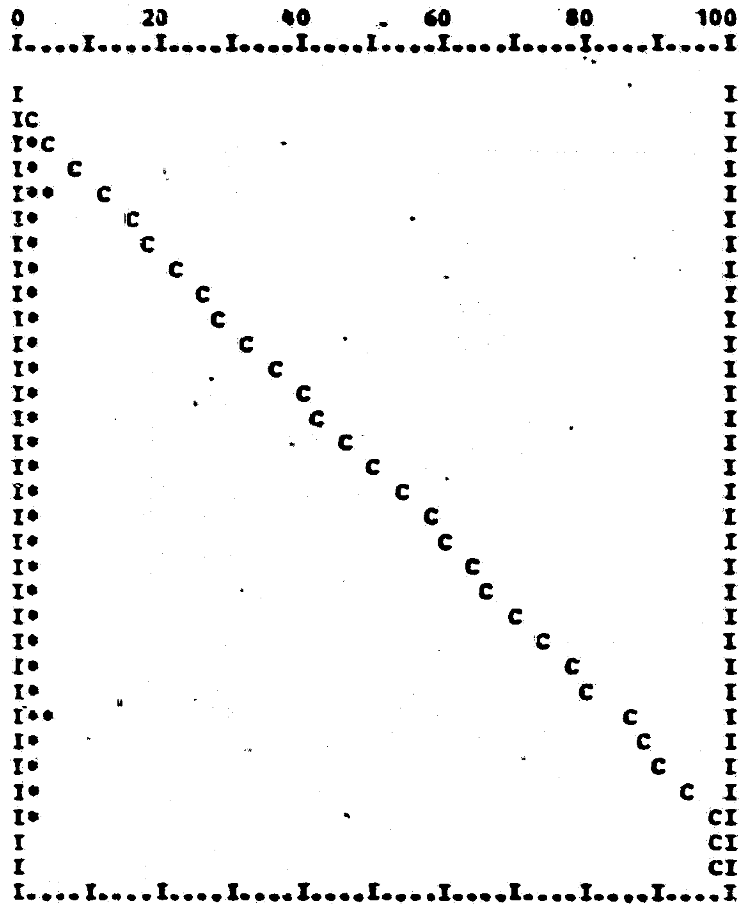


Figure 4.5.14

The average cost for three years is equal to \$9,531.74. This figure is higher than in the first case. In other words, it is more expensive to operate the pump under the preventive maintenance policy. The histogram of the down-times is shown in Figure 4.5.17. They suggest very much an Erlang or Weibull distribution while in the previous case, the exponential distribution seems to be the first choice. Goodness of fit tests must be performed to confirm this. Another interesting statistic to look at in comparing both models would be the total downtime. It is not shown in the computer results but it can easily be computed. For the first case it is equal to :

$$8.00778 \times 1546/100 = 123.80 \text{ hours}$$

and for the second case it is equal to:

$$3.16798 \times 3989/100 = 126.37 \text{ hours}$$

The results are summarized below.

policy	average downtime	average cost for a downtime period	average total downtime	average total cost of downtime	average cost per hour of downtime	downtime distribution
without maintenance	8.01 hrs.	\$545.34	123.80 hrs.	\$8430.89	\$68.17	?
with maintenance	3.17 hrs.	\$238.82	126.37 hrs.	\$9531.74	\$75.33	?

```
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
NEW
FUEL PUMP WITH PREVENTIVE MAINTENANCE.
  1  1  2  1  1  0  0  0  0
  87654321        100
  2  1  0  1
  3  4  1  1 H
  4     1  1
  5     1  1
  6     1  1P
  7     1  1
  8     1  1
  9  3  1  1          -1.0 I
 10  1  0  1
 11  2  1  1
  0
  1    3400.00         .0 99999999.         1.
  2                    1.       12.
  3         1.5        1.        3.    .333333
  4 26290.0
  5   720.0
  6         .75        .5        1.    .0833
  7         1.5        1.        2.    .1667
  8        1.25        1.       1.5    .0833
  0
           2  3 .1  4      1
           2  3 .5  1      2
           3  4
           5  6  6  2                      20.0
  .2794    6  9  7  2            450.0     20.0
  .7206    6  9  8  2                      20.0
           7  8  2  3            500.0      0.0
           8  9  3  2              0.0     30.0
           9  2
          10 11  4  1
           0
  1  4  7  5  4
  2  4  5  7  4
  0
RUN
STOP
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
```

Figure 4.5.15 Data Input for the Case with Preventive Maintenance

GRASP SIMULATION PROJECT : FUEL PUMP WITH PREVENTIVE MAINTENANCE.

FINAL RESULTS FOR 100 SIMULATION RUN(S)

NODE	PROB./ COUNT	MEAN	STAND. DEV.	NO OF OBS.	MIN	MAX	NODE TYPE
11	1.0000	0.262800E 05	0.000000E 00	100.	0.262800E 05	0.262800E 05	F
COST		0.953174E 04	0.177129E 04	100.	0.537340E 04	0.147531E 05	
9	1.0000	0.316798E 01	0.268040E 01	3989.	0.160938E 01	0.139329E 02	I
COST		0.238825E 03	0.236024E 03	3989.	0.323008E 02	0.574316E 03	

C-3

Figure 4.5.16 Computer Results for the Case with Preventive Maintenance

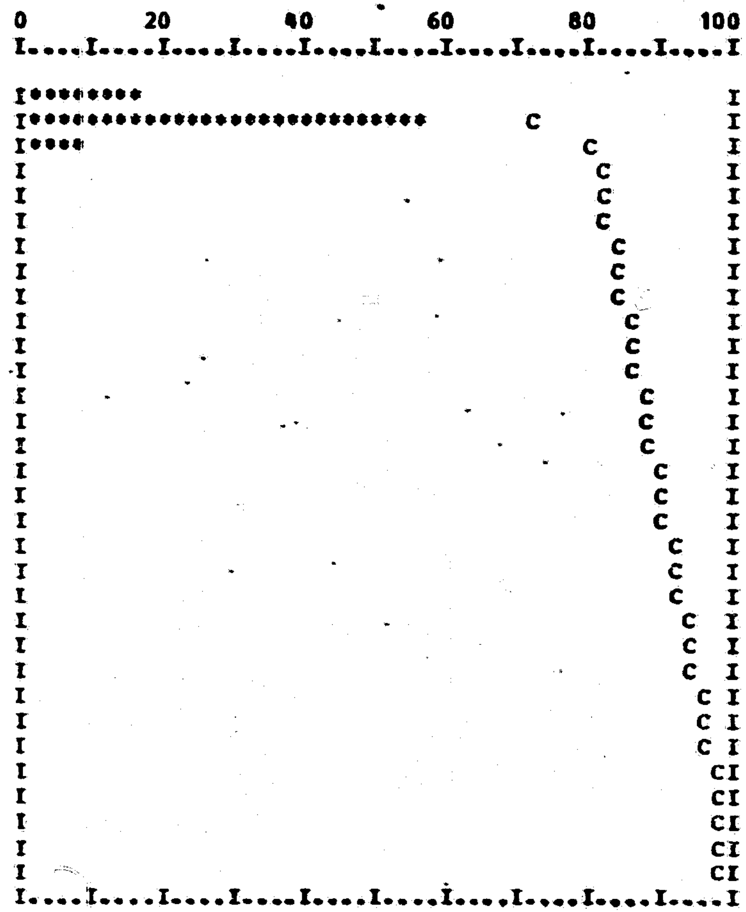
ORIGINAL PAGE IS OF POOR QUALITY

ORIGINAL PAGE IS
OF POOR QUALITY

STAT HISTOGRAM FOR NODE 9

OBSV FREQ	RELA FREQ	CUML FREQ	LOWER BOUND OF CELL
663	0.166	0.166	-INF
2241	0.562	0.728	0.1925E 01
352	0.088	0.816	0.2310E 01
28	0.007	0.823	0.2695E 01
16	0.004	0.827	0.3080E 01
27	0.007	0.834	0.3465E 01
38	0.010	0.844	0.3850E 01
19	0.005	0.848	0.4235E 01
30	0.008	0.856	0.4620E 01
26	0.007	0.862	0.5005E 01
34	0.009	0.871	0.5390E 01
27	0.007	0.878	0.5775E 01
28	0.007	0.885	0.6160E 01
17	0.004	0.889	0.6545E 01
23	0.006	0.895	0.6930E 01
29	0.007	0.902	0.7315E 01
28	0.007	0.909	0.7700E 01
25	0.006	0.915	0.8085E 01
23	0.006	0.921	0.8470E 01
32	0.008	0.929	0.8855E 01
33	0.008	0.937	0.9240E 01
23	0.006	0.943	0.9625E 01
21	0.005	0.948	0.1001E 02
24	0.006	0.954	0.1039E 02
26	0.007	0.961	0.1078E 02
28	0.007	0.968	0.1116E 02
24	0.006	0.974	0.1155E 02
25	0.006	0.980	0.1193E 02
23	0.006	0.986	0.1232E 02
28	0.007	0.993	0.1270E 02
22	0.006	0.998	0.1309E 02
6	0.002	1.000	0.1347E 02

3989			



Fibure 4.5.17

4.5.5 Data Input for the Complex Example in 3.8

This section will present the complete example given in Figure 3.8.2. The Level 3(a) and 3(b) systems will be simulated separately and their TBF and repair distributions determined. The PUNCH Keyword will be used in the Level 3 simulations to punch the TBF and repair distributions. These empirical distributions will be used as input to Level 2 where the HIST Keywork will be utilized. The Level 1 and 2 simulations will be performed in one computer run by making use of the SAVE Keyword. Also, the data input for Levels 1 and 2 will provide examples of subsystem generation.

Figure 4.5.18 is the data input for Level 3(a).

Another default value has been used on the NEW-2 cards (lines 5 to 32). The node type (column 13) specifies probabilistic or deterministic output. If this field is left blank, (as in Figure 4.5.18), GRASP will assume deterministic output.

Finally, the PUNCH Keywords lines 88 and 89 punch the TBF and repair distributions as empirical distributions 1 and 2, respectively. Recall from Figure 3.8.4 that node 26 collects the TBF distribution and node 27 collects the repair distribution.

Node summaries and histograms for Level 3(a) are shown in Figures 4.5.19, 20, and 21. The node summaries reveal that the mean time between failures for Level 3(a) is 7.9 time units (node 26) and the mean time to repair is 19.57 time units.

The empirical distributions that are produced from these nodes can be derived from the histograms as follows. The cell probabilities are given in the "Relative Frequency" column. The lower limits of the cells are found in the "LOWER BOUND OF CELL" column. The cell width can be found by subtracting two successive cell lower bounds, and the lower limit of the first cell (for empirical distribution purposes) is found by subtracting this cell width from the lower bound of the second cell.

The data input, node summaries, and node histograms for Level 3(b) are shown in Figures 4.5.22, 23, 24, and 25. The mean time between failures for Level 3(b) is 8.635 time units, and the mean repair time is 0.2365.

An additional default value has been used for input to Level 3(b). The repair distribution for unit 6 is (negative) exponential (Erlang-1) with parameters in Parameter Set 2. Normally, we would expect a "1.0" in field 5 of line 31, Figure 4.5.22. This would indicate that "1" in "Erlang-1". However, when this field is blank, as in line 31, GRASP assumes Erlang-1. Hence, exponential distributions can be obtained by specifying Distribution Type 4 and using only the mean, minimum value, and maximum value in the Parameter Set.

```
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
NEW
 COMPLEX EXAMPLE L3A.
  1  1  3  1  1
  16942811           1
  2  1  1  1
  3     1  1
  4     1  1
  5  1  1  1
  6     1  1
  7     1  1
  8  1  1  1
  9     1  1
 10     1  1
 11  1  1  1
 12     1  1
 13     1  1
 14  1  1  1
 15     1  1
 16     1  1
 17     2  2
 18     1  1
 19     1  1
 20     1  1
 21     1  1
 22     4  4
 23     1  1
 24     1  1
 26  3  2  2            -1.0
 25  1  1  1
 27  3  2  2      .04   .040
 28  2999999               F
  0
  1         25.        0      500.      18.
  2           .1       0       10.       1.
  3         25.        0      500.       1.
  4           .1       0       10.
  5         15.        0      500.       2.
  6           .2       0       20.       .4
  7          5.        0      100.       2.
  8           .1       0       .33      .04
  0
             2  3  1  8
             3  4  2  4
             3 20  0 10
             4  2  0 10
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
             4 22  0 10
             5  6  3  4
             6  7  4  4
             6 20  0 10
             7  5  0 10
             7 22  0 10
             8  9  5  4
             9 10  6  8
             9 20  0 10
            10  8  0 10
            10 22  0 10
            11 12  7  8
            12 13  8  2
            12 17  0 10
            13 11  0 10
            13 17  0 10   -1
            13 18  0 10
            14 15  7  8
            15 16  8  2
            15 17  0 10
            16 14  0 10
            16 17  0 10   -1
            16 18  0 10
            17 20  0 10    1
            19 22  0 10    2
            20 22  0 10   -1
            20 24  0 10    3
            21 22  0 10   -1
            22 23  0 10    4
            23 26  0 10
            23 27  0 10
            24 26  0 10
            24 27  0 10
            25 26  0 10
            26 28  0 10
             0
   1 18 19
   2 19 18
   3 20 21
   4 21 20
   0
RUN
PUNCH   26   AS   1
PUNCH   27   AS   2
STOP
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
```

ORIGINAL PAGE IS OF POOR QUALITY

Figure 4.5.18 Data Input for Level 3(a)

FINAL RESULTS FOR 1 SIMULATION RUN(S)

NODE	PROB./COUNT	MEAN	STD.DEV.	NO OF OBS.	MIN.	MAX.	NODE TYPE
29	1.0000	0.2742E+05	0.0	1.	0.2742E+05	0.2742E+05	F
26	1.0000	0.7900E+01	0.7357E+01	999.	0.3906E-02	0.4285E+02	D
27	1.0000	0.1957E+02	0.1750E+03	998.	0.0	0.5280E+04	D

Figure 4.5.19 Node Summary for Level 3(a)

STAT HISTOGRAM FOR NODE 26

OBSV FREQ	RELA FREQ	CUML FREQ	LOWER BOUND OF CELL
197	0.197	0.197	-INF
85	0.085	0.282	0.1350E+01
95	0.095	0.377	0.2700E+01
84	0.084	0.461	0.4050E+01
78	0.078	0.540	0.5400E+01
65	0.065	0.605	0.6750E+01
74	0.074	0.679	0.8100E+01
55	0.055	0.734	0.9450E+01
43	0.043	0.777	0.1080E+02
33	0.033	0.810	0.1215E+02
34	0.034	0.844	0.1350E+02
31	0.031	0.875	0.1485E+02
20	0.020	0.895	0.1620E+02
22	0.022	0.917	0.1755E+02
13	0.013	0.930	0.1890E+02
13	0.013	0.943	0.2025E+02
9	0.009	0.952	0.2160E+02
6	0.006	0.958	0.2295E+02
7	0.007	0.965	0.2430E+02
9	0.009	0.974	0.2565E+02
4	0.004	0.978	0.2700E+02
3	0.003	0.981	0.2835E+02
1	0.001	0.982	0.2970E+02
8	0.008	0.990	0.3105E+02
3	0.003	0.993	0.3240E+02
3	0.003	0.996	0.3375E+02
0	0.0	0.996	0.3510E+02
1	0.001	0.997	0.3645E+02
1	0.001	0.998	0.3780E+02
0	0.0	0.998	0.3915E+02
0	0.0	0.998	0.4050E+02
2	0.002	1.000	0.4185E+02

999			

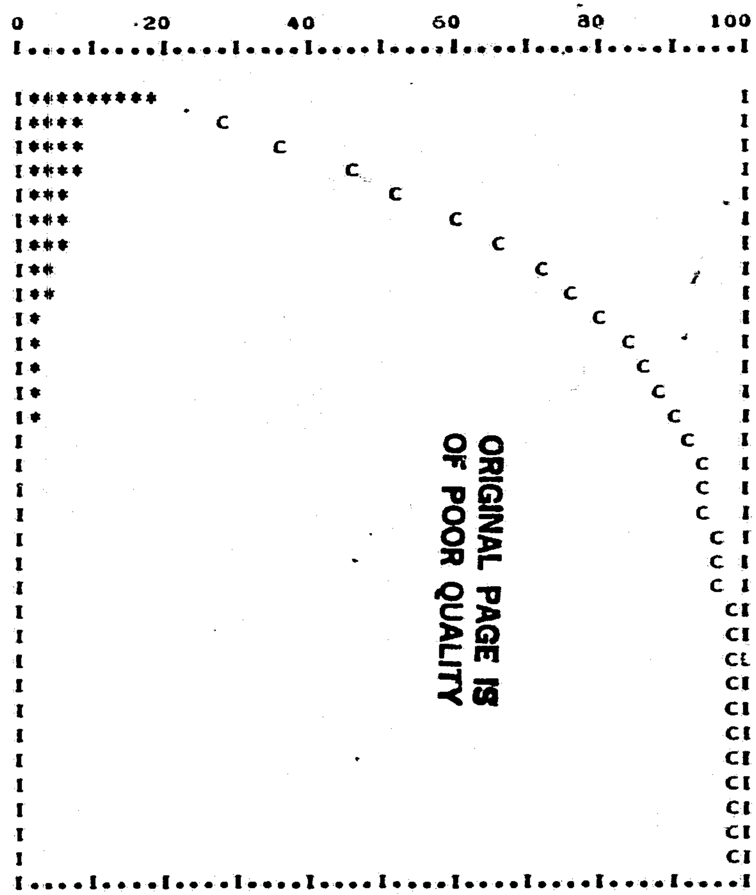


Figure 4.5.20 TBF Histogram for Level 3(a)

ORIGINAL PAGE IS OF POOR QUALITY

STAT HISTOGRAM FOR NODE 27

OBSV FREQ	RELA FREQ	CUML FREQ	LOWER BOUND OF CELL
280	0.281	0.281	-INF
136	0.136	0.417	0.4000E-01
93	0.093	0.510	0.8000E-01
42	0.042	0.552	0.1200E+00
43	0.043	0.595	0.1600E+00
33	0.033	0.628	0.2000E+00
16	0.016	0.644	0.2400E+00
15	0.015	0.659	0.2800E+00
16	0.016	0.675	0.3200E+00
8	0.008	0.683	0.3600E+00
9	0.009	0.692	0.4000E+00
4	0.004	0.696	0.4400E+00
4	0.004	0.700	0.4800E+00
5	0.005	0.705	0.5200E+00
2	0.002	0.707	0.5600E+00
3	0.003	0.710	0.6000E+00
4	0.004	0.714	0.6400E+00
1	0.001	0.715	0.6800E+00
2	0.002	0.717	0.7200E+00
1	0.001	0.718	0.7600E+00
1	0.001	0.719	0.8000E+00
4	0.004	0.723	0.8400E+00
0	0.0	0.723	0.8800E+00
1	0.001	0.724	0.9200E+00
0	0.0	0.724	0.9600E+00
0	0.0	0.724	0.1000E+01
3	0.003	0.727	0.1040E+01
2	0.002	0.729	0.1080E+01
0	0.0	0.729	0.1120E+01
1	0.001	0.730	0.1160E+01
0	0.0	0.730	0.1200E+01
269	0.270	1.000	0.1240E+01

998			

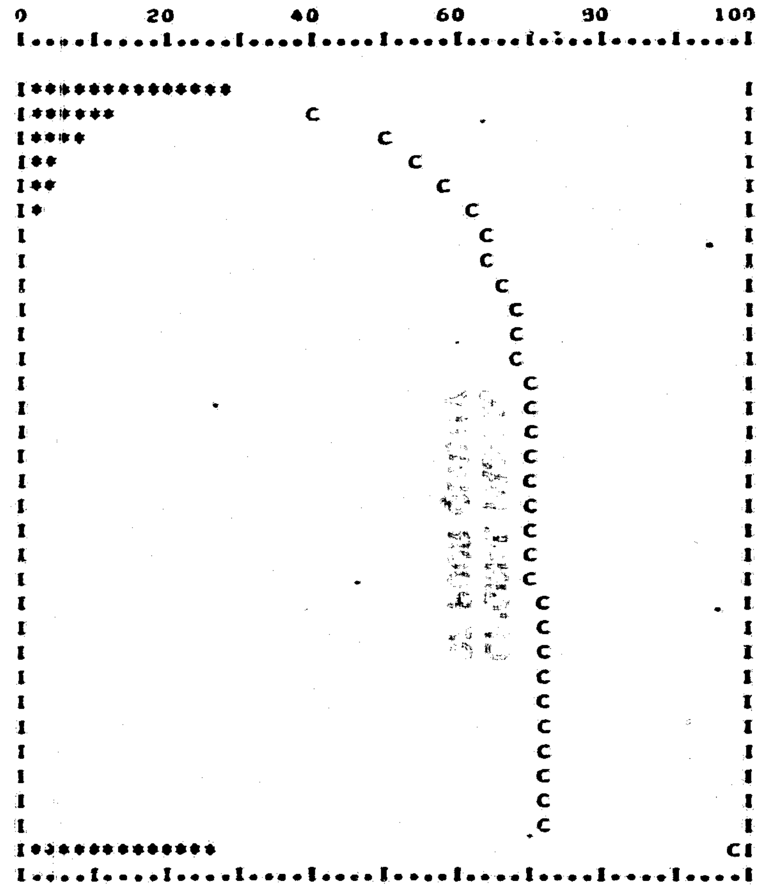


Figure 4.5.21 Repair Time Histogram for Level 3(a)

The data input for Levels 2 and 1 are shown in Figure 4.5.26. Recall that both of these networks have been simulated during the same computer run. The information in this figure down to the second NEW card (line 111) is for Level 2. This network has been input using subsystems. Refer to Figure 3.8.7. Non-subsystem nodes in this network are nodes numbered 80 to 88. The other nodes form three copies of one subsystem type. Nodes 53 to 61 are the first copy. The others are nodes 62 to 70 and 71 to 79. Notice that the subsystem nodes are numbered consecutively.

Line numbers in the following discussion are those in Figure 4.5.26. Non-subsystem nodes are input first on NEW-2 cards (lines 5 to 14). Line 15 is the NEW-2A card for the subsystem. The three fields specify that there are nine nodes in the subsystem, three copies are desired, and there are 12 subsystem arcs in each copy. Since this is the first NEW-2A card encountered, the subsystem following this card will be designated "Subsystem Type 1". Lines 16-25 are NEW-2B cards that describe the nodes in the first copy of the subsystem type, i.e., the copy with the lowest node numbers.

Each copy of the subsystem is identical with respect to its nodes. Hence, fields 12 (columns 48-50) of the NEW-2B cards are all blank. The result is that three identical copies of the subsystem nodes will be generated. No NEW-2C cards will be read. Lines 26-28 are the parameter sets for units 10 and 11. Lines 0.1, 40.1, 75.1, 75.2, 120.1 and 162.1 have been added so that column numbers can easily be identified.

Lines 29 to 45 are NEW-4 cards for the non-subsystem arcs. Recall that these are arcs with at least one end at a non-subsystem node. Lines 46 to 58 are NEW-4A cards for the subsystem arcs in the first copy of Subsystem Type 1. Note that the parameter on the arcs vary from copy to copy, however. For example, arc 53-54 (Figure 3.8.7) in copy 1 has time characteristic "(1, 13)"

ORIGINAL PAGE IS
OF POOR QUALITY

```
NEW
 COMPLEX EXAMPLE L38
  1  1  3  1  1
   6691285           1
 29  1  1  1
 30     1  1
 31     1  1
 32  1  1  1
 33     1  1
 34     1  1
 35  1  1  1
 36     1  1
 37     1  1
 38  1  1  1
 39     1  1
 40     1  1
 41     2  2
 42     1  1
 43     1  1
 44     1  1
 45     1  1
 46     3  3
 47     1  1
 48     1  1
 49  1  1  1
 50  3  2  2                -1.D
 51  3  2  2                -1.D
 52  2999999                   F
  0
  1          20.           0      300.       15.
  2           .2           0       20.
  3          11.           0      300.        2.
  4           .2           0       20.
  5           5.           0      100.
  6           .1           0        1.
  0
            29 30  1  6
            30 31  2  4
            30 44  0 10
            31 29  0 10
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
            31 46  0 10
            32 33  3  4
            33 34  4  4
            33 44  0 10
            34 32  0 10
            34 46  0 10
            35 36  5  4
            36 37  6 11
            36 41  0 10
            37 35  0 10
            37 42  0 10
            37 41  0 10    -1
            38 39  5  4
            39 40  6 11
            39 41  0 10
            40 38  0 10
            40 42  0 10
            40 41  0 10    -1
            41 44  0 10     1
            43 46  0 10     2
            44 48  0 10     3
            44 46  0 10    -1
            45 46  0 10    -1
            46 47  0 10     4
            47 50  0 10
            47 51  0 10
            48 50  0 10
            48 51  0 10
            49 50  0 10
            50 52  0 10
             0
  1 42 43
  2 43 42
  3 44 45
  4 45 44
  0
RUN
PUNCH   50    AS     3
PUNCH   51    AS     4
STOP
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
```

FINAL RESULTS FOR 1 SIMULATION RUN(S)

NODE	PROB./COUNT	MEAN	STD.DEV.	NO OF OBS.	MIN.	MAX.	NODE TYPE
52	1.0000	0.8862E+04	0.0	1.	0.8362E+04	0.8862E+04	F
50	1.0300	0.8635E+01	0.7319E+01	999.	0.3906E-02	0.4561E+02	D
51	1.0000	0.2365E+00	0.5769E+00	998.	0.1221E-02	0.1095E+02	D

Figure 4.5.23 Node Summaries for Level 3(b)

ORIGINAL PAGE IS
OF POOR QUALITY

STAT HISTOGRAM FOR NODE 50

OBSV FREQ	RELA FREQ	CUML FREQ	LOWER BOUND OF CELL
115	0.115	0.115	-INF
109	0.109	0.224	0.1450E+01
113	0.113	0.337	0.2900E+01
102	0.102	0.439	0.4350E+01
93	0.093	0.533	0.5800E+01
76	0.076	0.609	0.7250E+01
69	0.069	0.678	0.8700E+01
58	0.058	0.736	0.1015E+02
40	0.040	0.776	0.1160E+02
38	0.038	0.814	0.1305E+02
30	0.030	0.844	0.1450E+02
31	0.031	0.875	0.1595E+02
26	0.026	0.901	0.1740E+02
23	0.023	0.924	0.1885E+02
15	0.015	0.939	0.2030E+02
12	0.012	0.951	0.2175E+02
10	0.010	0.961	0.2320E+02
10	0.010	0.971	0.2465E+02
7	0.007	0.978	0.2610E+02
5	0.005	0.983	0.2755E+02
3	0.003	0.986	0.2900E+02
3	0.003	0.989	0.3045E+02
3	0.003	0.992	0.3190E+02
0	0.0	0.992	0.3335E+02
0	0.0	0.992	0.3480E+02
1	0.001	0.993	0.3625E+02
3	0.003	0.996	0.3770E+02
0	0.0	0.996	0.3915E+02
1	0.001	0.997	0.4060E+02
2	0.002	0.999	0.4205E+02
0	0.0	0.999	0.4350E+02
1	0.001	1.000	0.4495E+02

999			

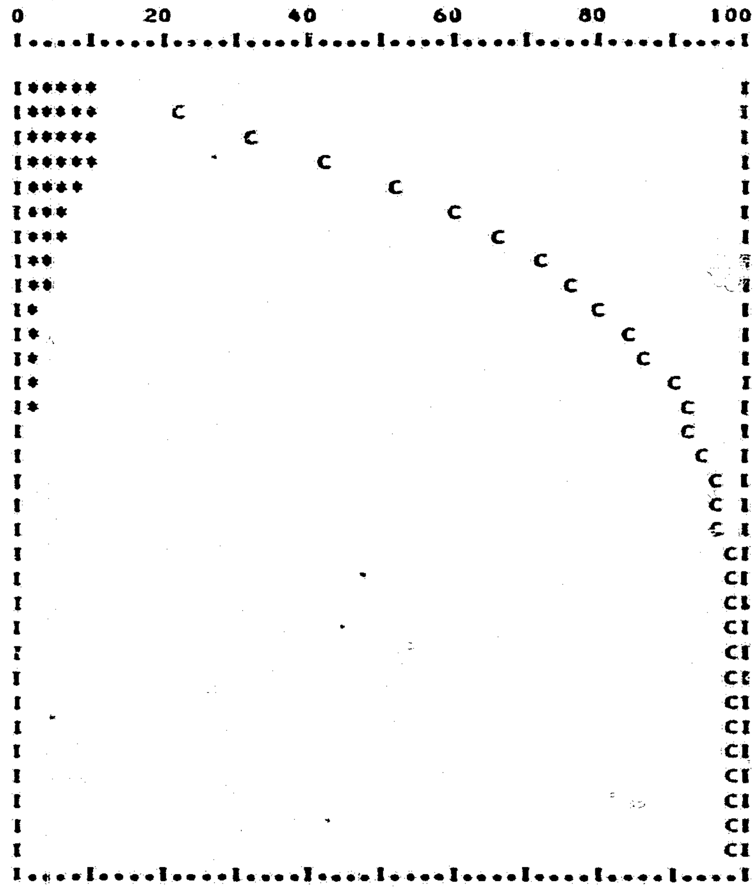


Figure 4.5.24 TBF Histogram for Level 3(b)

ORIGINAL PAGE IS
OF POOR QUALITY

```
STAT HISTOGRAM FOR NODE   51

OBSV   RELA    CUML    LOWER BOUND    0        20        40        60        80        100
FREQ   FREQ    FREQ      OF CELL      I....I....I....I....I....I....I....I....I....I....I

 854   0.856   0.856      -INF        I*******************************************       I
 108   0.108   0.964   0.3450E+00     I*****                                           C I
  20   0.020   0.984   0.6900E+00     I*                                                CI
   5   0.005   0.989   0.1035E+01     I                                                 CI
   1   0.001   0.990   0.1380E+01     I                                                 CI
   1   0.001   0.991   0.1725E+01     I                                                 CI
   0   0.0     0.991   0.2070E+01     I                                                 CI
   1   0.001   0.992   0.2415E+01     I                                                 CI
   2   0.002   0.994   0.2760E+01     I                                                 CI
   0   0.0     0.994   0.3105E+01     I                                                 CI
   0   0.0     0.994   0.3450E+01     I                                                 CI
   1   0.001   0.995   0.3795E+01     I                                                 CI
   0   0.0     0.995   0.4140E+01     I                                                 CI
   0   0.0     0.995   0.4485E+01     I                                                 CI
   1   0.001   0.996   0.4830E+01     I                                                 CI
   0   0.0     0.996   0.5175E+01     I                                                 CI
   1   0.001   0.997   0.5520E+01     I                                                 CI
   1   0.001   0.998   0.5865E+01     I                                                 CI
   0   0.0     0.998   0.6210E+01     I                                                 CI
   0   0.0     0.998   0.6555E+01     I                                                 CI
   0   0.0     0.998   0.6900E+01     I                                                 CI
   0   0.0     0.998   0.7245E+01     I                                                 CI
   0   0.0     0.998   0.7590E+01     I                                                 CI
   1   0.001   0.999   0.7935E+01     I                                                 CI
   0   0.0     0.999   0.8280E+01     I                                                 CI
   0   0.0     0.999   0.8625E+01     I                                                 CI
   0   0.0     0.999   0.8970E+01     I                                                 CI
   0   0.0     0.999   0.9315E+01     I                                                 CI
   0   0.0     0.999   0.9660E+01     I                                                 CI
   0   0.0     0.999   0.1000E+02     I                                                 CI
   0   0.0     0.999   0.1035E+02     I                                                 CI
   1   0.001   1.000   0.1069E+02     I                                                 CI
 ---                                  I....I....I....I....I....I....I....I....I....I....I
 998
```

Figure 4.5.25 Repair Histogram for Level 3(b)

```
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
NEW
 COMPLEX EXAMPLE L2.
  1  1  3  1  1  0  0  1
   1629947          1
 80      1  1
 81      1  1
 82      3  3
 83      1  1
 84      1  1
 85   1  1  1
 86   3  2  2          -1.0
 87   7  2  2          -1.0
 88  2500500              F
  0
    9    3   12
 53   1  1  1
 54      1  1
 55      1  1
 56   1  1  1
 57      1  1
 58      1  1
 59      2  2
 60      1  1
 61      1  1
  0
  1       - 2.        0         50.        2.
  2         .1        0         10.
  0
          61 82   0 10     2
          59 80   0 10     1
          70 82   0 10     4
          68 80   0 10     3
          79 82   0 10     6
          77 80   0 10     5
          92 83   0 10     8
          80 82   0 10    =1
          90 94   0 10     7
          81 82   0 10    -1
          83 86   0 10
          83 87   0 10
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
          84 86   0 10
          84 87   0 10
          85 86   0 10
          86 88   0 10
           0
          53 54   1 13                        2
          54 55   2 13                        2
          54 59   0 10
          55 53   0 10
          55 60   0 10
          55 39   0 10    -1
          56 57   1 13                        2
          57 58   2 13                        2
          57 59   0 10
          58 59   0 10    -1
          58 60   0 10
          58 56   0 10
           0
          62 63   3 13                        3
          63 64   4 13                        3
          65 66   3 13                        3
          66 67   4 13                        3
          71 72   1  4
          72 73   2  4
          74 75   1  4
          75 76   2  4
  1 60 61
  2 61 60
  3 69 70
  4 70 69
  5 78 79
  6 79 78
  7 80 81
  8 81 80
  0
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
```


Figure 4.5.26 Data Input for Levels 1 and 2

ORIGINAL PAGE IS
OF POOR QUALITY

```
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
HIST
    1    320.1350E+01
.1972.0851.0951.0841.0781.0651.0741.0551.0430.0330.0340.0310.0200.0220.0130.0130
.0090.0060.0070.0090.0040.0030.0010.0080.0030.0030.0    .0010.0010.0    .0    .0020
0.0        0.1350E+010.2700E+010.4050E+010.5400E+010.6750E+010.8100E+010.9450E+01
0.1080E+020.1215E+020.1350E+020.1485E+020.1620E+020.1755E+020.1890E+020.2025E+02
0.2160E+020.2295E+020.2430E+020.2565E+020.2700E+020.2835E+020.2970E+020.3105E+02
0.3240E+020.3375E+020.3510E+020.3645E+020.3780E+020.3915E+020.4050E+020.4185E+02
    2    320.4000E-01
.2806.1363.0932.0421.0431.0331.0160.0150.0160.0080.0090.0040.0040.0050.0020.0030
.0040.0010.0020.0010.0010.0040.0    .0010.0    .0    .0030.0020.0    .0010.0    .2695
0.1490E-070.4000E-010.8000E-010.1200E+000.1600E+000.2000E+000.2400E+000.2800E+00
0.3200E+000.3600E+000.4000E+000.4400E+000.4800E+000.5200E+000.5600E+000.6000E+00
0.6400E+000.6800E+000.7200E+000.7600E+000.8000E+000.8400E+000.8800E+000.9200E+00
0.9600E+000.1000E+010.1040E+010.1080E+010.1120E+010.1160E+010.1200E+010.1240E+01
    3    320.1450E+01
.1151.1091.1131.1021.0931.0761.0691.0581.0400.0380.0300.0310.0260.0230.0150.0120
.0100.0100.0070.0050.0030.0030.0030.0    .0    .0010.0030.0    .0010.0020.0    .0010
0.0        0.1450E+010.2900E+010.4350E+010.5800E+010.7250E+010.8700E+010.1015E+02
0.1160E+020.1305E+020.1450E+020.1595E+020.1740E+020.1885E+020.2030E+020.2175E+02
0.2320E+020.2465E+020.2610E+020.2755E+020.2900E+020.3045E+020.3190E+020.3335E+02
0.3480E+020.3625E+020.3770E+020.3915E+020.4060E+020.4205E+020.4350E+020.4495E+02
    4    320.3450E+00
.8557.1082.0200.0050.0010.0010.0    .0010.0020.0    .0    .0010.0    .0    .0010.0
.0010.0010.0    .0    .0    .0    .0010.0    .0    .0    .0    .0    .0    .0    .0010
0.0        0.3450E+000.6900E+000.1035E+010.1380E+010.1725E+010.2070E+010.2415E+01
0.2760E+010.3105E+010.3450E+010.3795E+010.4140E+010.4485E+010.4830E+010.5175E+01
0.5520E+010.5865E+010.6210E+010.6555E+010.6900E+010.7245E+010.7590E+010.7935E+01
0.8280E+010.8625E+010.8970E+010.9315E+010.9660E+010.1000E+020.1035E+020.1069E+02
         0
RUN
PUNCH   86   AS    5
PUNCH   87   AS    6
SAVE    86   AS    5
SAVE    87   AS    6
NEW
  COMPLEX EXAMPLE L1.
   1  1  4  1  1  1  0  1  0
     6673829          300
101
102
103      3  3
104
105
106  3  2  2     .015   .0150
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
107  3  2  2     .015   .0150
108  1
109  3  2  2      2.5    2.50
110  2                     F
  0
    3     3     3
 89  1
 90
 91
  0
  0
              90102
              91013
              93102
              94103
              96102
              97103
             101103            -1
             102103            -1
             102105             1
             102106
             103104             2
             103106
             104107
             104109
             105107
             105109
             108109
             108110  72  10
               0
              89 90   5 13
              90 91   6 13
              91 89
               0
    1102101
    2101102
  0
             0           0           -1.                    0 0
RUN
PUNCH  109    AS     1
PUNCH  107    AS     2
STOP
123456789-123456789-123456789-123456789-123456789-123456789-123456789-123456789-
```

and the corresponding arc in copy 2 (arc 62-63) has "(3, 13)". Arc 53-54 is given on line 46. Note that field 10 (columns 45-47) is "2". This indicates that copy 2 of this subsystem has different parameters for this arc. Lines 45, 50, and 51 have similar entries corresponding to arcs 54-55, 56-57, and 57-58.

Line 58 terminates reading of NEW-4A cards. At this point, GRASP has generated the first copy of the subsystem. It now begins to generate the second copy, and it generates the arcs in the same order that they were read for the first copy. Whenever it encounters a non-zero field 10, it reads a data card (type NEW-2B) that must contain the new parameters for that arc. Lines 57-60 correspond to the changed arcs in the second copy. Note that they appear in the same order as their corresponding arcs in the first copy. So, each time GRASP encounters a non-zero field 10 while generating copy 2, it will read a card which has the correct information for the arc currently being inserted in the network.

Note that field 10 in lines 57 to 60 is "3". After line 60 is read, GRASP will finish generating copy 2. Then it will begin copy 3. A similar procedure will be followed in generating this copy, only now, GRASP will reference the values for field 10 in lines 59 to 62. Thus, lines 63 to 66 are the arcs in copy 3 that have different characteristics than those in copy 1.

So, lines 59 to 66 are all NEW-4B cards. Note that GRASP generates the start and end nodes for those arcs. They have been included in lines 59 to 66 for clarity only.

Line 76 is the HIST Keyword, and it causes the following empirical distributions to be read. These are the empirical distributions that were punched as output from Levels 3(a) and 3(b) with the PUNCH Keyword. For example, line 78 contains cell probabilities for empirical distribution 1. This is the TBF distribution from Level 3(a). The first three cell probabilities are 0.1972, 0.0851, 0.0951, and 0.0841. The histogram for this distribution was shown in Figure 4.5.20. These probabilities can be confirmed by looking at the Relative Frequency column of this figure. The lower bounds of the cells given on lines 80 to 83 can similarly be checked.

Node summaries and histograms for Level 2 are given in Figures 4.5.27, 28, and 29. From Figure 4.5.27 it can be seen that Level 2 has mean time between failures of 36.15 and mean repair time of 32.92.

The SAVE Keywords at lines 109 and 110 of Figure 4.5.26 causes the histograms in Figures 4.5.28 and 29 to be converted to empirical distributions for use in Level 1. The histograms are also to be punched for later reference as shown at lines 107 and 108.

The input for level 1 starts at line 111 of Figure 4.5.26. Subsystems are used in this network, too. The first copy of the subsystem consists of nodes 89, 90, and 91 (see Figure 3.8.8). The non-subsystem nodes are 101 to 110.

In Figure 4.5.26, lines 115 to 124 are the NEW-2 cards for non-subsystem nodes. The most extensive use of default values is shown in this network. For example, line 115 is the NEW-2 card for node 101. Note that only field 1 (the node number) is non-blank. The default values for N1 and N2 are both one. Thus, a deterministic, non-statistics node with N1 = N2 = 1 can be specified by simply putting the node number in field 1.

The NEW-2A card is shown on line 126. It specifies that there are three nodes in the subsystem, three copies are desired, and there are three arcs in the subsystem. Lines 127 to 130 are the NEW-2B cards for the first copy. No parameter sets are used, so line 131 has a zero in field 1.

ORIGINAL PAGE IS
OF POOR QUALITY

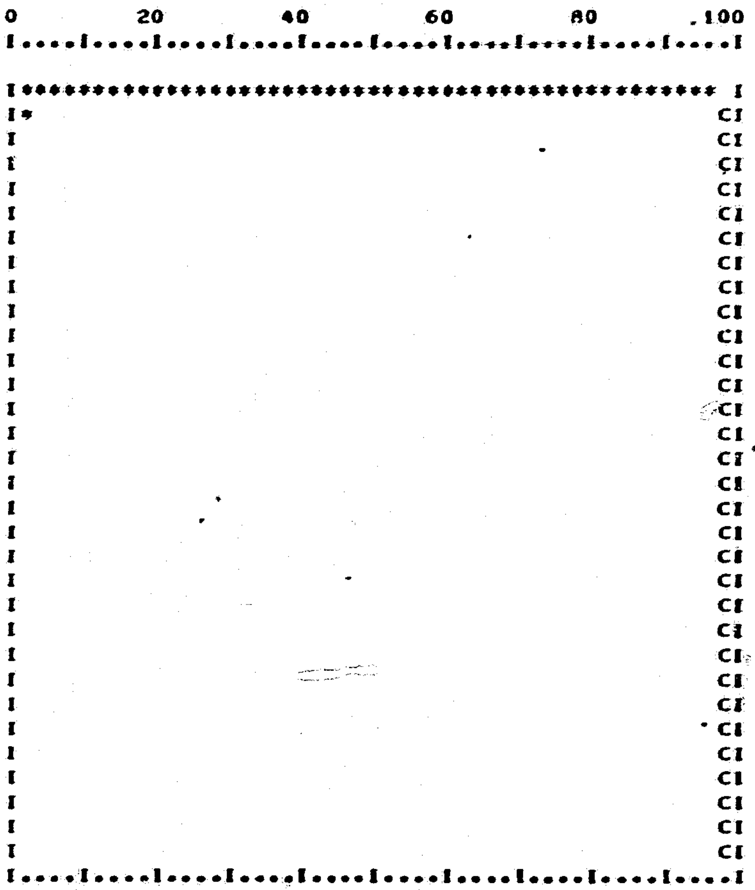
FINAL RESULTS FOR 1 SIMULATION RUN(S)

NODE	PROB./COUNT	MEAN	STD.DEV.	NO OF OBS.	MIN.	MAX.	NODE TYPE
88	1.0000	0.3450E+05	0.0	1.	0.3450E+05	0.3450E+05	F
86	1.0000	0.3615E+02	0.1707E+03	500.	0.0	0.3783E+04	D
87	1.0000	0.3292E+02	0.3653E+03	[illegible]	0.0	0.6407E+04	D

Figure 4.5.27 Node Summaries for Level 2

STAT HISTOGRAM FOR NODE 86

OBSV FREQ	RELA FREQ	CUML FREQ	LOWER BOUND OF CELL
489	0.978	0.978	-INF
10	0.020	0.998	0.1200E+03
0	0.0	0.998	0.2400E+03
0	0.0	0.998	0.3600E+03
0	0.0	0.998	0.4800E+03
0	0.0	0.998	0.6000E+03
0	0.0	0.998	0.7200E+03
0	0.0	0.998	0.8400E+03
0	0.0	0.998	0.9600E+03
0	0.0	0.998	0.1080E+04
0	0.0	0.998	0.1200E+04
0	0.0	0.998	0.1320E+04
0	0.0	0.998	0.1440E+04
0	0.0	0.998	0.1560E+04
0	0.0	0.998	0.1680E+04
0	0.0	0.998	0.1800E+04
0	0.0	0.998	0.1920E+04
0	0.0	0.998	0.2040E+04
0	0.0	0.998	0.2160E+04
0	0.0	0.998	0.2280E+04
0	0.0	0.998	0.2400E+04
0	0.0	0.998	0.2520E+04
0	0.0	0.998	0.2640E+04
0	0.0	0.998	0.2760E+04
0	0.0	0.998	0.2880E+04
0	0.0	0.998	0.3000E+04
0	0.0	0.998	0.3120E+04
0	0.0	0.998	0.3240E+04
0	0.0	0.998	0.3360E+04
0	0.0	0.998	0.3480E+04
0	0.0	0.998	0.3600E+04
1	0.002	1.000	0.3720E+04
500			



ORIGINAL PAGE IS OF POOR QUALITY

Figure 4.5.28 TBF Histogram for Level 2

```
STAT HISTOGRAM FOR NODE   87

OBSV    RELA     CUML      LOWER BOUND     0        20        40        60        80       100
FREQ    FREQ     FREQ       OF CELL        I....I....I....I....I....I....I....I....I....I....I

 495   0.992    0.992        -INF          I*************************************************I
   0   0.0      0.992      0.2050E+03      I                                                 CI
   0   0.0      0.992      0.4100E+03      I                                                 CI
   0   0.0      0.992      0.6150E+03      I                                                 CI
   0   0.0      0.992      0.8200E+03      I                                                 CI
   0   0.0      0.992      0.1025E+04      I                                                 CI
   1   0.002    0.994      0.1230E+04      I                                                 CI
   0   0.0      0.994      0.1435E+04      I                                                 CI
   0   0.0      0.994      0.1640E+04      I                                                 CI
   0   0.0      0.994      0.1845E+04      I                                                 CI
   0   0.0      0.994      0.2050E+04      I                                                 CI
   1   0.002    0.996      0.2255E+04      I                                                 CI
   0   0.0      0.996      0.2460E+04      I                                                 CI
   0   0.0      0.996      0.2665E+04      I                                                 CI
   0   0.0      0.996      0.2870E+04      I                                                 CI
   0   0.0      0.996      0.3075E+04      I                                                 CI
   0   0.0      0.996      0.3280E+04      I                                                 CI
   0   0.0      0.996      0.3485E+04      I                                                 CI
   0   0.0      0.996      0.3690E+04      I                                                 CI
   0   0.0      0.996      0.3895E+04      I                                                 CI
   0   0.0      0.996      0.4100E+04      I                                                 CI
   1   0.002    0.998      0.4305E+04      I                                                 CI
   0   0.0      0.998      0.4510E+04      I                                                 CI
   0   0.0      0.998      0.4715E+04      I                                                 CI
   0   0.0      0.998      0.4920E+04      I                                                 CI
   0   0.0      0.998      0.5125E+04      I                                                 CI
   0   0.0      0.998      0.5330E+04      I                                                 CI
   0   0.0      0.998      0.5535E+04      I                                                 CI
   0   0.0      0.998      0.5740E+04      I                                                 CI
   0   0.0      0.998      0.5945E+04      I                                                 CI
   0   0.0      0.998      0.6150E+04      I                                                 CI
   1   0.002    1.000      0.6355E+04      I                                                 CI
 ---                                       I....I....I....I....I....I....I....I....I....I....I
 499
```

Figure 4.5.29 Repair Histogram for Level 2

Lines 132 to 149 are NEW-4 cards for non-subsystem arcs. Another default has been used here. If the Distribution Type is left blank, GRASP will assume Distribution Type 10. Hence, an arc with parameters "0.0(0,10)[0,]" and no count type, activity type, or associated cost C-node, can be specified by indicating only the start and end nodes. As an example, line 132 specifies such an arc between nodes 90 and 102.

Subsystem arcs for the first copy of the subsystem are shown on lines 151 to 153. Since all of the copies of the subsystem are identical, no NEW-4B cards are needed. Note that lines 151 and 152 reference empirical distributions 5 and 6 which were SAVEd in the simulation of level 2.

Once information is stored into the arrays for empirical distributions (by HIST or SAVE Keywords) it remains there until changed by another HIST or SAVE which references the same distribution number. In other words, these arrays are not cleared by the RUN Keyword as are the statistical arrays. So, empirical distributions can be stored and will remain available until used, even if several other simulations are run between the time the distribution is stored and the time it is referenced.

Node summaries and histograms for Level 1 are shown in Figure 4.5.30 to 4.5.34. Figure 4.5.30 reveals that the mean time between failures is 34.33 (from node 109). Note that the information for nodes 106 and 107 is identical (repair times). This results from the fact that the time statistics for C-nodes are identical to those for Delay nodes. Node 107 was included to graphically depict this characteristic. Obviously, node 107 is not needed in this network. The lesson here is that when information is desired on both accumulated times and the length of the individual times, a single C-node may suffice. This situation often arises in reliability analysis as it did in this case where mean repair time and cumulative down time are both of interest.

FINAL RESULTS FOR 300 SIMULATION RUN(3)

NODE	PROB./COUNT	MEAN	STD.DEV.	NO OF OBS.	MIN.	MAX.	NODE TYPE
110	1.0000	0.7200E+03	0.7071E+01	300.	0.7200E+03	0.7200E+03	F
106	0.9533	0.1642E+03	0.1412E+03	761.	0.3435E+00	0.6556E+03	C
107	0.9533	0.1642E+03	0.1412E+03	761.	0.3435E+00	0.6556E+03	D
109	1.0000	0.3433E+02	0.2696E+02	1017.	0.1587E-01	0.1353E+03	D

Figure 4.5.30 Node Summaries for Level 1

STAT HISTOGRAM FOR NODE 109

OBSV FREQ	RELA FREQ	CUML FREQ	LOWER BOUND OF CELL	0 20 40 60 80
60	0.059	0.059	-INF	I**
51	0.050	0.109	0.2500E+01	I** C
45	0.044	0.153	0.5000E+01	I** C
66	0.065	0.218	0.7500E+01	I*** C
37	0.036	0.255	0.1000E+02	I* C
49	0.048	0.303	0.1250E+02	I** C
31	0.030	0.333	0.1500E+02	I* C
36	0.035	0.369	0.1750E+02	I* C
39	0.038	0.407	0.2000E+02	I* C
49	0.048	0.455	0.2250E+02	I** C
41	0.040	0.496	0.2500E+02	I** C
35	0.034	0.530	0.2750E+02	I* C
29	0.029	0.559	0.3000E+02	I* C
28	0.028	0.586	0.3250E+02	I* C
33	0.032	0.618	0.3500E+02	I* C
23	0.023	0.641	0.3750E+02	I* C
32	0.031	0.673	0.4000E+02	I* C
27	0.027	0.699	0.4250E+02	I* C
19	0.019	0.718	0.4500E+02	I C
20	0.020	0.737	0.4750E+02	I C
28	0.028	0.765	0.5000E+02	I* C
16	0.016	0.781	0.5250E+02	I C
23	0.023	0.803	0.5500E+02	I* C
27	0.027	0.830	0.5750E+02	I* C
18	0.018	0.848	0.6000E+02	I C
12	0.012	0.859	0.6250E+02	I C
10	0.010	0.869	0.6500E+02	I
14	0.014	0.883	0.6750E+02	I
14	0.014	0.897	0.7000E+02	I
13	0.013	0.910	0.7250E+02	I
8	0.008	0.917	0.7500E+02	I
84	0.083	1.000	0.7750E+02	I****

1017				

Figure 4.5.31 TBF Histogram for Level 1

ORIGINAL PAGE IS
OF POOR QUALITY

```
STAT HISTOGRAM FOR NODE  106

OBSV    RELA     CUML      LOWER BOUND     0        20        40        60        80       100
FREQ    FREQ     FREQ       OF CELL        I....I....I....I....I....I....I....I....I....I....I

   0    0.0      0.0          -INF         I                                                  I
   0    0.0      0.0       0.1500E-01      I                                                  I
   0    0.0      0.0       0.3000E-01      I                                                  I
   0    0.0      0.0       0.4500E-01      I                                                  I
   0    0.0      0.0       0.6000E-01      I                                                  I
   0    0.0      0.0       0.7500E-01      I                                                  I
   0    0.0      0.0       0.9000E-01      I                                                  I
   0    0.0      0.0       0.1050E+00      I                                                  I
   0    0.0      0.0       0.1200E+00      I                                                  I
   0    0.0      0.0       0.1350E+00      I                                                  I
   0    0.0      0.0       0.1500E+00      I                                                  I
   0    0.0      0.0       0.1650E+00      I                                                  I
   0    0.0      0.0       0.1800E+00      I                                                  I
   0    0.0      0.0       0.1950E+00      I                                                  I
   0    0.0      0.0       0.2100E+00      I                                                  I
   0    0.0      0.0       0.2250E+00      I                                                  I
   0    0.0      0.0       0.2400E+00      I                                                  I
   0    0.0      0.0       0.2550E+00      I                                                  I
   0    0.0      0.0       0.2700E+00      I                                                  I
   0    0.0      0.0       0.2850E+00      I                                                  I
   0    0.0      0.0       0.3000E+00      I                                                  I
   0    0.0      0.0       0.3150E+00      I                                                  I
   1    0.001    0.001     0.3300E+00      I                                                  I
   0    0.0      0.001     0.3450E+00      I                                                  I
   0    0.0      0.001     0.3600E+00      I                                                  I
   0    0.0      0.001     0.3750E+00      I                                                  I
   0    0.0      0.001     0.3900E+00      I                                                  I
   0    0.0      0.001     0.4050E+00      I                                                  I
   0    0.0      0.001     0.4200E+00      I                                                  I
   0    0.0      0.001     0.4350E+00      I                                                  I
   0    0.0      0.001     0.4500E+00      I                                                  I
 760    0.999    1.000     0.4650E+00      I**************************************************I
 ---                                       I....I....I....I....I....I....I....I....I....I....I
 761
```

Figure 4.5.32 Repair Histogram for Level 1

FINAL RESULTS FOR RUN DATA OF ACCUMULATOR NODES

FINAL RESULTS FOR 300 SIMULATION RUN(S)

TYPE	MEAN	STD.DEV.	NO OF OBS.	MIN.	MAX.
TIME	0.5990E+03	0.6039E+02	300.	0.4111E+03	0.7191E+03
COST	0.0	0.0	300.	0.0	0.0

Figure 4.5.33 Summary of Master C-node
Accumulators for Level 1

STAT HISTOGRAM FOR ACCUMULATION NODE TIMES

OBSV FREQ	RELA FREQ	CUML FREQ	LOWER BOUND OF CELL
1	0.003	0.003	-INF
0	0.0	0.003	0.4149E+03
0	0.0	0.003	0.4246E+03
1	0.003	0.007	0.4342E+03
0	0.0	0.007	0.4439E+03
2	0.007	0.013	0.4535E+03
4	0.013	0.027	0.4632E+03
3	0.010	0.037	0.4728E+03
4	0.013	0.050	0.4825E+03
4	0.013	0.063	0.4921E+03
7	0.023	0.087	0.5018E+03
9	0.030	0.117	0.5114E+03
8	0.027	0.143	0.5211E+03
10	0.033	0.177	0.5307E+03
12	0.040	0.217	0.5404E+03
18	0.060	0.277	0.5500E+03
8	0.027	0.303	0.5597E+03
12	0.040	0.343	0.5693E+03
18	0.060	0.403	0.5790E+03
14	0.047	0.450	0.5886E+03
21	0.070	0.520	0.5983E+03
24	0.080	0.600	0.6079E+03
18	0.060	0.660	0.6176E+03
19	0.063	0.723	0.6272E+03
18	0.060	0.783	0.6369E+03
13	0.043	0.827	0.6465E+03
9	0.030	0.857	0.6562E+03
10	0.033	0.890	0.6658E+03
12	0.040	0.930	0.6755E+03
6	0.020	0.950	0.6851E+03
7	0.023	0.973	0.6948E+03
8	0.027	1.000	0.7044E+03
300			

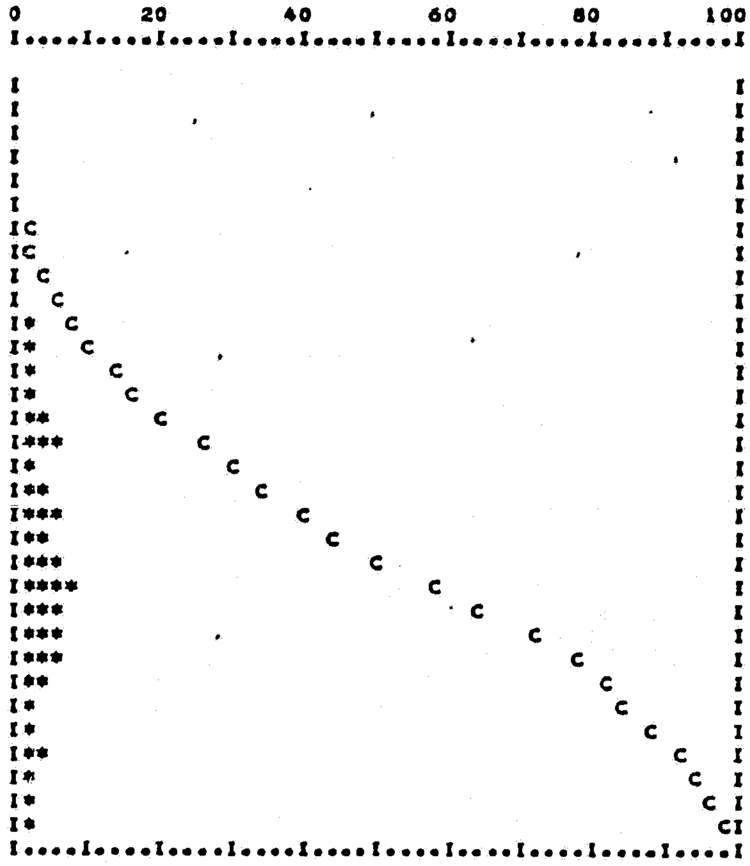


Figure 4.5.34 Master Accumulator Histogram for Level 1

The TBF histogram from node 106 is shown in Figure 4.5.26. Note that this histogram reveals that the repair time distribution is strongly uninodal.

Figure 4.5.27 is the summary for the master C-node accumulators, and it indicates that the average down time for the system during the mission time is 599, and the mission time is 1000 (from p. 98). The histogram for these times is given in Figure 4.5.28. Cost analysis was not specified in data input.

Before closing this section, the network decomposition technique requires a few additional comments. The runs and data input for this example were chosen to illustrate a variety of the capabilities of GRASP. For practical purposes, the entire decomposition example could be simulated in one computer run. The input for Levels 3(a) and 3(b) can be stacked (as were Levels 2 and 1) and the SAVE Keyword used to store empirical distributions 1 through 4. Immediately following this the Level 2 and Level 1 data cards would be inserted. The only difference is that the HIST Keyword and its associated data cards (lines 74 to 103 of Figure 4.5.20) would be removed. In this way, the entire system can be simulated in one computer run. Self-scaled histograms play an important part in this procedure since they remove the necessity of estimating histogram cell widths at the higher levels.

SECTION 5

PROGRAM STRUCTURE AND USER GUIDELINES

This section will present a summary of the structure and program organization of GRASP. The mechansim by which GRASP stores information in its filing arrays is explained and several considerations concerning running GRASP on the user's host computer are discussed.

5.1 Program Structure and Logic

The GRASP subprograms and their relationships are shown in Figure 5.1.1. Table 5.1.1 lists each subprogram with its calling parameters and a brief description of its function.

Program GRASP is the main line program. Its only function is to set certain program control variables and call GASP. GASP is the executive routine which processes events and controls the simulation. GASP calls DATIN which performs data input and decodes the control language. Return from Subroutine DATIN is prompted by the RUN Keyword.

Once control is returned to GASP, the simulation is executed. End of activity events are scheduled by Subroutine SCHAT. SCHAT uses SAMPL which calls the random variate generators, RNORM, TRNGL, GAM, BETAXF, PERTXF, DISCR, and DRAND. These routines sample from the distribution types to obtain the times for activities. The end of activity events are filed into the filing arrays with FILEM.

GASP removes events from the files with Subroutine RMOVE and collects statistics with HISTO, CSTUP, COLCT, and COLCC. When the simulation is complete, a call to SUMRY produces the output. Subroutine ADJST scales histograms and GREAD, GSAVE, and ASSGN are used to read, save and punch empirical distributions.

Figure 5.1.1 Subprogram Cross Reference

From the following diagram one may see which subprograms are used in any GRASP subprogram. For instance, the subroutine DATIN uses support subroutines:

FILEM
NFIND
BETAXF
PERTXF
GREAD
GSAVE
ERROR

Subroutine DATIN is called by subroutine GASP. In addition, the line numbers where each subprogram starts is also given. The random number generator used is the FUNCTION RANF (ISEED). This function is system dependent and is not shown in this cross reference.

	MAIN Program	DATIN	GASP	FILEM	SCHAT	RMOVE	COLCT	SUMRY	GRRAD	GSAVE	SAMPL
SUBROUTINE DATIN			●								
SUBROUTINE GASP	●										
SUBROUTINE FILEM (JD)		●	●		●						
SUBROUTINE SCHAT (NODE)			●								
SUBROUTINE RMGVE (KCOLL,JD)			●								
FUNCTION NFIND(NVAL,JQ,JATT,IRNK)		●	●								
SUBROUTINE HISTO (X1,ND)			●								
SUBROUTINE COLCT (XX,N)			●								
SUBROUTINE COLCC							●				
FUNCTION CSTUP(DUM)			●								
SUBROUTINE SUMRY			●								
SUBROUTINE SAMPL (DEV)					●						
FUNCTION RNORM(JD)											●
FUNCTION TRNGL(JD)											●
FUNCTION GAM(ALPHA,ISEED)											●
SUBROUTINE BETAXP		●									
SUBROUTINE PERTXP		●									
SUBROUTINE GREAD		●									
SUBROUTINE GSAVE (NOD,JQ,K,KWORD)		●									
FUNCTION DISCR(JP)											●
SUBROUTINE ASSGN (JD)									●	●	
SUBROUTINE ADJST								●			
SUBROUTINE ERROR (J)		●	●	●	●	●	●				

Table 5.1.1 GRASP Subprograms

SUBROUTINE DATIN	- Data Input
SUBROUTINE GASP	- Executive routine that controls execution of the simulation
SUBROUTINE FILEM (JD)	- Inserts and ranks entries in file JD
SUBROUTINE SCHAT (NODE)	- Schedules activities leaving a node
SUBROUTINE RMOVE (KCOLL, JD)	- Removes entry KCOLL from file JD
FUNCTION NFIND (NVAL, JQ, JATT, IRNK)	- Finds IRNK-th entry in file JQ whose JATT-th attribute has value NVAL
SUBROUTINE HISTO (XI, ND)	- Records observation XI in histogram ND
SUBROUTINE COLCT (XX, N)	- Records observation XX of statistic N
SUBROUTINE COLCC	- Updates C-node Time and Cost and Tests T_2 and C_2 Values
FUNCTION CSTUP (DUM)	- Calculates current accumualted costs
SUBROUTINE SUMRY	- Produces output at end of simulation
SUBROUTINE SAMPL (DEV)	- Calls for samples from distribution types and returns value as DEV
FUNCTION RNORM (JD)	- Produces normal sample from parameter set JD
FUNCTION TRNGL (JD)	- Triangular distribution
FUNCTION GAM (ALPHA, ISEED)	- Gamma variate generator
SUBROUTINE BETAXF	- Beta variate generator
SUBROUTINE PERTXF	- Generator for distribution Type 9
SUBROUTINE GREAD	- Reads empirical distributions

Table 5.5.1 (continued)

SUBROUTINE GSAVE (NOD, JQ, K, KWORD)	- Saves or punches histogram at node NOD as empirical distribution JQ. K and KWORD are for error processing
FUNCTION DISCR (JP)	- Samples from empirical distribution JP
SUBROUTINE ASSGN (JD)	- Establishes pointers for efficient sampling from empirical distribution JD
SUBROUTINE ADJST	- Scales histograms with negative cell widths
SUBROUTINE ERROR (J)	- Error processor
FUNCTION RANF (ISEED)	- Random number generator (system dependent)

5.2 GRASP Filing Structure

Several references have been made to the filing arrays of GRASP. All information on arcs and events are maintained in a single array contained in blank common. This array is QSET. It is EQUIVALENCEed to the array NSET (in the FORTRAN sense) so both integer and real number representations may be maintained in the same array.

Information in NSET/QSET is arranged into files. Each node has a file which is numbered the same as the node number. In other words, node 5 information is maintained in file 5. The contents of this file are the attributes of all arcs leaving the node. Thus, the information on Data Card NEW-4 goes into the file of the start node of the arc. Each arc in a file is an "entry", and the entries are ranked in decreasing order of their probabilities (Field 1, NEW-4) and then by order of input (if there are ties in the probabilities).

Events in GRASP simulations are the completions of the activities on the arcs. Those events are all stored in file 1. This explains why no node can be numbered one. File one is reserved for events.

Two buffer arrays, ATRIB and JTRIB, each of length six, are used to transfer information into and out of NSET/QSET. Each entry in NSET/QSET consists of 14 consecutive locations. The first 6 are the JTRIB'S, the second six are ATRIB'S, and the last two are a predecessor and successor pointer, respectively.

The contents of the predecessor location is the number of the location in NSET/QSET which contains the JTRIB (1) of the immediately preceding entry in that file (recall that the entries in a file are ranked and, therefore, have an order). The successor pointer has the same information for the immediately following entry. If an entry has no predecessor, the pointer is 99999. If an entry has no successor the pointer is 77777. If the entry is the last one which will fit in NSET/QSET (i.e., the filing array is full) then the successor pointer is 88888. Table 5.2.1 and 5.2.2 show the contents of the entries for node files and the event file.

As an example, suppose the first type NEW-2 card that is read by GRASP describes an arc that leaves Node 6. This information will become an entry in File 6 and will be stored in the first available group of 14 consecutive locations in NSET/QSET. A record is kept of the address where file 6 starts. Since we are considering the first NEW-4 card, this will be the first 14 locations of cells in NSET/QSET. Hence, in file 6 NSET(1) will contain the arc's end node (JTRIB(1) -- see Table 5.2.1) and QSET(12) will correspond to ATRIB(6) (not used for node files). Since this is the first entry in File 6, it has no predecessor and NSET(13) will be equal to 99999. Since no successor exists, NSET(14) = 77777.

Now, if the next arc read also leaves Node 6, its information will also go into file 6, but it will go into NSET/QSET locations 15 through 28. The successor pointer for the first entry (NSET(14)) will be changed to point to the address of the first cell in the second entry. In other words, NSET(14) will equal 15. The predecessor pointer for the second entry (NSET(27)) will equal one (since the first cell of the first entry is NSET(1) and the successor pointer will equal 99999. In this way, a chain of entries in the filing array is built which permits locating any entry in the file. For simplicity in this example, we have assumed that the two arcs have the same probability. If the probabilities differed, the first entry would be the one with the higher probability.

This elementary description of the filing structure is necessary to interpret the file dump produced by Subroutine ERROR. A complete description and an example of this dump is given in Section 5.3.

Also, since the filing array is in blank common, the program can be lengthened or shortened by changing the dimension of QSET in the main program (GRASP) only. QSET is dimensioned to length one in all other subprograms. Hence, if core storage is a prime consideration, and if the systems to be simulated don't have very many arcs, the program can be easily shortened by changing the length of QSET. Array adjustment procedures are described in Section 5.4.

Table 5.2.1 Contents of Entries in Node Files

Location Within Entries	Contents	Buffer Array Equivalence
1	the arc's end node	JTRIB (1)
2	parameter set	JTRIB (2)
3	distribution type	JTRIB (3)
4	count type	JTRIB (4)
5	activity number	JTRIB (5)
6	associated C-node	JTRIB (6)
7	probability of arc	ATRIB (1)
8	not used	ATRIB (2)
9	fixed cost	ATRIB (3)
10	variable cost	ATRIB (4)
11	not used	ATRIB (5)
12	not used	ATRIB (6)
13	predecessor pointer	--
14	successor pointer	--

Table 5.2.2 Contents of Entries in File 1 (Event File)

Location Within Entries	Contents	Buffer Array Equivalence
1	the arc's end node	JTRIB (1)
2	parameter set	JTRIB (2)
3	distribution type	JTRIB (3)
4	count type	JTRIB (4)
5	activity number	JTRIB (5)
6	associated C-node	JTRIB (6)
7	activity completion time	ATRIB (1)
8	time mark node activated	ATRIB (2)
9	fixed cost	ATRIB (3)
10	variable cost	ATRIB (4)
11	network cost at time mark node activated	ATRIB (5)
12	start time of activity (if JTRIB (6) > 0)	ATRIB (6)
13	predecessor pointer	--
14	successor pointer	--

5.3 File Dumps

GRASP files are dumped whenever a call is made to subroutine ERROR by another subprogram which has detected an error condition. In addition, a file dump can be requested as a debugging aid by the user by specifying an appropriate value for field 19 on date card type NEW-1.

Part of a file dump for the example in Figure 3.4.2 is shown in Figure 5.3.1. This dump was requested on data card type NEW-1. The event file is empty for all such dumps, so no entries in File 1 will be listed. This will always be true for user requested dumps because the dump is produced before or after the simulation is performed. If a dump is produced as a result of an error condition, the event file will, in general, not be empty, and File 1 entries will be printed.

Now, referring to Figure 5.3.1, we see that this dump was produced at time 500. In other words, this dump was produced after the network was simulated. The "P=" and "S=" notation indicates the predecessor and successor pointers in NSET/QSET.

Files 2 through 8 are shown in Figure 5.3.1. Recall that these are node files, so the entries in these files will correspond to the arcs leaving nodes 2,...., 8 in Figure 3.4.2. Of course, there will also be a file for every other node in Figure 3.4.2 that has a leaving arc.

Each entry in a file is represented by three lines. The first line starts with "CELL=". The value to the right of this is the cell number of NSET/QSET which contains the first value (JTRIB(1)) of this entry. In other words, this entry in File 2 is physically stored in NSET(1) through NSET(14). This is not the only entry in File 2. It has no predecessor,

ORIGINAL PAGE IS
OF POOR QUALITY

```
FILE STATUS AT TIME   0.5000E+03
    P=PREDECESSOR POINTER   S=SUCCESSOR POINTER

FILE     2

CELL=      1     P= 99999    S=     15
JTRIB                    3          1           5           0           0           0
ATRIB              0.1000E+01  0.0        0.0         0.0         0.0         0.0

CELL=     15     P=     1    S=     29
JTRIB                    6          0          10           0          -1           0
ATRIB              0.1000E+01  0.0        0.0         0.0         0.0         0.0

CELL=     29     P=    15    S= 77777
JTRIB                    7          0          10           0          -1           0
ATRIB              0.1000E+01  0.0        0.0         0.0         0.0         0.0

FILE     3

CELL=     43     P= 99999    S=     57
JTRIB                    2          2           4           0          -1           0
ATRIB              0.1000E+01  0.0        0.0         0.0         0.0         0.0

CELL=     57     P=    43    S=     71
JTRIB                    6          0          10           0           0           0
ATRIB              0.1000E+01  0.0        0.0         0.0         0.0         0.0

CELL=     71     P=    57    S= 77777
JTRIB                    7          0          10           0           0           0
ATRIB              0.1000E+01  0.0        0.0         0.0         0.0         0.0

FILE     4

CELL=     85     P= 99999    S=     99
JTRIB                    5          1           5           0           0           0
ATRIB              0.1000E+01  0.0        0.0         0.0         0.0         0.0

CELL=     99     P=    85    S=    113
JTRIB                    6          0          10           0          -1           0
ATRIB              0.1000E+01  0.0        0.0         0.0         0.0         0.0

CELL=    113     P=    99    S= 77777
JTRIB                    7          0          10           0          -1           0
ATRIB              0.1000E+01  0.0        0.0         0.0         0.0         0.0

FILE     5

CELL=    127     P= 99999    S=    141
JTRIB                    4          2           4           0          -1           0
ATRIB              0.1000E+01  0.0        0.0         0.0         0.0         0.0

CELL=    141     P=   127    S=    155
JTRIB                    6          0          10           0           0           0
ATRIB              0.1000E+01  0.0        0.0         0.0         0.0         0.0

CELL=    155     P=   141    S= 77777
JTRIB                    7          0          10           0           0           0
ATRIB              0.1000E+01  0.0        0.0         0.0         0.0         0.0

FILE     6

CELL=    169     P= 99999    S= 77777
JTRIB                    7          0          10           0           0           0
ATRIB              0.1000E+01  0.0        0.0         0.0         0.0         0.0

FILE     7

CELL=    183     P= 99999    S= 77777
JTRIB                    6          0          10           0          -1           0
ATRIB              0.1000E+01  0.0        0.0         0.0         0.0         0.0

FILE     8

CELL=    197     P= 99999    S= 77777
JTRIB                    9          3           1           0           0           0
ATRIB              0.1000E+01  0.0        0.0         0.0         0.0         0.0
```

Figure 5.3.1 Dump of Node Files for Figure 3.4.2

P = 99999, but it has a successor, S = 15. Since we have 3 arcs leaving node 2, we have three entries in File 2. Each entry represents an arc. The meaning of JTRIB and ATRIB values (refer to Tables 5.2.1 and 5.2.2) agree with the information supplied in Figure 3.4.2. The second entry corresponds to the arc (2,6) and it has P = 1 and S = 29. This indicates that the previous entry starts at NSET(1) and the next entry starts at NSET(29). The last entry in this file corresponds to arc (2,7), S = 77777 indicates that this is the last entry. Files 3, 4 and 5 also have 3 entries each. Files 6, 7, and 8 have only one entry each. P = 99999 and S = 77777 for each of them.

The way in which a chain is built by the predecessor and successor pointers make the storage allocation very efficient. The program always keeps track on the first available space in NSET/QSET, the address of the first and last entry in each file by using some other pointers. This mechanism is the same as in GASP IV (reference 19).

The statistical and histogram storage arrays are also dumped whenever the dump is produced from an error condition or the user requests it <u>after</u> execution.

The array SUMA contains the information from which the node summaries are produced. If X represents the observations then each row of SUMA will contain the following:

SUMA(·,1)	=	sum of X's
SUMA(·,2)	=	sum of squares of X's
SUMA(·,3)	=	number of observations
SUMA(·,4)	=	minimum of the observations
SUMA(·,5)	=	maximum of the observations

ARRAY SUMA

9 (0)	0.5000E+03	0.2500E+06	0.1000E+01	0.5000E+03	0.5000E+03
7 (0)	0.4527E+03	0.1233E+05	0.2000E+02	0.6122E+01	0.4504E+02
6 (0)	0.4924E+03	0.1414E+05	0.2100E+02	0.7072E+01	0.4741E+02
9COST	0.0	0.0	0.1000E+01	0.0	0.0
7COST	0.0	0.0	0.2000E+02	0.0	0.0
6COST	0.0	0.0	0.2100E+02	0.0	0.0

Figure 5.3.2 Dump of Array SUMA
for Figure 3.4.2

9TIME	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0
7TIME	1 1 0	0 1 0	1 1 0	0 1 2	0 1 0	1 0 0	1 0 0	2 1 0	1 1 0	0 0 1	1 2
6TIME	1 2 0	0 0 1	1 1 0	2 0 1	0 1 0	0 1 0	2 0 1	1 1 0	1 1 0	2 0 1	0 0
9COST	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0
7COST	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0
6COST	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0 0	0 0

Figure 5.3.3 Dump of Array JCELS for Figure 3.4.2

Figure 5.3.2 is a dump of SUMA after execution for Figure 3.4.2. The rows of SUMA are printed in order. Each statistics node requires at least two rows in SUMA. One collects time statistics and the other collects cost statistics. In addition, if count types exist, each statistics node will require a row for each count type. Further, if C-nodes exist in the network, the last two rows in SUMA will contain the information for the master time and cost accumulators, in that order. Each row is labeled with the number of the associated node. Rows for time statistics are listed first. Count type information immediately follows the time information. The notation "6(1)" would indicate that that row is for node 6, count type 1. The cost rows follow these, and finally the time and cost accumulators are listed.

Included in the dump of statistical arrays is a printing of the histogram storage array JCELS. The 32 cells of each histogram are printed and labeled with node numbers. Time histograms are printed first and then cost histograms. As before, the last two rows in JCELS contain the histograms for the master time and cost accumulators. See Figure 5.3.3 for the dump of JCELS for Figure 3.4.2.

5.4 Compatibility and Portability

With current array dimensions, GRASP requires about 33000_{10} words of core memory. In addition, if self-scaled histograms are utilized, a disk storage device is required.

Specific user actions are as follows:

1. The variables in the main program for card reader (NCRDR), printer (NPRNT), punch (NPNCH) and disk file (NHIS) equipment numbers must be set properly. Further, if the dimension of QSET is changed, the variable MAXQS must be set equal to the

dimension of QSET, IMN = maximum node number allowed, and MAX = the maximum number of empirical distributions permitted

2. Function RANF is the function that generates uniform pseudo-random numbers on the interval (0,1). Its argument ISEED is defined by the user and automatically changes every time this function is called. It is recommended that the random number generator contained on the user's computer system be used.

3. GRASP was originally developed on a CDC 6400 system (reference 21). This AMDAHL version includes some new features, such as the node release mechanism, and does not retain some of the earlier characteristics. For instance, the initial value of the seed for the random number generator is no longer available during the simulation because it is not needed for most purposes. Other omissions are the use of the random number generator as a specific subprogram, and the subroutine previously used for time limit checking. A time limit is always available through the job card when we run the program.

5.5 Adjusting the Array Sizes of GRASP

The maximum size network that GRASP can accommodate is determined by the dimensions of certain arrays that are contained in the labeled COMMON blocks GL through G7. These COMMON statements are shown in Figure 5.4.1. This section will list the present size limitations for GRASP and give instruction on how to alter those dimensions if desired.

```
      COMMON QSET(5600)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      COMMON /G3/ KST1(100),XLOW(200),NREL1(300),NREL2(300),NREL(300),
     1  MREL(300),KST4(100),KST3(100),NTYPE(300),WIDTH(200),NSTS,NCTS
      COMMON /G4/ XSTUS(100),SUMCT,CSTUS(100),NCND,NCNL,NCNU,T2,C2,
     1  ITFLG,ICFLG,TYYY,TCCC,TIMTD,COSTD
      COMMON /G5/ MSTN(2000,2),MST(300),KST2(300),IGRF,JGRAF,SCAL
      COMMON /G6/ NCEL(10),PROB(32,10),VAL(33,10),ISEC(3,10),MAX,NHIS,
     1  KHIST,NPNCH
      COMMON /G7/ ID,IMN,IDMP,NDTU,NSKD,NSTR,NETR,TSTR,TETR,NSRC,NDRT,
     1  NSORC(100),NPO(100),NABA(200)
```

Figure 5.4.1 GRASP Common Statements

Immediately following this text are ten numbered sections. Each section describes an attribute of a GRASP network that is limited by array sizes. There are three parts to each section. The first is a variable or expression that can be computed from the GRASP network that the user intends to simulate. When this value is obtained, it must be compared to the "current maximum" value which is the second part of each section. If the actual value is not greater than this maximum, then the array sizes are sufficient to process the network.

If some attribute of the network exceeds one or more fo the current maximum values, then the "Arrays Affected" must be redimensioned. The affected arrays are the third part of each numbered section. The minimum sizes for these arrays are given in terms of the variable or expression that was computed for the users network. Note, however, that when the COMMON blocks are changed, numerical values must be inserted in place of the variables shown in the "Arrays Affected".

For example, consider Section 1: Largest Node Number. The variable name is IMN. Suppose the largest node number in our network is 115. The current maximum is 300. Since 115 is less than 300, no arrays need to be changed. On the other hand, if the largest node number is 341, the dimensions of all of the affected arrays must be changed to at least 342.

The variable name or expression given in each section is intended to help the user in computing the values for his network and in figuring the minimum dimensions required if some arrays must have their dimensions changed. With three exceptions, the user is not required to set values for these variables in GRASP. The three exceptions are IMN, the variable for the maximum node number, MAX, the variable for the maximum number of empirical distributions which can be stored by GRASP at one time, and MAXQS, the variable for the length of QSET. Values for these variables must be set by the user in the main program whenever the arrays in Sections 1, 9, or 10 are changed.

In other words, when changing array dimensions for those arrays mentioned in Sections 2 through 8, the only FORTRAN statements that the user must change are the COMMON statements. When changing the arrays in Sections 1, 9, or 10, the FORTRAN statements that set values for IMN, MAX, and MAXQS must also be changed.

1. Largest Node Number

Variable: IMN

Current Maximum = 300

We can have at most 299 nodes in the network since we cannot have node #1.

Arrays Affected: MFE(IMN), MLE(IMN), NQ(IMN), NREL1(IMN), NREL(IMN, NTYPE(IMN), NFTBU(IMN), NREL2(IMN), MFEN(IMN, LSINK(IMN), MREL(IMN)
(The variable IMN must be set in the main program whenever these arrays are redimensioned).

2. Largest Number of Statistics Nodes

Variable: NSKS

Current Maximum: 100

Arrays Affected: JCELS(2*NSKS,32), NSINK(NSKS), XLOW(2*NSKS), WIDTH(2*NSKS), NSNR(NSKS), JSINK(NSKS), XSTUS(NSKS), CSTUS(NSKS)

Note: The first dimension of JCELS must be greater than or equal to 44, since the array is also used during subsystem input.

3. Maximum Number of Count Types

Variable: $NCTS \leq [\frac{NXX - NX}{NSKS}] - 2$

Current Maximum: 2*NCTS + (NCTS*NSKS) + 2 $\leq$ 250

- where NXX is the first dimension of SUMA and NX = 0 if no C-nodes, NX = 2 if C-nodes in the network. [ ] menas truncated to the next lower integer

Arrays Affected: KOUNT(NCTS)
SUMA(2*NCTS + (NCTS * NSKS) + 2, 5)

4. Maximum Number of Count Types (NCTS) Plus Statistics Nodes

Expression: 2*NSKS + (NCTS * NSKS) +2

Current Maximum: 123 with C-nodes
125 without C-nodes

Arrays Affected: SUMA(2 * NSKS + (NCTS * NSKS) + 2, 5)

5. Maximum Parameter Set Number

Variable: NPRMS

Current Maximum: 100

Arrays Affected: PARAM(NPRMS,4)

6. Maximum Number of Source Nodes

Variable: NCRC

Current Maximum: 100

Arrays Affected: NSORC(NSRC)

7. Largest Positive Activity Number

Variable: MMACT

Current Maximum: 100

Arrays Affected: NPO(MMACT)

8. <u>Maximum Number of Activities Triggering Modifications (MACT) Plus the Number of Modifications (MODS)</u>

   Expression: MACT + 2 * MODS

   Current Maximum: 200

   Arrays Affected: NABA(MACT + 2 * MODS)

9. <u>Maximum Number of Arcs in a Network that Can Occur in a Simulation Run</u>

   Variable: ID

   Current Maximum: 400

   Arrays Affected: NSETN(ID,2), QSET(14 * ID)
   (when QSET is redimensioned,
   the variable MAXQS msut be
   set to the length of QSET
   in the main program)

10. <u>Maximum Number of Empirical Distributions</u>

    Variable: MAX

    Current Maximum: 10

    Arrays Affected: PROB(32,MAX), VAL(33,MAX),
    ISEC(3,MAX), NCEL(MAX)
    (The variable MAX must be set in
    the main program whenever these
    arrays are redimensioned)

GENERAL REFERENCES

1. Barlow, Richard E., and Proschan, Frank, Mathematical Theory of Reliability, John Wiley & Sons, Inc., 1975.

2. Bazovsky, Igor, Reliability Theory and Practice, Prentice-Hall, Inc., 1961.

3. Case, Kenneth E., and Morrison, Kenneth R., "A Simulation of System Reliability Using GERT III", Paper presented to Virginia Academy of Science meeting May 14, 1971.

4. Drenick, R. F., "The Failure Law of Complex Equipment", J. Society of Industrial and Applied Mathematics, Vol. 8, No. 4, December 1960, pp. 680-690.

5. Fishman, G.S., Concepts and Methods in Discrete Event Digital Simulation, John Wiley & Sons, Inc., 1973.

6. Gaver, D.P., "Failure Time for a Redundant Repairable System of Two Dissimilar Elements", IEEE Transactions on Reliability, Vol. R-13, No. 1, March 1964, pp. 14-22.

7. Gaver, D.P., "Time to Failure and Availability of Paralleled Systems with Repair", IEEE Transactions on Reliability, Vol. R-12, No. 2, June 1964, pp. 30-38.

8. Jewell, W.S., "Comments on 'Reliability Considerations for a Two Element Redundant System with Generalized Repair Times'", Operations Research, Vol. 15, No. 1, 1967, pp. 157-159.

9. Liebowitz, Burt H., "Reliability Considerations for a Two Element Redundant System with Generalized Repair Times". Operations Research, Vol. 14, 1966, pp. 233-241.

10. Luetjen, Palmer, "Maxisim/Tiger User's Manual", Ship System Design and Development Division Naval Ship Engineering Center, August 1972.

11. Ostle, Bernard, Statistics in Research, The Iowa State University Press, Fifth printing, 1972.

12. Pritsker, A. Alan B., and Sigal, C. Elliot, The GERT-IIIZ User's Manual, Pritsker and Associates, Inc., 1974.

13. Pritsker, A. Alan B., and Kiviat, Philip J., Simulation with GASP II, A FORTRAN Based Simulation Language, Prentice-Hall, Inc., 1969.

14. Pritsker, A. Alan B., and Happ, W. William, "GERT: Graphical Evaluation and Review Technique, Part 1 - Fundamentals", The J. of Industrial Engineering, Vol. XVII, No. 5, May 1966, pp. 267-274.

15. Pritsker, A. Alan B., and Whitehouse, Gary E., "GERT: Graphical Evaluation and Review Technique, Part 2 - Probabilistic and Industrial Engineering Applications", The J. of Industrial Engineering, Vol. XVII, No. 5, June 1966, pp. 293-301.

16. Pritsker, A. Alan B., and Burgess, Roland R., "The GERT Simulation Program: GERTS III, GERTS IIIQ, GERTS IIIC, and GERTS IIIR", NASA/ERC Contract NAS-12-2113, VPI, 1970.

17. Pritsker, A. Alan B., "New GERT Concepts and a GERT Network Simulation Program", Purdue University, School of Industrial Engineering, Research Memorandum Series No. 7108, May 1971.

18. Pritsker, A. Alan B., and Phillips, Don T., "GERT Network Analysis of Complex Queueing Systems", Purdue University, School of Industrial Engineering, Research Memorandum Series No. 72-1, January 1972.

19. Pritsker, A. Alan B., The GASP IV Simulation Language, John Wiley & Sons, Inc., 1974.

20. Phillips, Don T., "Applied Goodness of Fit Testing", OR Monograph Series No. 1, The American Institute of Industrial Engineers, Inc., 1972.

21. Shulaker, Edward R., and Phillips, Don T., "GRASP-I -- GERT Reliability Simulation Program", Chapter 5 of Vol. 2 of "System Engineering of Hierarchy Computer Systems for Large Steel Manufacturing Complexes", Published under NSF Grant ATA-73-07822 A01, Report No. 63, PLAIC, Purdue University, 1974.

22. Shulaker, Edward R., GRASP-I, GERTS Reliability Analysis Simulation Program, Unpublished Master's thesis, Purdue University, West Lafayette, Indiana, 1974.

23. Whitehouse, Gary E., and Pritsker, A. Alan B., GERT: Part III -- Further Statistical Results; Counters, Renewal Times, and Correlations", AIIE Transactions, Vol. I, No. 1, March 1969, pp. 45-50.

24. Whitehouse, Gary E., Systems Analysis and Design Using Network Techniques, Prentice-Hall, Inc., 1973.

APPENDIX A

A GLOSSARY OF IMPORTANT GRASP VARIABLES

Variable	Use
ATRIB(6), JTRIB(6)	Buffer array for data transfer.
COSRN	Cost in C-node for this run.
COSTD	Old cost-to-date in C-nodes.
CSTUS(100)	Storage area used in collecting cost statistics.
C2	Termination criterian in a C-node (NEW - 6 card)
DD	Character variable equal to 'D' characterizing the output of a node.
DUM(1408)	Working array equivalent to JCELS(.,.)
EPS	Threshold value within which 2 or more events are considered to have occurred simultaneously
ICFLG	Input variable indicating whether the C2-criterian will terminate the simulation run or not.
ICHNG	When equal to 1, indicate that the EDIT mode is on.
ICRD	Number of the input file (usually = 5 for card reader)
ID	Maximum number of entries that can fit in NSET/QSET.
IDMP	Option for dumping the content of the files.
IGRF	Option for the type of histograms to be printed.
IMN	Maximum node number allowed by array sizes (300).
IM1, IM2	Length of array ATRIB, JTRIB (=6).
INOUT	Type of node, 1 for source, 2 for sink, 3 for stat., 4 for a mark node.
IPARS	Parameter set number
IPASS	Indicates when equal to 0, that the number of arcs in a particular subsystem is greater than 99.

Variable	Use
IPRT	Device number for the output (usually = 6 for line printer).
IP1	Points to the location of the 1st available group of 3 cells in the array NRP (.)
IP2	Dummy pointer in node replacement section.
IRNK	Rank of an area being edited.
ISEC(3, 10)	storage area used in subroutine ASSGN and DISCR.
ISED	Seed for random number generation.
ISIM	Number of simultaneous events at time TNOW.
ISIMX	Maximum value of ISIM at any time (=50).
ITFLG	Input variable indicating whether the T2 criterian will terminate the simulation run or not.
ITED	Variable set equal to 1 if the number of subsystems exceeds 10.
JCELS(.,32)	Storage array for histograms.
JGRAF	Takes the value 1 if ATRIB(3) or ATRIB(4) > 0.
JJJ(50)	Working array for simultaneous events.
KACTS	Number of arcs in a particular subsystem.
KEYS((7)	Storage array for the key words NEW, RUN, STOP, HIST, SAVE, EDIT and PUNC.
KHIS	Number assigned to an empirical distribution.
KHIST	Number of histograms to be collected.
KOUNT(I)	Current value of count type I.
KST1(K) = NODE	Array containing the statistics node numbers.
KST2(NODE) = K	K is the k-th statistics node. KST1 and KST2 point to one another.

Variable	Use
KST3(K)	Statistics code (F,A,B,I,D,C) maintained in the k-th statistics node.
KST4(K)	Counts the number of times a statistic of type KST3(K) has been collected in the k-th statistics node
KSSP	Copy number of a particular subsystem.
KSUBS	Subsystem type.
KTYP(S)	Storage for the keywords PARM, ARCS, NODE, CNOD and ONE, used during the EDIT mode.
MACT	Maximum number of arcs triggering modifications.
MAX	Maximum number of empirical distributions.
MAXQS	Maximum length of NSET/QSET.
MDATA(6)	Storage for the characters F,A,B,S,D and C indicating the type of statistics to be collected.
MFA	Location in NSET/QSET of 1st available space for storing an entry in a file
MFE(I)	Location in NSET/QSET of 1st entry of file I.
MLE(I)	Location in NSET/QSET of last entry of file I.
MMACT	Largest number an arc can have (=100=/dimension of NPO)
MODI	Input variable equal to 1 if modifications exist in the network.
MODS	Number of modifications in the network.
MREL(I)	Current value of the release counter of node I.
MREL1	= N1 an input for the node being read in.
MREL2	= N2 an input for the node being read in.
MSINK	Type of statistics to be collected: F,A,B,C,D or I.

Variable	Use
MST(NODE) = I MSTN(I,1) = cell in NSET/QSET MSTN(I,2) = I2	MST starts a chain of arcs that arrive to a node. MSTN(I,1) is the location in NSET/QSET of an arc arriving at node NODE. MSTN(I2,1) is the next such arc, etc.
MXC	2nd argument in the size of array JCELS(.,.) (=32)
NABA (100)	Storage array for network modification.
NABYA (24)	Buffer for storing the content of arc data card in node replacement section.
NACTN	An activity number that triggers the replacement of a node to occur. If negative, it represents a continuation of previous card information.
NACTS	Number of activities in a particular subsystem.
NAME(20)	Array representing the general information about the network
NCEL(I)	Number of cells in empirical distribution #I.
NCELS	Number of cells in current empirical distribution.
NCLT	Number of quantities on which statistics are to be collected.
NCND	Number of C-nodes.
NCRC	Maximum number of source nodes allowed.
NCTS	Number of count types.
NC2R	Current value of C2.
NDIN1	Indicates the 1st node in, in node replacement section.
NDIN2	Indicates the next node in, in node replacement section.
NDOUT	Indicates the node out in node replacement section.

Variable	Use
NDR(N)	If node N is to be replaced by another node, then NDR(N) points to the location in NRP(.) where the 1st replacing node is located.
NDT	Number of delay type nodes.
NEND(50)	Working array representing end nodes of arc terminating at the same time
NETR	Run number for end of tracing.
NHIS	Number of the output file for storing an empirical distribution in a histogram form.
NHIST	Number of histograms collected.
NNSUB	Number of nodes in particular subsystem.
NOQ	Largest node number read in.
NPNCH	Number of the output file for intermediate results to be used in a later simulation.
NPO(NACTN)	For activity type NACTN, it indicates the starting location in NABA where modification information is stored, such that: NABA(NPOS) = NABYA(I) = node out NABA(NPOS+1) = NABYA(I+1) = node in Entries in NABA are separated by zeros.
NPOS	Total number of nodes affected by replacement. It represents the number of items in NABA and is an even number.
NPRMS	Largest parameter set number.
NPRT	Output option if equal to 1 then the input data cards will not be printed out.
NQ(S)	Current number of entries in file I.
NREL(I)	Initial value of the release counter of node I. It is either NREL1(I) or NREL2(I).

Variable	Use
NRNS	Input value specifying the total number of runs for the simulation.
NRP(.)	Contains a replacing node in each group of 3 cells.
NRUN	Number of the current simulation run
NSBS	Input value specifying the number of subsystems in the network.
NSET(I)	Integer representation of filing array. Used for integer attribute values and pointers.
NSINK(K)	Node number of the K-th statistics node.
NSKD	Total number of sink and delay nodes.
NSKND	Number of sink and not delay nodes.
NSKS	Maximum number of statistics nodes allowed.
NSKT	Total number of sink node releases to end the simulation run; (the sink nodes must be different).
NSNK	Total number of sink nodes.
NSORC(100)	Storage array for source nodes.
NSRC	Total number of source nodes.
NSSP	Number of times a particular subsystem occurs.
NSTC	Total number of accumulator nodes.
NSTD	Total number of stat. and delay nodes.
NSTND	Total number of stat. and not either a delay or accum. nodes.
NST01(99) - NST08(99)	Temporary arrays for storing node information needed for subsystem generation.
NSTR	Run number for beginning of tracing.

Variable	Use
NSTS	Number of statistics nodes.
NSTST	Number of statistics, sink and accumulated type nodes.
NSUBI(10,4)	Storage area for input subsystems.
NT = NTYPE(NODE)	
NUACT	Used for subsystem arcs in NEW-4A input card.
NUNOD	Used for subsystem nodes in NEW-2B input card.
NT2C2	Number of times system exceeded both T2 and C2.
NT2R	Current value of T2
PARM(.,4)	Storage area for parameter sets.
PP	Character variable equal to 'P' specifying that the output of a node is stochastic.
PRAM(4)	Parameter values as described in Table 4.3.2.
PRMV	Character variable characterizing the input side of a node. Can take values equal to HH = 'H', AA = 'A', or UU = 'U'?
PROB(32,.)	Probability storage for empirical distributions.
PTOPT	Output characteristics of a node equal to PP = 'P' for stochastic nodes or DD = 'D' or blank for deterministic nodes.
QSET(.)	Real valued representation of file storage areas.
SCAL	Scale factor for distribution type 10.
ST01(99) - ST06(99)	Temporary arrays for storing arc information needed for subsystem generation.
SUMA(.,5)	Statistical storage for node summaries.
SUMCT	Total cost for an activity.
TCCC	Cummulative cost of still active C-nodes at end of runs.

Variable	Use
TIMTD	Old time-to-date in c-nodes.
TIM1	If TIM1 = 0, then start tracing when TNOW $\geq$ TIM1.
TIM2	Stop tracing when TNOW > TIM2.
TIMRN	Time in C-node for this run.
TNOW	Current simulation time.
TYYY	Cumulative times of still active C-nodes at end of runs.
T2	Maximum value for C-node time accumulator statistics.
VAL(33,.)	Cell values for empirical distributions.
WIDC	Same as WIDH but for cost histograms.
WIDH	Width of each cell of a histogram.
WIDTH(200)	Storage area for width of cells at nodes where histograms are to be collected.
WTC1	Same as WTT1 but for costs.
WTT1	Width of cell for master C-node time histogram.
XCOST	Accumulated cost in C-node to date.
XLC	Same as XLL but for cost histograms.
XLC1	Same as XLT1 but for costs.
XLL	Lower limit of the second cell of a histogram.
XLOW(.)	Storage array for the lower limit of the second cells of the histograms.
XLT1	Lower limit of second cell of master C-node time histogram.
SXTUS(100)	Work storage area used in collecting time statistics.
XTIME	Accumulated time in C-node to date.

APPENDIX B

GRASP ERROR TYPES

Error Number	Subroutine	Probable Cause
10	GASP	The program is very likely going through an infinite loop. Check the network model around the node indicated
11	RMOVE	Attempt to remove an entry from an empty event file. Since entries that are removed from the event file are scratched, it is likely that the program stopped because the flow of transactions ends at a node that is not a C-node or a sink node
13	DATIN	Attempted to Edit in arc that does not exist
14	GSAVE	Attempted to save a histogram with no observations recorded
22	GASP	Number of statistics nodes to realize the network is < 0. Probable cause is incorrect dimension of QSET
23	SCHAT	Pointing error in filing array. Probable incorrect dimension of QSET of value of MAXQS
87	FILEM	Filing array field. Increase dimension of QSET and check value of MAXQS
88	FILEM	Filing pointer error. Increase dimension of QSET and check value of MAXQS
89	NFIND	Entry in file to be located; has negative location. Probable incorrect dimension of QSET
90	COLCT	Probable cause is incorrect data input of number of statistics nodes or count types

Error Number	Subroutine	Probable Cause
95	DATIN	Increase dimension of array NABA
97	RMOVE	Attempted to remove an entry from a file that does not exist. Probable incorrect dimensions of QSET or attempt to release a node that has no arc originating from it.
98	GASP	There are too many simultaneous arc completions. Check the model or increase the dimension of NEND, JJJ and change the value of ISMIX in the subroutine GASP
99	DISCR	Requested sample from empirical distribution number greater than MAX. Check data input and value of MAX

Any other error whose number does not appear above is very likely to be an input error.

APPENDIX C

GRASP LISTING

```
C
C     GRASP VERSION DATED JULY 1ST, 1981. THE CAPABILITY TO PROCESS
C     SIMULTANEOUS EVENTS, MULTIPLE AND COMPOUND NODE REPLACEMENTS
C     WAS ADDED TO THE GRASP VERSION DATED JUNE 1ST, 1979. THE 1979
C     VERSION CONTAINS EXTENSIONS TO THE NODE RELEASE MECHANISM,
C     AND IS BASED ON THE MARCH 15, 1976 VERSION. THE FIRST VERSION
C     OF GRASP WAS BASED ON SUBSYSTEM AND C-NODE ADAPTATION OF
C     GERTS-IIIZ, VERSION DATED JUNE 20, 1973.
C
      COMMON QSET(5600)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      COMMON /G3/ KST1(100),XLOW(200),NREL1(300),NREL2(300),NREL(300),
     1  MREL(300),KST4(100),KST3(100),NTYPE(300),WIDTH(200),NSTS,NCTS
      COMMON /G4/ XSTUS(100),SUMCT,CSTUS(100),NCND,NCNL,NCNU,T2,C2,
     1  ITFLG,ICFLG,TYYY,TCCC,TIMTD,COSTD
      COMMON /G5/ MSTN(2000,2),MST(300),KST2(300),IGRF,JGRAF,SCAL
      COMMON /G6/ NCEL(10),PROB(32,10),VAL(33,10),ISEC(3,10),MAX,NHIS,
     1  KHIST,NPNCH
      COMMON /G7/ ID,IMN,IDMP,NDTU,NSKD,NSTR,NETR,TSTR,TETR,NSRC,NDRT,
     1  NSORC(100),NPO(100),NABA(200)
      DOUBLE PRECISION SUM1,SUM2
C
C*****SET EQUIPMENT NUMBERS FOR CARD READER(ICRD) ,    PRINTER (IPRT)
C***** , PUNCH(NPNCH), AND HISTOGRAM STORAGE(NHIS)(OPTIONAL)
C*****SET MAX LENGTH OF QSET(MAXQS), MAX NODE NUMBER(IMN), AND MAX
C*****NUMBER OF EMPIRICAL DISTRIBUTIONS(MAX)
C
      ICRD=5
      IPRT=6
      NPNCH=2
      NHIS=1
      MAXQS=5600
      IMN=300
      MAX=10
      IM1=6
      IM2=6
      IDMP=0
      ID=MAXQS/(IM1+IM2+2)
   10 CALL GASP
      GO TO 10
      END
C
      SUBROUTINE DATIN
      DIMENSION NSET(5600)
      DIMENSION NARYA(24), MDATA(6), PRAM(4), KEYS(7), KTYP(5),
     1  DUM(1408)
      DIMENSION NSUBI(10,4), NST01(99), NST02(99), NST03(99), NST04(99)
      DIMENSION NST05(99), NST06(99), NST07(99), ST01(99), ST02(99)
      DIMENSION ST03(99), ST04(99), ST05(99), ST06(99), NST08(99)
      COMMON QSET(5600)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
```

```
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      COMMON /G3/ KST1(100),XLOW(200),NREL1(300),NREL2(300),NREL(300),
     1  MREL(300),KST4(100),KST3(100),NTYPE(300),WIDTH(200),NSTS,NCTS
      COMMON /G4/ XSTUS(100),SUMCT,CSTUS(100),NCND,NCNL,NCNU,T2,C2,
     1  ITFLG,ICFLG,TYYY,TCCC,TIMTD,COSTD
      COMMON /G5/ MSTN(2000,2),MST(300),KST2(300),IGRF,JGRAF,SCAL
      COMMON /G7/ ID,IMN,IDMP,NDTU,NSKD,NSTP,NETR,TSTR,TETR,NSRC,NDRT,
     1  NSORC(100),NPO(100),NABA(200)
      EQUIVALENCE (JCELS(1,1),DUM(1)), (NSTO1(1),DUM(1)),
     1  (NSTO2(1),DUM(100)), (NSTO3(1),DUM(199)), (NSTO4(1),DUM(298)),
     2  (NSTO5(1),DUM(397)), (NSTO6(1),DUM(496)), (NSTO7(1),DUM(595)),
     3  (NSTO8(1),DUM(1288)), (STO1(1),DUM(694)), (STO2(1),DUM(793)),
     4  (STO3(1),DUM(892)), (STO4(1),DUM(991)), (STO5(1),DUM(1090)),
     5  (STO6(1),DUM(1189))
      EQUIVALENCE (NSET(1),QSET(1))
      DATA PP/1HP/,DD/1HD/,UU/1HU/,AA/1HA/,HH/1HH/,BLK/1H /
      DATA MDATA(1),MDATA(2),MDATA(3),MDATA(4),MDATA(5),MDATA(6)/1HF,
     1  1HA,1HB,1HI,1HD,1HC/
      DATA KEYS(1),KEYS(2),KEYS(3),KEYS(4),KEYS(5),KEYS(6)/4HNEW ,
     1  4HRUN ,4HSTOP,4HHIST,4HSAVE,4HEDIT/,KEYS(7)/4HPUNC/
      DATA KTYP(1),KTYP(2),KTYP(3),KTYP(4),KTYP(5)/4HPARM,4HARCS,4HNODE,
     1  4HCNOD,4HONE /
      DATA KNO,KYES/2HNO,3HYES/,IBLK/1H /
      DOUBLE PRECISION SUM1,SUM2
C
      IF (IDMP.GE.2) CALL ERROR (-IDMP)
      NRUN=1
      KSSP=0
      ITED=0
      ICHNG=0
      ITFLG=0
      ICFLG=0
      EPS=.00001
C   EPS IS THE THRESHOLD VALUE WITHIN WHICH 2 OR MORE EVENTS ARE
C   CONSIDERED TO BE TIED, OR TO HAVE OCCURED "SIMULTANEOUSLY".
C   THE PROGRAM CAN EASILY BE CHANGED SO THAT THIS VALUE CAN BE
C   SPECIFIED ON INPUT. IT IS ASSUMED THAT EVENTS E1, E2, E3 SUCH
C   THAT TIME(E2)=TIME((E1)+EPS, TIME(E3)=TIME(E2)+EPS, ARE ALSO
C   CONSIDERED TO BE TIED ALTHOUGH TIME(E3)-TIME(E1) = 2*EPS.
C   BY DIVIDING THE INPUT VALUE OF EPS BY N, THEN AS MANY AS N
C   EVENTS CAN BE CONSIDERED AS TIED IN THE SENS SPECIFIED ON INPUT.
      DO 10 K=1,200
         WIDTH(K)=0.
   10 CONTINUE
   20 READ (ICRD,1410,END=1300,ERR=1300) KWORD,NOD,KHIS
      WRITE (IPRT,1400)
   30 DO 40 K=1,7
         IF (KWORD.EQ.KEYS(K)) GO TO 60
   40 CONTINUE
      WRITE (IPRT,1420) KWORD
   50 CALL EXIT
   60 IF (K.EQ.5.OR.K.EQ.7) WRITE (IPRT,1430) KWORD,NOD,KHIS
      IF (K.NE.5.AND.K.NE.7) WRITE (IPRT,1430) KWORD
      GO TO (130,1290,50,70,80,90,80), K
```

ORIGINAL PAGE IS OF POOR QUALITY

```
   70 CALL GREAD
      GO TO 20
   80 CALL GSAVE (NOD,KHIS,K,KWORD)
      GO TO 20
   90 ICHNG=1
      KSSP=0
      KSUBS=0
      WRITE (IPRT,1440) NAME
  100 READ (ICRD,1410,END=1300,ERR=1300) KWORD,NOD,KHIS
      DO 110 K=1,5
         IF (KTYP(K).EQ.KWORD) GO TO 120
  110 CONTINUE
      ICHNG=0
      GO TO 30
  120 WRITE (IPRT,1430) KWORD
      GO TO (630,700,220,1180,140), K
  130 ITED=0
C
C*****DATA CARD TYPE NEW-1
C
  140 READ (ICRD,1450,END=1300,ERR=1300) NAME,NPRT,NSNK,NSTS,MODI,IGRF,
     1  NCND,NCTS,NSBS,IDMP,ISED,NRNS,NSTR,NETR,TSTR,TETR,SCAL
      WRITE (IPRT,1460) NAME
      IF (NRNS.LE.0) NRNS=1
      IF (SCAL.EQ.0.) SCAL=1.
      IF (TSTR.GT.0..AND.NSTR.GT.0) GO TO 160
      IF (TSTR.LE.0..AND.NSTR.LE.0) GO TO 160
      IF (TSTR.GT.0) GO TO 150
      TSTR=0.
      TETR=99999999.
      GO TO 160
  150 NSTR=1
      NETR=NRNS
  160 IF (NPRT.EQ.0.) GO TO 170
      WRITE (IPRT,1470) NSNK,NSTS,MODI,IGRF,NCND,NCTS,NSBS,IDMP,ISED
      WRITE (IPRT,1480) NRNS,NSTR,NETR,TSTR,TETR,SCAL
  170 IF (NETR.GE.NSTR) GO TO 180
      WRITE (IPRT,1490)
      ITED=1
  180 IF (TETR.GE.TSTR) GO TO 190
      WRITE (IPRT,1490)
      ITED=1
  190 IF (NSBS.LE.10) GO TO 200
      WRITE (IPRT,1500)
      ITED=1
  200 IF (ICHNG.EQ.1) GO TO 100
      DO 210 I=1,IMN
         KST2(I)=0
         NTYPE(I)=0
  210 CONTINUE
      NSKND=0
      NSKD=NSNK
      NSTD=NSNK+NCND
      NSTND=NSTS
      NDRT=0
```

```
      NSTC=NSNK+NCND
C
C*****NSKND--SINK NODE AND NOT A DELAY NODE
C*****NSKD--SINK AND DELAY NODE
C*****NSTD--STAT AND DELAY NODE
C*****NSTND--STAT AND NOT EITHER A DELAY OR AN ACCUMULATOR NODE
C*****NSTC--ACCUMULATOR NODE
C
      NDT=0
      NNCN=0
      NSTST=0
C
C*****NDT--NUMBER OF DELAY TYPE NODES
C*****NNCN--NUMBER OF ACCUMULATOR TYPE NODES
C*****NSTST--NUMBER OF STAT,SINK, AND ACCUMULATOR TYPE NODES
C
      NOQ=0
      NSRC=0
      KSUBS=0
      IPASS=0
      NSSP=0
      KSSP=0
      JGRAF=0
  220 IF (NPRT.NE.0.) WRITE (IPRT,1510)
C
C*****DATA CARD TYPE NEW-2 AND 2B.
C
  230 READ (ICRD,1520,END=1300,ERR=1300) NODE,INOUT,MREL1,MREL2,PTOPT,
     1   PRMV,XLL,WIDH,MSINK,XLC,WIDC,NUNOD
C
C*****NOTE--THE LAST CARD OF THIS TYPE MUST HAVE '0' IN COLUMN 3
C      INOUT CODE  1=SOURCE, 2=SINK, 3=STAT, 4=MARK,  0=OTHERWISE
C      MSINK     KST3                     DESCRIPTION
C        F         1       STAT NODE FOR 1ST STATISTICS
C        A         2       STAT NODE FOR ALL STATISTICS
C        B         3       STAT NODE FOR BETWEEN STATISTICS
C        I         4       STAT NODE FOR INCREMENT STATISTICS
C        D         5       STAT NODE FOR DELAY NODE STATISTICS
C        C         6       STAT NODE FOR ACCUMULATION NODE STATISTICS
C*****FOR ACTIVITIES COMING INTO A D-NODE, WE CANNOT HAVE COUNTER TYPES
C*****THE SAME APPLIES TO ACCUMULATOR TYPE NODES
C
      IF (ICHNG.EQ.1.AND.NODE.EQ.0) GO TO 100
      IF (PTOPT.EQ.PP) GO TO 240
      IF (PTOPT.EQ.BLK) PTOPT=DD
      IF (PTOPT.EQ.DD) GO TO 240
      ITED=1
  240 IF (NODE.EQ.0) GO TO 480
      NRR=0
      IF (KSUBS.GT.0) GO TO 540
  250 IF (NOQ.LT.NODE) NOQ=NODE
      IF (MREL1.EQ.0) MREL1=1
      IF (MREL2.EQ.0) MREL2=1
      NREL1(NODE)=MREL1
      NREL2(NODE)=MREL2
```

```
      IF (INOUT.EQ.0) GO TO 420
      IF (ICHNG.LE.0) GO TO 260
      KKK=KST2(NODE)
      GO TO 410
  260 GO TO (310,270,270,300), INOUT
  270 NSTST=NSTST+1
      IF (MSINK.EQ.IBLK) MSINK=MDATA(1)
      DO 280 INPT=1,6
         IF (MSINK.EQ.MDATA(INPT)) GO TO 290
  280 CONTINUE
      WRITE (IPRT,1530) NODE,MSINK
      ITED=1
  290 IF (INOUT.GT.2) GO TO 360
      GO TO 320
  300 KST2(NODE)=-1
      GO TO 420
  310 NSRC=NSRC+1
      NSORC(NSRC)=NODE
      GO TO 420
  320 IF (INPT.LE.4) GO TO 350
      KKK=NSKD
      NSKD=NSKD-1
  330 NDT=NDT+1
  340 KST3(KKK)=INPT
      GO TO 400
  350 NSKND=NSKND+1
      KKK=NSKND
      IF (NSKND.GT.NSNK) GO TO 390
      GO TO 340
  360 IF (INPT.LE.4) GO TO 380
      IF (INPT.GT.5) GO TO 370
      NSTD=NSTD+1
      KKK=NSTD
      GO TO 330
  370 KKK=NSTC
      NSTC=NSTC-1
      NNCN=NNCN+1
      IF (NSTC.LT.NSNK) GO TO 390
      GO TO 340
  380 KKK=NSTND
      NSTND=NSTND-1
      IF (NSTND-NSNK.GE.NCND) GO TO 340
  390 WRITE (IPRT,1540) NODE
      ITED=1
      GO TO 420
  400 KST1(KKK)=NODE
      KST2(NODE)=KKK
  410 XLOW(KKK)=XLL
      WIDTH(KKK)=WIDH
      ISUB=KKK+NSTS
      XLOW(ISUB)=XLC
      WIDTH(ISUB)=WIDC
  420 IF (NPRT.GT.0.) WRITE (IPRT,1550) NODE,NREL1(NODE),NREL2(NODE),
     1  PRMV,PTOPT,MSINK
      IF (ICHNG.EQ.1) GO TO 230
```

```
      IF (PRMV.EQ.HH) GO TO 440
      IF (PRMV.NE.AA) GO TO 430
      NRR=2
      GO TO 450
  430 IF (PRMV.NE.UU) GO TO 450
      NRR=3
      GO TO 450
  440 NRR=1
  450 IF (PTOPT.EQ.PP) GO TO 460
      NTOPT=1
      GO TO 470
  460 NTOPT=2
  470 NTYPE(NODE)=2*NRR+NTOPT
      IF (KSSP.GT.1) GO TO 570
      GO TO 230
  480 IF (IPASS) 490,490,550
C
C*****TEST IF THERE ARE SUBSYSTEMS.
C
  490 IF (NSBS.EQ.0) GO TO 610
  500 KSSP=0
      NNDS=0
C
C*****DATA CARD TYPE NEW-2A.
C*****READ IN SUBSYSTEM INFORMATION.
C*****NNSUB=NUMBER OF NODES IN PARTICULAR SUBSYSTEM.
C*****NSSP=NUMBER OF TIMES PARTICULAR SUBSYSTEM OCCURS.
C*****NACTS=NUMBER OF ACTIVITIES IN PARTICULAR SUBSYSTEM.
C
      READ (ICRD,1560,END=1300,ERR=1300) NNSUB,NSSP,NACTS
C
C*****TEST FOR ERROR CONDITIONS.
C
      IF (NNSUB.LE.99) GO TO 510
      WRITE (IPRT,1570)
      ITED=1
  510 IF (NSSP.LE.20) GO TO 520
      WRITE (IPRT,1580)
      ITED=1
  520 IF (NACTS.LE.99) GO TO 530
      WRITE (IPRT,1590)
      ITED=1
  530 IPASS=1
      KSSP=KSSP+1
      KSUBS=KSUBS+1
C
C*****STORE SUBSYSTEM INFORMATION.
C
      NSUBI(KSUBS,1)=KSUBS
      NSUBI(KSUBS,2)=NSSP
      NSUBI(KSUBS,3)=NNSUB
      NSUBI(KSUBS,4)=NACTS
C
C*****WRITE OUT SUBSYSTEM HEADING.
C
```

```
      IF (NPRT.GT.0.) WRITE (IPRT,1600) KSUBS,KSSP
      GO TO 230
C
C*****READ IN SUBSYSTEM NODE DATA.
C*****THE LAST CARD OF THIS TYPE MUST HAVE A ZERO IN FIELD 1.
C
  540 NNDS=NNDS+1
C
C*****STORE SUBSYSTEM NODE INFORMATION.
C*****STO-( ) AND NSTO-( ) ARE THE TEMPORARY ARRAYS WHERE THE NODE AND
C*****ARC INFORMATION NEEDED FOR SUBSYSTEM GENERATION IS STORED.
C
      NSTO1(NNDS)=NODE
      NSTO2(NNDS)=INOUT
      NSTO3(NNDS)=MREL1
      NSTO4(NNDS)=MREL2
      STO1(NNDS)=PTOPT
      STO2(NNDS)=PRMV
      STO3(NNDS)=XLL
      STO4(NNDS)=WIDH
      NSTO5(NNDS)=MSINK
      STO5(NNDS)=XLC
      STO6(NNDS)=WIDC
      NSTO6(NNDS)=NUNOD
C
C*****PROCESS NODE INFORMATION.
C
      GO TO 250
  550 IF (NNDS.EQ.NNSUB) GO TO 560
      WRITE (IPRT,1610)
      ITED=1
C
C*****TEST IF ALL SUBSYSTEMS OF THIS TYPE ARE GENERATED.
C
  560 IF (KSSP.EQ.NSSP) GO TO 590
      NNDS=0
      KSSP=KSSP+1
C
C*****GENERATE NEW SUBSYSTEM (NODE INFORMATION).
C
      IF (NPRT.GT.0.) WRITE (IPRT,1600) KSUBS,KSSP
  570 IF (NNDS.EQ.NNSUB) GO TO 560
      NNDS=NNDS+1
      NODE=NSTO1(NNDS)+NNSUB
      NSTO1(NNDS)=NODE
      INOUT=NSTO2(NNDS)
      MREL1=NSTO3(NNDS)
      MREL2=NSTO4(NNDS)
      PTOPT=STO1(NNDS)
      PRMV=STO2(NNDS)
      XLL=STO3(NNDS)
      WIDH=STO4(NNDS)
      MSINK=NSTO5(NNDS)
      XLC=STO5(NNDS)
      WIDC=STO6(NNDS)
```

```
      NUNOD=NST06(NNDS)
C
C*****PROCESS NEW SUBSYSTEM NODE INFORMATION.
C*****FIRST TEST TO SEE IF NODE INFORMATION IS TO BE CHANGED.  IF NOT
C*****PROCEED.  IF SO, READ IN NEW DATA AND UPDATE VARIABLES. THEN
C*****PROCEED.
C
      IF (NUNOD-KSSP) 250,580,250
C
C*****DATA CARD TYPE NEW-2C.
C
  580 READ (ICRD,1620,END=1300,ERR=1300) INOUT,MREL1,MREL2,PTOPT,PRMV,
     1  XLL,WIDH,MSINK,XLC,WIDC,NUNOD
      NST06(NNDS)=NUNOD
      GO TO 250
C
C*****IF ALL SUBSYSTEMS OF A PARTICULAR TYPE ARE PROCESSED, CHECK TO SEE
C*****IF ALL DIFFERENT TYPES ARE PROCESSED.  IF ALL ARE PROCESSED,
C*****PROCEED, IF NOT CONTINUE.
C
  590 IF (KSUBS.NE.NSBS) GO TO 500
      IPASS=0
      KSUBS=0
      KSSP=0
      IF (NCND.LE.NNCN) GO TO 600
      WRITE (IPRT,1630)
      ITED=1
  600 IF (NSTS.LE.NSTST) GO TO 610
      WRITE (IPRT,1640)
      ITED=1
  610 IF (NPRT.GT.0.) WRITE (IPRT,1650) (NSORC(J),J=1,NSRC)
      IF (NPRT.GT.0.) WRITE (IPRT,1660) (KST1(K),K=1,NSNK)
      IF (NSNK.EQ.NSTS) GO TO 620
      KPP=NSNK+1
      IF (NPRT.GT.0.) WRITE (IPRT,1670) (KST1(K),K=KPP,NSTS)
  620 IM=IM1-1
      MXC=32
      NPRMS=0
C
C*****DATA CARD TYPE NEW-3.
C
  630 IF (NPRT.GT.0.) WRITE (IPRT,1680)
  640 READ (ICRD,1690,END=1300,ERR=1300) IPARS,(PRAM(J),J=1,4)
      IF (ICHNG.EQ.1.AND.IPARS.EQ.0) GO TO 100
      IF (IPARS.EQ.0) GO TO 660
      DO 650 J=1,4
         PARM(IPARS,J)=PRAM(J)
  650 CONTINUE
      IF (NPRT.GT.0.) WRITE (IPRT,1700) IPARS,(PRAM(J),J=1,4)
      IF (IPARS.LE.NPRMS) GO TO 640
      NPRMS=IPARS
      GO TO 640
C
C*****INITIALIZE NSET
C
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
  660 KOL=77777
      KOF=88888
      MX1=IM1+IM2+2
      MX2=2*(MX1-1)+1
      INDX=0
      DO 670 J=1,ID
         INDX=INDX+MX1
         NSET(INDX)=INDX+1
         NSET(INDX-1)=INDX-MX2
  670 CONTINUE
      NSET(IM1+IM2+1)=0
      NSET(INDX)=KOF
      DO 680 I=1,ID
         MSTN(I,1)=0
         MSTN(I,2)=I+1
  680 CONTINUE
      MSTN(ID,2)=KOF
      DO 690 K=1,NDQ
         MST(K)=0
         NQ(K)=0
         MFE(K)=0
         MLE(K)=0
  690 CONTINUE
      MFA=1
      MFAN=1
      NDTL=NSKD+1
      NDTU=NSKD+NDT+NCND
      NCNL=NSTC+1
      NCNU=NCNL+NCND-1
C
C*****WRITE MAIN HEADING AND INITIALIZE STATISTICAL POINTERS
C
      NCLT=NSTS*(NCTS+2)+2
      NHIST=NSTS*2+2
  700 IF (NPRT.GT.0.) WRITE (IPRT,1710)
      NUACT=0
C
C*****DATA CARD TYPE NEW-4 AND 4A.
C
  710 READ (ICRD,1720,END=1300,ERR=1300) ATRIB(1),JQ,(JTRIB(JK),JK=1,
     1  IM),ATRIB(3),ATRIB(4),NUACT,JTRIB(6),IRNK
      IF (ICHNG.EQ.1.AND.JQ.EQ.0) GO TO 100
      IF (JTRIB(3).EQ.0) JTRIB(3)=10
      IF (ATRIB(1).LE.0.) ATRIB(1)=1.0
      IF (ATRIB(3).GT.0.) JGRAF=1
      IF (ATRIB(4).GT.0.) JGRAF=1
      ATRIB(2)=0.
      ATRIB(5)=0.
      IF (JQ.EQ.0) GO TO 910
      JQ=IABS(JQ)
      IF (KSUBS.GT.0) GO TO 930
C
C*****PROCESS ARC INFORMATION.
C
  720 JD=JTRIB(3)
```

```
      IF (JD.EQ.10) GO TO 840
      IF (JD.EQ.13) GO TO 840
      J=JTRIB(2)
      IF (J.GT.0) GO TO 730
      JTRIB(2)=-JTRIB(2)
      GO TO 840
  730 IF (PARM(J,4).GE.0.0) GO TO 740
      WRITE (IPRT,1730) J
      GO TO 840
  740 GO TO (840,840,840,840,750,840,780,790,800,840,810,820,840), JD
  750 PARM(J,4)=ALOG((PARM(J,4)/PARM(J,1))**2+1.0)
      PARM(J,1)=ALOG(PARM(J,1))-.5*PARM(J,4)
      PARM(J,4)=SQRT(PARM(J,4))
      IF (PARM(J,2).GT.0.) GO TO 760
      PARM(J,2)=-1.0E20
      GO TO 770
  760 PARM(J,2)=ALOG(PARM(J,2))
  770 PARM(J,3)=ALOG(PARM(J,3))
      GO TO 830
  780 CALL BETAXF
      GO TO 830
  790 TEMP=PARM(J,4)**2
      PARM(J,4)=PARM(J,1)**2/TEMP
      PARM(J,1)=PARM(J,1)/TEMP
      GO TO 830
  800 CALL PERTXF
      GO TO 830
  810 PARM(J,4)=PARM(J,3)-PARM(J,2)
      PARM(J,1)=(PARM(J,1)-PARM(J,2))/PARM(J,4)
      GO TO 830
  820 PARM(J,1)=1./PARM(J,1)
      PARM(J,4)=1./PARM(J,4)
  830 PARM(J,4)=-PARM(J,4)
  840 IF (ICHNG.LE.0) GO TO 870
      IF (IRNK.EQ.0) IRNK=1
      INDX=NFIND(JTRIB(1),JQ,1,IRNK)
      IF (INDX.EQ.0) CALL ERROR (13)
      DO 850 K=1,IM1
         NSET(INDX)=JTRIB(K)
         INDX=INDX+1
  850 CONTINUE
      DO 860 K=2,IM2
         QSET(INDX+1)=ATRIB(K)
         INDX=INDX+1
  860 CONTINUF
      GO TO 900
  870 KCOLN=MFA
      CALL FILEM (JQ)
      JJ=JTRIB(1)
      IF (NTYPE(JJ).LE.4) GO TO 900
      NDRT=1
      MFANT=MSTN(MFAN,2)
      IF (MST(JJ).EQ.0) GO TO 880
      MSTN(MFAN,2)=MST(JJ)
      GO TO 890
```

```
  880 MSTN(MFAN,2)=KOL
  890 MSTN(MFAN,1)=KCOLN
      MST(JJ)=MFAN
      MFAN=MFANT
C
C*****WRITE INDIVIDUAL ARC.
C
  900 IF (NPRT.GT.0.) WRITE (IPRT,1740) JQ,(JTRIB(JK),JK=1,IM),ATRIB(1),
     1  ATRIB(3),ATRIB(4),JTRIB(6)
      IF (KSSP.GT.1) GO TO 940
      GO TO 710
C
C*****TEST TO SEE IF THERE ARE SUBSYSTEMS.
C
  910 IF (IPASS.GT.0) GO TO 980
      IF (NSBS.EQ.0) GO TO 1010
  920 KSSP=0
      KACTS=0
      IPASS=1
      KSSP=KSSP+1
      KSUBS=KSUBS+1
C
C*****WRITE SUBSYSTEM HEADING.
C
      IF (NPRT.GT.0.) WRITE (IPRT,1600) KSUBS,KSSP
      GO TO 710
C
C*****READ ARC OF SUBSYSTEM. LAST CARD MUST HAVE A ZERO IN FIELD 1.
C
  930 KACTS=KACTS+1
C
C*****STORE SUBSYSTEM ARC INFORMATION.
C
      NST01(KACTS)=NUACT
      ST01(KACTS)=ATRIB(1)
      NST02(KACTS)=JQ
      NST03(KACTS)=JTRIB(1)
      NST04(KACTS)=JTRIB(2)
      NST05(KACTS)=JTRIB(3)
      NST06(KACTS)=JTRIB(4)
      NST07(KACTS)=JTRIB(5)
      ST02(KACTS)=ATRIB(3)
      ST03(KACTS)=ATRIB(4)
      NST08(KACTS)=JTRIB(6)
      GO TO 720
C
C*****TEST IF ALL OF PARTICULAR SUBSYSTEM'S ACTIVITIES ARE IN. IF NOT,
C*****THEN GENERATE NEXT ONE.
C
  940 IF (KACTS.EQ.NSUBI(KSUBS,4)) GO TO 990
  950 KACTS=KACTS+1
      JQ=NST02(KACTS)+NSUBI(KSUBS,3)
      NST02(KACTS)=JQ
      ATRIB(1)=ST01(KACTS)
      JTRIB(1)=NST03(KACTS)+NSUBI(KSUBS,3)
```

```
      NST03(KACTS)=JTRIB(1)
      JTRIB(2)=NST04(KACTS)
      JTRIB(3)=NST05(KACTS)
      JTRIB(4)=NST06(KACTS)
      JTRIB(5)=NST07(KACTS)
      ATRIB(3)=ST02(KACTS)
      ATRIB(4)=ST03(KACTS)
      NUACT=NST01(KACTS)
      JTRIB(6)=NST08(KACTS)
      IF (JTRIB(6).LE.0) GO TO 960
      JTRIB(6)=JTRIB(6)+NSUBI(KSUBS,3)
  960 NST08(KACTS)=JTRIB(6)
C
C*****PROCESS NEW SUBSYSTEM ARC INFORMATION
C*****FIRST TEST TO SEE IF ARC IS TO BE CHANGED.  IF NOT PROCEED.
C*****IF SO READ IN NEW DATA AND UPDATE VARIABLES AND THEN PROCEED.
C
      IF (NUACT-KSSP) 720,970,720
C
C*****DATA CARD TYPE NEW-4B.
C
  970 READ (ICRD,1750,END=1300,ERR=1300) ATRIB(1),(JTRIB(JK),JK=2,IM),
     1  ATRIB(3),ATRIB(4),NUACT,JTRIB(6)
      NST01(KACTS)=NUACT
      IF (JTRIB(3).EQ.0) JTRIB(3)=10
      IF (ATRIB(1).LE.0.) ATRIB(1)=1.0
      IF (ATRIB(3).GT.0..OR.ATRIB(4).GT.0.) JGRAF=1
      GO TO 720
  980 IF (KACTS.EQ.NSUBI(KSUBS,4)) GO TO 990
C
C*****IF ALL SUBSYSTEM ACTIVITIES NOT READ IN CORRECTLY,
C*****PRINT ERROR MESSAGE.
C
      WRITE (IPRT,1760)
      ITED=1
  990 IF (KSSP.EQ.NSUBI(KSUBS,2)) GO TO 1000
      KACTS=0
C
C*****GENERATE NEW SUBSYSTEM.
C
      KSSP=KSSP+1
C
C*****WRITE NEW SUBSYSTEM HEADING AND THEN GENERATE NEXT SUBSYSTEM.
C
      IF (NPRT.GT.0.) WRITE (IPRT,1600) KSUBS,KSSP
      GO TO 950
C
C*****SEE IF ALL SUBSYSTEMS GENERATED.
C
 1000 IF (KSUBS.NE.NSBS) GO TO 920
C
C*****TEST PROBABILITIES AND PROCEED.
C
      KSUBS=0
 1010 DO 1090 JQ=2,NOQ
```

```
          PROB=0.0
          LINE=MFE(JQ)
          IF (LINE.LE.0) GO TO 1090
 1020     NT=NTYPE(JQ)
          IE=LINE+IM
          IQ=IE+1
          DPROB=QSET(IQ)
          GO TO (1040,1030,1040,1030,1040,1030,1040,1030), NT
 1030     QSET(IQ)=QSET(IQ)+PROB
          PROB=PROB+DPROB
 1040     INDX=LINE+IM1+IM2+1
          LINE=NSET(INDX)
          IF (LINE-77777) 1020,1060,1050
 1050     WRITE (IPRT,1770)
          CALL EXIT
 1060     GO TO (1090,1070,1090,1070,1090,1070,1090,1070), NT
 1070     IF (PROB.LT..998) GO TO 1080
          IF (PROB.LE.1.002) GO TO 1090
 1080     WRITE (IPRT,1760) JQ
          ITED=1
 1090 CONTINUE
      MMACT=100
      DO 1100 J=1,MMACT
          NPO(J)=0
 1100 CONTINUE
      IF (MODI.LE.0) GO TO 1170
      IF (NPRT.GT.0.) WRITE (IPRT,1790)
      NPOS=0
 1110 NPOS=NPOS+1
C
C*****DATA CARD TYPE NEW-5.
C
      READ (ICRD,1800,END=1300,ERR=1300) NACTN,NABYA
      IF (NACTN) 1270,1160,1120
 1120 NPO(NACTN)=NPOS
 1130 DO 1140 I=1,24,2
          IF (NABYA(I).LE.0) GO TO 1150
          NABA(NPOS)=NABYA(I)
          NABA(NPOS+1)=NABYA(I+1)
          KNT1=I+1
          NPOS=NPOS+2
 1140 CONTINUE
 1150 NABA(NPOS)=0
      IF (NPRT.GT.0.) WRITE (IPRT,1810) NACTN,(NABYA(J),J=1,KNT1)
      GO TO 1110
 1160 NABA(NPOS)=0
C
C*****DATA CARD TYPE NEW-6.
C
 1170 IF (NCND.LE.0) GO TO 1250
 1180 READ (ICRD,1820,END=1300,ERR=1300) T2,C2,XLT1,WTT1,XLC1,WTC1,
     1   ITFLG,ICFLG
      NWWW=NHIST-1
      XLOW(NWWW)=XLT1
      WIDTH(NWWW)=WTT1
```

```
      XLOW(NHIST)=XLC1
      WIDTH(NHIST)=WTC1
      IF (ITFLG.NE.0) ITFLG=2
      IF (ICFLG.NE.0) ICFLG=2
      IF (WTC1.GT.0.) GO TO 1190
      IF (NPRT.GT.0.) WRITE (IPRT,1830)
 1190 IF (WTT1.GT.0.) GO TO 1200
      IF (NPRT.GT.0.) WRITE (IPRT,1840)
      K=KNO
      IF (ITFLG.NE.0) K=KYES
      IF (NPRT.GT.0.) WRITE (IPRT,1850) K
      K=KNO
      IF (ICFLG.NE.0) K=KYES
      IF (NPRT.GT.1.) WRITE (IPRT,1860) K
 1200 IF (T2.EQ.0.0) GO TO 1210
      IF (T2.LT.0.0) T2=0.0
      IF (NPRT.GT.0.) WRITE (IPRT,1870) T2
      GO TO 1220
 1210 T2=-1.0
 1220 IF (C2.EQ.0.0) GO TO 1230
      IF (C2.LT.0.0) C2=0.0
      IF (NPRT.GT.0.) WRITE (IPRT,1880) C2
      GO TO 1240
 1230 C2=-1.0
 1240 IF ((T2.GE.0.0.OR.C2.GE.0.0).AND.NCND.LE.0) GO TO 1250
      GO TO 1260
 1250 T2=-1.0
      C2=-1.0
 1260 IF (ICHNG.EQ.1) GO TO 100
      GO TO 20
 1270 NPOS=NPOS-1
      GO TO 1130
 1280 CALL ERROR (95)
      GO TO 20
C
C*****INITIALIZE FOR THE NEXT RUN
C
 1290 TNOW=0.0
      IF (IDMP.EQ.1.OR.IDMP.EQ.3) CALL ERROR (-IDMP)
      IF (ITED.EQ.0.) GO TO 1310
 1300 WRITE (IPRT,1890)
      GO TO 20
 1310 DO 1320 L=1,NSTS
         CSTUS(L)=0.0
         XSTUS(L)=-1.0
 1320 CONTINUE
      RNUM=RANF(ISED)
      DO 1330 I=1,NCLT
         SUM1(I)=0.D0
         SUM2(I)=0.D0
         SUM3(I)=0.
         SUM4(I)=1.0E20
         SUM5(I)=-1.0E20
 1330 CONTINUE
      IF (NHIST) 1280,1360,1340
```

```
 1340 DO 1350 K=1,NHIST
      DO 1350 L=1,MXC
         JCELS(K,L)=0
 1350 CONTINUE
 1360 IF (NPRMS.LE.0) GO TO 1380
      DO 1370 I=1,NPRMS
         PARM(I,4)=ABS(PARM(I,4))
 1370 CONTINUE
 1380 DO 1390 K=1,NSTS
         KST4(K)=0
 1390 CONTINUE
      IF (NPRT.GT.0.) WRITE (IPRT,1400)
      RETURN
C
C
 1400 FORMAT (1H1)
 1410 FORMAT (A4,2I4)
 1420 FORMAT (///1X,24H UNRECOGNIZABLE KEYWORD=,1X,A4)
 1430 FORMAT (1X,A4,1X,I5,3X,2HAS,I5)
 1440 FORMAT (/,1X,3(1H*),'EDIT GRASP SIMULATION PROJECT :',1X,20A4,/)
 1450 FORMAT (20A4,/,9I3,/,4I10,3F10.0)
 1460 FORMAT (' GRASP SIMULATION PROJECT : ',20A4)
 1470 FORMAT (/42X,23H**NETWORK DESCRIPTION**/,/,15X,
     1  'NUMBER OF SINK NODES: NSNK =',I3/15X,'STA',
     2  'TISTICS COLLECTED ON: NSTS =',I3,' NODES.'/15X,
     3  'ARE THERE MODIF','ICATIONS IN THE NETWORK?: MODI =',I3/15X,
     4  'TYPE OF HISTOGRAMS ','TO BE COLLECTED: IGRF =',I3/15X,
     5  'NUMBER OF ACCUM. NODE TYPES',': NCND =',I3/15X,
     6  'NUMBER OF COUNT TYPES: NCTS =',I3/15X,'NUMBER',
     7  ' OF SUBSYSTEM TYPES: NSBS =',I3/15X,'DUMPING OF FILES OPTION: I
     8','DMP =',I3,/15X,'SEED FOR INITIAL RANDOM NUMBER: ISED =',I10)
 1480 FORMAT (15X,'NUMBER OF RUNS TO END THE SIMULATION: NRNS =',I10,
     1  /15X,'RUN NUMBER TO BEGIN TRACING: NSTR =',I10/15X,
     2  'RUN NUMBER ','TO END TRACING: NETR =',I10,/15X,
     3  'TIME TO BEGIN TRACING: TSTR',' =',F10.0,/15X,
     4  'TIME TO END TRACING: TETR =',F10.0/15X,'SCALE',
     5  ' FOR CONSTANT TIMES: SCAL =',F10.0/)
 1490 FORMAT ('  INPUT ERROR: NETR < NSTR , OR TETR < TSTR')
 1500 FORMAT (/10X,19HNSBS EXCEEDS LIMIT/)
 1510 FORMAT (//,42X,'**NODE DESCRIPTION**',//,5H NODE,4X,6HNUMBER,5X,
     1  18HNUMBER OF RELEASES,4X,5HA=AND,11X,6HOUTPUT,4X,
     2  16HSTATISTICS TYPE /8X,8HRELEASES,8X,4HFOR ,6HREPEAT,5X,
     3  17HH= HALT , U=A U H,3X,4HTYPE/)
 1520 FORMAT (4I3,2A1,2F6.2,A1,2F10.4,I3)
 1530 FORMAT (' INPUT ERROR. NODE ',I3,' HAS AN ILLEGAL STAT. CODE: ',
     1  A1)
 1540 FORMAT (/5X,4HNODE,I4,61H CAUSES NO. OF STAT / SINK NODES TO EXCEE
     1D THE NO. SPECIFIED.)
 1550 FORMAT (1X,I4,I9,I18,13X,A1,13X,A1,14X,A3,A4,I4)
 1560 FORMAT (3I5)
 1570 FORMAT (/10X,20HNNSUB LIMIT EXCEEDED/)
 1580 FORMAT (/10X,19HNSSP LIMIT EXCEEDED/)
 1590 FORMAT (/10X,20HNACTS LIMIT EXCEEDED/)
 1600 FORMAT (//15X,15HSUBSYSTEM TYPE ,I5,9X,5HCOPY ,I5//)
 1610 FORMAT (/10X,31HALL SUBSYSTEM NODES NOT READ IN/)
```

```
1620 FORMAT (3X,3I3,2A1,2F6.2,A1,2F10.4,I3)
1630 FORMAT (/5X,36HALL ACCUMULATION NODES NOT PROCESSED/)
1640 FORMAT (/5X,42HALL SINK AND STATISTIC NODES NOT PROCESSED/)
1650 FORMAT (/15X,19HSOURCE NODE NUMBERS/(17X,10I5))
1660 FORMAT (/15X,17HSINK NODE NUMBERS/(17X,10I5))
1670 FORMAT (/15X,34HSTATISTICS COLLECTED ALSO ON NODES/(17X,20I5))
1680 FORMAT (//39X,23H**  ARC PARAMETERS   **//16X,9HPARAMETER,32X,
    1  10HPARAMETERS/17X,6H SET  ,16X,1H1,13X,1H2,13X,1H3,13X,1H4/)
1690 FORMAT (I3,4F10.4)
1700 FORMAT (1X,I20,7X,4F14.4)
1710 FORMAT (//42X,24H**  ARC DESCRIPTION   **//3X,5HSTART,2X,3HEND,3X,
    1  9HPARAMETER,2X,12HDISTRIBUTION,2X,5HCOUNT,2X,8H   ARC  ,2X,
    2  11HPROBABILITY,4X,5HSETUP,5X,8HVARIABLE,4X,
    3  15HASSOCIATED COST/4X,4HNODE,2X,4HNODE,4X,6H SET  ,7X,4HTYPE,7X,
    4  4HTYPE,3X,6HNUMBER,18X,4HCOST,8X,4HCOST,11X,6HC-NODE/)
1720 FORMAT (F8.3,6I3,2F9.2,2I3,28X,I2)
1730 FORMAT (/10X,13HPARAMETER SET,I5,3X,22HWAS MODIFIED ONLY ONCE)
1740 FORMAT (1X,I5,I7,I10,I11,2I10,F13.4,2F11.2,11X,I3)
1750 FORMAT (F8.3,6X,4I3,2F9.2,2I3)
1760 FORMAT (/10X,36HALL SUBSYSTEM ACTIVITIES NOT READ IN/)
1770 FORMAT (///36X,26HERROR EXIT, TYPE 70 ERROR.)
1780 FORMAT (/5X,4HNODE,I4,59H HAS OUTPUT BRANCHES WHOSE PROBABILITIES
    1DO NOT SUM TO ONE./)
1790 FORMAT (//39X,25H**NETWORK MODIFICATIONS**//1X,8H  ARC   ,12(1X,
    1  9HNODE NODE)/9X,12(1X,9HOUT   IN )/)
1800 FORMAT (25I3)
1810 FORMAT (1X,I4,2X,12(I5,I5))
1820 FORMAT (2F10.4,2F6.2,2F10.4,2I2)
1830 FORMAT (/5X,41HC-NODE TOTAL COST HISTOGRAM HAS NO VALUES)
1840 FORMAT (/5X,41HC-NODE TOTAL TIME HISTOGRAM HAS NO VALUES)
1850 FORMAT (/5X,37HWILL T2 CRITERION END SIMULATION RUN-,A4)
1860 FORMAT (/5X,37HWILL C2 CRITERION END SIMULATION RUN-,A4)
1870 FORMAT (////15X,22HT2 VALUE FOR C-NODE IS,5X,F10.4)
1880 FORMAT (//15X,22HC2 VALUE FOR C-NODE IS,5X,F10.4)
1890 FORMAT (/,' DATA SET CANNOT BE PROCESSED BECAUSE OF AN ERROR',
    1  ' IN THE DATA OR IN THE NETWORK MODEL')
     END
C
     SUBROUTINE GASP
     DIMENSION NSET(5600), NEND(50), JJJ(50), NDR(300), NRP(300),
    1  KOUNT(250)
     COMMON QSET(5600)
     COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NDQ,NSNK,
    1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
     COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
    1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
     COMMON /G3/ KST1(100),XLOW(200),NREL1(300),NREL2(300),NREL(300),
    1  MREL(300),KST4(100),KST3(100),NTYPE(300),WIDTH(200),NSTS,NCTS
     COMMON /G4/ XSTUS(100),SUMCT,CSTUS(100),NCND,NCNL,NCNU,T2,C2,
    1  ITFLG,ICFLG,TYYY,TCCC,TIMTD,COSTD
     COMMON /G5/ MSTN(2000,2),MST(300),KST2(300),IGRF,JGRAF,SCAL
     COMMON /G6/ NCEL(10),PROB(32,10),VAL(33,10),ISEC(3,10),MAX,NHIS,
    1  KHIST,NPNCH
     COMMON /G7/ ID,IMN,IDMP,NDTU,NSKD,NSTR,NETR,TSTR,TETR,NSRC,NDRT,
    1  NSORC(100),NPO(100),NABA(200)
```

```
      EQUIVALENCE (NSET(1),QSET(1))
      DOUBLE PRECISION SUM1,SUM2
C
      DO 10 MU=1,IM1
         JTRIB(MU)=0
   10 CONTINUE
      DO 20 M=1,IM2
         ATRIB(M)=0.
   20 CONTINUE
      DO 30 I=1,IMN
         NREL1(I)=0
         NREL2(I)=0
   30 CONTINUE
      DO 40 I=1,100
         KST4(I)=0
         KST3(I)=0
         KST1(I)=0
      DO 40 J=1,4
         PARM(I,J)=0
   40 CONTINUE
C    INITIALIZE ISIMX=MAXIMUM VALUE FOR ISIM=MAXIMUM NUMBER OF TIED
C    EVENTS THAT CAN BE HANDLED=DIMENSION OF NEND=100
      ISIMX=50
C    MAXRP IS EQUAL TO 3 * THE MAXIMUM NUMBER OF NODE REPLACEMENTS.
      MAXRP=300
      CALL DATIN
      TIMTD=0.0
      COSTD=0.0
      TCCC=0.0
      TYYY=0.0
      NT2R=0
      NC2R=0
      NT2C2=0
      KHIST=0
      NCLT=NSTS*(NCTS+2)+2
      MCLT=NCLT-NSTS-2
C
C*****INITIALIZE FOR THIS RUN OF THE SIMULATION
C
   50 DO 60 MU=1,IM1
         JTRIB(MU)=0
   60 CONTINUE
      DO 70 M=1,IM2
         ATRIB(M)=0.
   70 CONTINUE
      CALL FILEM (1)
      TNOW=0.
      DO 80 I=1,MAXRP,3
         NRP(I)=0
         NRP(I+1)=I-3
         NRP(I+2)=I+3
   80 CONTINUE
      NRP(MAXRP)=1
      NRP(2)=MAXRP-2
      IP1=1
```

```
      NSTSR=1
      IF (NCTS.LE.0) GO TO 100
      DO 90 J=1,NCTS
         KOUNT(J)=0
   90 CONTINUE
  100 DO 110 IJ=2,NOQ
         NDR(IJ)=0
         MREL(IJ)=NREL1(IJ)
         NREL(IJ)=NREL1(IJ)
  110 CONTINUE
      SUMCT=0.0
      ARCST=0.
      TOLD=0.
      ITRACE=0
      IF (NSTR.LE.NRUN.AND.NRUN.LE.NETR) ITRACE=1
      ITRC=1
      IF (TSTR.GT.0.) GO TO 120
      IF (NSTR.GT.0) ITRC=2
      GO TO 130
  120 ITRC=3
      IF (NSTR.GT.0) ITRC=4
  130 IF (ITRACE.EQ.1) WRITE (IPRT,1020) NRUN
C
C*****OBTAIN NEXT EVENT WHICH IS FIRST ENTRY IN FILE 1.  ATRIB(1) IS
C*****EVENT TIME, JTRIB(1) IS THE END NODE
C
      NDSP=0
      NEQT=0
  140 ISIM=0
  150 ISIM=ISIM+1
      KCOL=MFE(1)
      CALL RMOVE (KCOL,1)
      IF ((ATRIB(1)-TNOW).LE.EPS) GO TO 160
      NDSP=JTRIB(1)
      NEQT=1
      GO TO 170
  160 NEQT=NEQT+1
      IF (NEQT.LT.100) GO TO 170
      WRITE (IPRT,1030) NDSP,NEQT,TNOW
      CALL ERROR (10)
  170 TNOW=ATRIB(1)
      GO TO (200,200,180,190), ITRC
  180 ITRACE=0
      IF (TNOW.GT.TETR) GO TO 200
      IF (TNOW.LT.TSTR) GO TO 200
      ITRACE=1
      GO TO 200
  190 ITRACE=0
      IF (NRUN.GT.NETR) GO TO 200
      IF (NRUN.LT.NSTR) GO TO 200
      IF (TNOW.GT.TETR) GO TO 200
      IF (TNOW.LT.TSTR) GO TO 200
      ITRACE=1
  200 NEND(ISIM)=JTRIB(1)
      JJJ(ISIM)=JTRIB(5)
```

```
C
C*****IF END NODE IS ZERO START NETWORK BY SCHEDULING ACTIVITIES
C*****FROM EACH SOURCE NODE
C
      IF (NEND(ISIM)) 250,210,230
  210 DO 220 N=1,NSRC
         M=NSORC(N)
         CALL SCHAT (M)
         MREL(M)=NREL2(M)
         NREL(M)=NREL2(M)
  220 CONTINUE
      GO TO 140
C
C*****REDUCE RELEASE COUNT (MREL(NEND(ISIM))) BY 1 IF ARC NUMBER IS
C***** .GE. ZERO.  IF NEGATIVE, INCREASE BY (-JTRIB(5))
C
  230 IF (JTRIB(5).GE.0) GO TO 240
      MREL(NEND(ISIM))=MREL(NEND(ISIM))-JTRIB(5)
      GO TO 250
  240 MREL(NEND(ISIM))=MREL(NEND(ISIM))-1
  250 LL=0
      L=JTRIB(4)
      IF (L.LE.0) GO TO 260
      LL=KOUNT(L)+1
      KOUNT(L)=LL
C
C*****CHECK FOR TRACE ON THIS RUN
C
  260 COSAR=SUMCT-ARCST
      DUR=TNOW-TOLD
      IF (ITRACE.EQ.1) WRITE (IPRT,1040) TNOW,JTRIB(5),(JTRIB(N),N=1,4),
     1  LL,MREL(NEND(ISIM)),COSAR,DUR
      ARCST=SUMCT
      TOLD=TNOW
C
C*****ARC HAS BEEN COMPLETED.  IF A C-NODE IS ASSOCIATED WITH THIS
C*****ARC, UPDATE THE C-NODE RUNNING COST.
C
      IF (JTRIB(6).LE.0) GO TO 270
      KQ=JTRIB(6)
      KK=KST2(KQ)
      COST=ATRIB(3)+ATRIB(4)*(TNOW-ATRIB(6))
      LCLT=MCLT+KK
      CALL COLCT (COST,LCLT)
      LCLT=NSTS+KK
      CALL HISTO (COST,LCLT)
      ATRIB(6)=0.0
      IF (ITFLG.EQ.3.OR.ICFLG.EQ.3) GO TO 860
  270 IF (NEND(ISIM).LE.0) GO TO 140
      IF (MFE(1).EQ.0) GO TO 280
      INEXT=MFE(1)+6
      TIME1=QSET(INEXT)
      IF (ISIM.GT.ISIMX) CALL ERROR (98)
      IF ((TIME1-TNOW).LE.EPS) GO TO 150
C
```

```
C   REINITIALIZE THE RELEASE COUNTERS FOR OVERFLODED NODES.
C
  280 DO 300 I=1,ISIM
          IF (NREL(NEND(I)).GE.0) GO TO 290
          IF (MREL(NEND(I)).LT.NREL(NEND(I))) MREL(NEND(I))=NREL(NEND(I))
          GO TO 300
  290     IF (MREL(NEND(I)).GT.NREL(NEND(I))) MREL(NEND(I))=NREL(NEND(I))
  300 CONTINUE
C*****NETWORK MODIFICATION SECTION
C       FOR EACH NODE IN THE NETWORK GO THROUGH ALL THE ARCS THAT
C       TERMINATE AT THE SAME TIME. FOR EACH ARC, IF ANY, THAT TRIGGERS
C       A REPLACEMENT OF THIS NODE BY ANOTHER ONE, GO THROUGH THE LIST
C       OF TIED ARCS ONE MORE TIME. IF AN ARC THAT TRIGGERS THE OPPOSITE
C       REPLACEMENT CAN BE FOUND, THEN THE REPLACEMENT OF THE NODE IS
C       CANCELED. OTHERWISE THE REPLACEMENT HOLDS.
C
      DO 470 N=2,NOQ
         DO 460 I=1,ISIM
            J1=JJJ(I)
            IF (J1.LE.0) GO TO 460
            NP1=NPO(J1)
            IF (NP1.LE.0) GO TO 460
  310       NDOUT=NABA(NP1)
            IF (NDOUT.EQ.0) GO TO 460
            IF (N.NE.NDOUT) GO TO 450
            NDIN1=NABA(NP1+1)
            DO 340 II=1,ISIM
               J2=JJJ(II)
               IF (J2.LE.0) GO TO 340
               NP2=NPO(J2)
               IF (NP2.LE.0) GO TO 340
  320          NDIN2=NABA(NP2+1)
               IF (NDIN2.LE.0) GO TO 340
               IF (NDIN2.NE.NDOUT) GO TO 330
               IF (NABA(NP2).EQ.NDIN1) GO TO 450
  330          NP2=NP2+2
               GO TO 320
  340       CONTINUE
C
C   IF THE OUTPUT OF NODE N IS TO BE REPLACED BY THE OUTPUT OF ONE
C   OR SEVERAL NODES UPON ITS RELEASE THEN NDR(N) POINTS TO THE
C   LOCATION IN NRP(.) WHERE THE FIRST REPLACING NODE IS LOCATED.
C   IN OTHER WORDS N WILL BE REPLACED BY NODE NRP(NDR(N)). IF THERE
C   IS MORE THAN ONE REPLACEMENT THEN NRP(NDR)+2 POINTS TO THE NEXT
C   REPLACING NODE, OTHERWISE IT IS ZERO, ETC. IP1 IS A POINTER
C   INDICATING THE LOCATION OF THE FIRST AVAILABLE GROUP OF 3 CELLS
C   IN NRP(.). IP1+1 POINTS TO THE PREVIOUS AVAILABLE CELL AND
C   IP1+2 POINTS TO THE NEXT ONE. THE WORKING MECHANISM OF NRP(.)
C   IS THE SAME AS IN QSET/NSET. WHEN ENTRIES ARE INSERTED OR
C   DELETED FROM NRP(.) POINTERS ARE UPDATED ACCORDINGLY.
C
            IF (NDR(N).GT.0) GO TO 420
            NDR(N)=IP1
            NRP(IP1)=NDIN1
            NRP(NRP(IP1+1)+2)=NRP(IP1+2)
```

```
            NRP(NRP(IP1+2)+1)=NRP(IP1+1)
            NRP(IP1+1)=0
            IP1=NRP(IP1+2)
            NRP(NDR(N)+2)=0
C     CHECK IF IT IS NOT A REVERSE REPLACEMENT.
  350       JP=NDR(NDIN1)
  360       IF (JP.EQ.0) GO TO 450
            IF (NRP(JP).EQ.N) GO TO 370
            JP=NRP(JP+2)
            GO TO 360
  370       NRP(JP)=0
            IF (NRP(JP+2).GT.0) GO TO 390
            IF (NRP(JP+1).GT.0) GO TO 410
            NDR(NDIN1)=0
  380       NRP(JP+2)=IP1
            NRP(JP+1)=NRP(IP1+1)
            NRP(IP1+1)=JP
            IP1=JP
            GO TO 450
  390       IF (NRP(JP+1).GT.0) GO TO 400
            NDR(NDIN1)=NRP(JP+2)
            NRP(NDR(NDIN1)+1)=0
            GO TO 380
  400       NRP(NRP(JP+1)+2)=NRP(JP+2)
            NRP(NRP(JP+2)+1)=NRP(JP+1)
            GO TO 380
  410       NRP(NRP(JP+1)+2)=0
            GO TO 380
  420       IF (NRP(NDR(N)).EQ.NDIN1) GO TO 450
            IP2=NDR(N)+2
  430       IF (NRP(IP2).EQ.0) GO TO 440
            IF (NRP(NRP(IP2)).EQ.NDIN1) GO TO 450
            IP2=NRP(IP2)+2
            GO TO 430
  440       NRP(IP2)=IP1
            NRP(IP1)=NDIN1
            NRP(NRP(IP1+1)+2)=NRP(IP1+2)
            NRP(NRP(IP1+2)+1)=NRP(IP1+1)
            IP1=NRP(IP1+2)
            NRP(NRP(IP2)+1)=IP2-2
            NRP(NRP(IP2)+2)=0
            GO TO 350
C
  450       NP1=NP1+2
            GO TO 310
  460     CONTINUE
  470 CONTINUE
C
      DO 840 II=1,ISIM
C
C*****DELAY AND ACCUMULATOR NODE SECTION
C
        K=KST2(NEND(II))
        IF (K.LE.NSKD.OR.K.GT.NDTU) GO TO 480
        IF (XSTUS(K).GE.0) GO TO 480
```

```
      XSTUS(K)=TNOW
      IF (K.GE.NCNL.AND.K.LE.NCNU) GO TO 480
      CSTUS(K)=CSTUP(DUM)
  480 IF (NREL(NEND(II)).LT.0) GO TO 490
      IF (MREL(NEND(II)).GT.0) GO TO 840
      GO TO 510
  490 IF (MREL(NEND(II))) 840,510,510
C
C*****MARK NODE SECTION.
C
  500 ATRIB(2)=TNOW
      ATRIB(5)=CSTUP(DUM)
      NT=NTYPE(NEND(II))
      GO TO 560
  510 MREL(NEND(II))=NREL2(NEND(II))
      NREL(NEND(II))=NREL2(NEND(II))
C
C*****TEST TO SEE IF STATISTICS ARE TO BE COLLECTED ON THE NODE.
C*****NODE HAS BEEN RELEASED.
C
      IF (K.EQ.-1) GO TO 500
      IF (K.EQ.0) GO TO 530
      JSKA=IABS(KST3(K))
      IF (KST3(K)) 680,530,520
  520 IF (KST3(K).GT.1) GO TO 660
      GO TO 670
  530 NT=NTYPE(NEND(II))
      IF (NT.LE.4) GO TO 560
      KK=MST(NEND(II))
  540 INDX=MSTN(KK,1)
      IF (NSET(INDX).LT.0) NSET(INDX)=-NSET(INDX)
      KK=MSTN(KK,2)
      IF (KK.LT.77777) GO TO 540
      K=MFE(1)
      IF (K.EQ.0) GO TO 560
  550 INDX=K
      NTT=IABS(NSET(INDX))
      IF (NTT.EQ.NEND(II)) NSET(INDX)=NTT
      NTT=INDX+IM1+IM2+1
      K=NSET(NTT)
      IF (K.LT.77777) GO TO 550
  560 GO TO (610,610,570,570,610,610,570,570), NT
  570 ATRBB=ATRIB(2)
      CTRBB=ATRIB(5)
  580 KCOL=NFIND(NEND(II),1,1,1)
      IF (KCOL.LE.0) GO TO 600
      CALL RMOVE (KCOL,1)
C
C*****REMOVAL OF EVENTS, ADD COSTS ACCRUED TO DATE FOR C-NODE
C*****MONITORED COST ACTIVITIES.
C
      IF (JTRIB(6).LE.0) GO TO 590
      KQ=JTRIB(6)
      KK=KST2(KQ)
      COST=ATRIB(3)+ATRIB(4)*(TNOW-ATRIB(6))
```

```
      LCLT=MCLT+KK
      CALL COLCT (COST,LCLT)
      LCLT=NSTS+KK
      CALL HISTO (COST,LCLT)
      ATRIB(6)=0.0
      IF (ITFLG.EQ.3.OR.ICFLG.EQ.3) GO TO 860
  590 SUMCT=SUMCT-(ATRIB(1)-TNOW)*ATRIB(4)
      GO TO 580
  600 ATRIB(2)=ATRBB
      ATRIB(5)=CTRBB
  610 NODE=NEND(II)
      NNN=NDR(NODE)
      IF (NNN.NE.0) GO TO 630
      IF (NQ(NODE)) 850,840,620
  620 CALL SCHAT (NODE)
      GO TO 840
  630 NODE=NRP(NNN)
      IF (NDR(NODE).EQ.0) GO TO 640
      NNN=NDR(NODE)
      GO TO 630
  640 IF (NQ(NODE).LE.0) GO TO 650
      CALL SCHAT (NODE)
  650 NNN=NRP(NNN+2)
      IF (NNN.EQ.0) GO TO 840
      GO TO 630
  660 KST3(K)=-KST3(K)
      KST4(K)=KST4(K)+1
      GO TO 680
C
C*****REDUCE SINK NODE COUNT BY ONE. COLLECT STATISTICS.
C
  670 KST3(K)=0
  680 IF (K.GT.NSNK) GO TO 690
      NSTSR=NSTSR-1
  690 IF (JSKA.EQ.0) GO TO 720
      GO TO (720,720,730,740,700,700), JSKA
C
C*****DELAY AND ACCUMULATOR NODE STATISTICS
C
  700 YY=TNOW-XSTUS(K)
      XSTUS(K)=-1.0
      IF (K.LT.NCNL) GO TO 710
      IF (K.GT.NCNU) GO TO 710
      COST=CSTUS(K)
      CSTUS(K)=0.0
      GO TO 750
  710 CCC=CSTUP(DUM)
      COST=CCC-CSTUS(K)
C
      GO TO 750
C*****FIRST OR ALL STATISTICS
C
  720 YY=TNOW
      COST=CSTUP(DUM)
      GO TO 750
```

```
C
C*****BETWEEN STATISTICS
C
  730     YY=TNOW-XSTUS(K)
          CCC=CSTUP(DUM)
          COST=CCC-CSTUS(K)
          CSTUS(K)=CCC
          XSTUS(K)=TNOW
          IF (YY-TNOW) 750,750,830
C
C*****INCREMENT STATISTICS
C
  740     YY=TNOW-ATRIB(2)
          COST=CSTUP(DUM)-ATRIB(5)
  750     CALL HISTO (YY,K)
          NL=K+(K-1)*NCTS
          CALL COLCT (YY,NL)
          IF (K.LT.NCNL) GO TO 760
          IF (K.LE.NCNU) GO TO 770
  760     LCLT=MCLT+K
          CALL COLCT (COST,LCLT)
          LCLT=NSTS+K
          CALL HISTO (COST,LCLT)
  770     IF (NCTS) 830,830,780
C
C*****COUNTER STATISTICS
C
  780     LINE=MFE(1)
          IF (LINE.LE.0) GO TO 810
  790     INDX=LINE+IM1
          IF (TNOW.NE.QSET(INDX)) GO TO 810
          INDX=LINE+3
          IF (NSET(INDX).LE.0) GO TO 800
          L=NSET(INDX)
          KOUNT(L)=KOUNT(L)+1
          NSET(INDX)=0
  800     JLINE=LINE+IM1+IM2+1
          LINE=NSET(JLINE)
          IF (LINE.LT.77777) GO TO 790
  810     DO 820 I=1,NCTS
             NL=NL+1
             XC=KOUNT(I)
             CALL COLCT (XC,NL)
  820     CONTINUE
  830     IF (ITFLG.EQ.3.OR.ICFLG.EQ.3) GO TO 860
C
C*****TEST TO SEE IF SIMULATION RUN IS COMPLETE.
C
          IF (NSTSR) 850,860,530
  840 CONTINUE
      GO TO 140
  850 CALL ERROR (22)
C
C*****NETWORK HAS BEEN SIMULATED ONE MORE TIME.
C
```

```
C
C*****IF ANY C-NODES, COLLECT RUN STATS AND CONTINUE. OTHERWISE SKIP
C*****THIS SECTION.
C
  860 IF (NCND.LE.0) GO TO 910
      XTIME=0.0
      XCOST=0.0
      DO 870 KXK=NCNL,NCNU
         MXM=KXK+(KXK-1)*NCTS
         LXL=MCLT+KXK
         XTIME=XTIME+SUM1(MXM)
         XCOST=XCOST+SUM1(LXL)
C
C*****IF ANY C-NODE IS STILL ACTIVE, UPDATE STATS TO FINAL VALUES.
C
         IF (XSTUS(KXK).LT.0.) GO TO 870
         TYYY=TYYY+(TNOW-XSTUS(KXK))
  870 CONTINUE
C
C*****XTIME--ACCUMULATED TIME IN C-NODES TO DATE.
C*****XCOST--ACCUMULATED COST IN C-NODES TO DATE.
C*****TIMRN--TIME IN C-NODES FOR THIS RUN.
C*****COSRN--COST IN C-NODES FOR THIS RUN.
C*****TIMTD--OLD TIME-TO-DATE IN C-NODES.
C*****COSTD--OLD COST-TO-DATE IN C-NODES.
C*****TYYY--ACCUMULATIVE TIMES OF STILL ACTIVE C-NODES AT END OF RUNS
C*****TCCC--ACCUMULATIVE COSTS OF STILL ACTIVE C-NODES AT END OF RUNS
C*****UPDATE COSTS FOR ANY COST ARC MONITORED BY A C-NODE.
C
      LINE=MFE(1)
      IF (LINE.LE.0) GO TO 900
  880 INDQX=LINE+IM1
      IF (QSET(INDQX+5).LE.0.) GO TO 890
      TCCC=TCCC+QSET(INDQX+2)+QSET(INDQX+3)*(TNOW-QSET(INDQX+5))
  890 ISUB=LINE+IM1+IM2+1
      LINE=NSET(ISUB)
      IF (LINE.LT.77777) GO TO 880
  900 XTIME=XTIME+TYYY
      XCOST=XCOST+TCCC
C
C*****CALCULATE TIME AND COST OF C-NODES OF THIS RUN AND THEN
C*****UPDATE TIME AND COST TO DATE.
C
      TIMRN=XTIME-TIMTD
      COSRN=XCOST-COSTD
      TIMTD=XTIME
      COSTD=XCOST
C
C*****UPDATE C-NODE RUN TIME AND COST STAT VARIABLES.
C
      KXKK=NCLT-1
      CALL COLCT (TIMRN,KXKK)
      NHIST=NSTS*2+2
      MXKK=NHIST-1
      CALL HISTO (TIMRN,MXKK)
```

```
      CALL COLCT (COSRN,NCLT)
      CALL HISTO (COSRN,NHIST)
  910 DO 940 K=1,NSTS
         IF (KST3(K)) 920,930,940
  920    KST3(K)=-KST3(K)
         GO TO 940
  930    KST3(K)=1
  940 CONTINUE
      IF (NDRT.EQ.0) GO TO 960
      INDX=1
      DO 950 K=1,ID
         NSET(INDX)=IABS(NSET(INDX))
         INDX=INDX+IM1+IM2+1+1
  950 CONTINUE
  960 DO 970 M=1,NSTS
         XSTUS(M)=-1.
  970 CONTINUE
C
C*****INITIALIZE FOR ANOTHER RUN BY REMOVING ALL EVENTS FROM EVENT
C*****FILE AND RESETTING NETWORK VALUES.
C
  980 IF (NQ(1)) 850,1000,990
  990 KCOL=MFE(1)
      CALL RMOVE (KCOL,1)
      GO TO 980
C
C*****TEST TO SEE IF ALL RUNS HAVE BEEN MADE.
C
 1000 IF (NRNS.LE.NRUN) GO TO 1010
      NRUN=NRUN+1
      GO TO 50
C
 1010 CALL SUMRY
      RETURN
C
C
C
 1020 FORMAT (///40X,'** EVENT TRACE FOR RUN',I5,' **',///,
     1  ' END OF ''ARC',4X,'ARC',4X,'END',3X,'PARAM.',2X,'DISTR.',2X,
     2  'COUNT',8X,'RELEASE',8X,'ARC',12X,'ARC',/,4X,'TIME',6X,'NUMBER',
     3  2X,'NODE',3X,'TYPE',4X,'TYPE',3X,'TYPE = (    )',2X,'COUNT',9X,
     4  'COST',9X,'DURATION',/)
 1030 FORMAT (//,' NODE ',I3,' MAY BE TRAPPED IN AN INFINITE LOOP.',/,
     1  ' THERE ARE ',I3,' SIMULTANEOUS ARC COMPLETIONS AT TIME ',
     2  F10.2)
 1040 FORMAT (1X,E12.6,1X,I4,2I7,2I8,I7,I8,2(4X,E12.6))
      END
      SUBROUTINE FILEM (JQ)
      DIMENSION NSET(5600)
      COMMON QSET(5600)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      COMMON /G3/ KST1(100),XLOW(200),NREL1(300),NREL2(300),NREL(300),
```

```
     1  MREL(300),KST4(100),KST3(400),NTYPE(300),WIDTH(200),NSTS,NCTS
      EQUIVALENCE (NSET(1),QSET(1))
      DOUBLE PRECISION SUM1,SUM2
C
C*****TEST TO SEE IF THERE IS AN AVAILABLE COLUMN FOR STORAGE
C
      IF (MFA.LE.MAXQS) GO TO 10
      WRITE (IPRT,170)
C
      CALL ERROR (87)
C
C*****PUT ATTRIBUTE VALUES IN FILE
C
   10 INDX=MFA
      DO 20 I=1,IM1
         NSET(INDX)=JTRIB(I)
         INDX=INDX+1
   20 CONTINUE
      DO 30 I=1,IM2
         QSET(INDX)=ATRIB(I)
         INDX=INDX+1
   30 CONTINUE
      MFEX=MFE(JQ)
      KNT=2
      INDX=MFA+IM1+IM2+1
      NXFA=NSET(INDX)
      MLEX=MLE(JQ)
      IF (MLEX) 40,50,90
   40 CALL ERROR (88)
   50 INDX=MFA+IM1+IM2
      NSET(INDX)=99999
      MFE(JQ)=MFA
   60 INDX=INDX+1
      NSET(INDX)=77777
      MLE(JQ)=MFA
   70 MFA=NXFA
      IF (MFA.GE.88888) GO TO 80
      INDX=MFA+IM1+IM2
      NSET(INDX)=99999
   80 NQ(JQ)=NQ(JQ)+1
      RETURN
   90 INDX1=MFA+IM1
      INDX2=MLEX+IM1
      IF (JQ.GT.1) GO TO 100
      IF (QSET(INDX1)-QSET(INDX2)) 160,120,110
  100 IF (QSET(INDX1).GT.QSET(INDX2)) GO TO 160
  110 INDX=MLEX+IM1+IM2+1
      MSU=NSET(INDX)
      NSET(INDX)=MFA
      INDX=MFA+IM1+IM2
      NSET(INDX)=MLEX
      IF (KNT.EQ.2) GO TO 60
      INDX=MFA+IM1+IM2+1
      NSET(INDX)=MSU
      INDX=MSU+IM1+IM2
```

```
      NSET(INDX)=MFA
      GO TO 70
  120 IF (JTRIB(5).LT.0) GO TO 130
      IF (MREL(JTRIB(1)).NE.1) GO TO 140
      ATRIB(1)=ATRIB(1)+EPS
      QSET(INDX1)=ATRIB(1)
      GO TO 110
  130 IF (NREL(JTRIB(1)).GE.0.OR.(MREL(JTRIB(1))-JTRIB(5)).GE.0) GO TO 1
     140
      ATRIB(1)=ATRIB(1)-EPS
      QSET(INDX1)=ATRIB(1)
      GO TO 160
  140 IF (NSET(INDX1-2).LT.0) GO TO 150
      IF (MREL(NSET(INDX1-6)).NE.1) GO TO 160
      QSET(INDX2)=QSET(INDX2)+EPS
      GO TO 160
  150 IF (NREL(NSET(INDX1-6)).GE.0.OR.(MREL(NSET(INDX1-6))-NSET(INDX1-
     1  2)).GE.0) GO TO 160
      QSET(INDX2)=QSET(INDX2)-EPS
      GO TO 110
  160 KNT=1
      INDX=MLEX+IM1+IM2
      MLEX=NSET(INDX)
      IF (MLEX.NE.99999) GO TO 90
      INDX=MFA+IM1+IM2
      NSET(INDX)=99999
      MFE(JQ)=MFA
      INDX=MFA+IM1+IM2+1
      NSET(INDX)=MFEX
      INDX=MFEX+IM1+IM2
      NSET(INDX)=MFA
      GO TO 70
C
C
  170 FORMAT (//24H OVERLAP SET GIVEN BELOW/)
      END
C
      SUBROUTINE SCHAT (NODE)
      DIMENSION NSET(5600)
      COMMON QSET(5600)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOO,NSNK,
     1  NPUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      COMMON /G3/ KST1(100),XLOW(200),NPEL1(300),NREL2(300),NREL(300),
     1  MREL(300),KST4(100),KST3(100),NTYPE(300),WIDTH(200),NSTS,NCTS
      COMMON /G4/ XSTUS(100),SUMCT,CSTUS(100),NCND,NCNL,NCNU,T2,C2,
     1  ITFLG,ICFLG,TYYY,TCCC,TIMTD,COSTD
      EQUIVALENCE (NSET(1),QSET(1))
      DOUBLE PRECISION SUM1,SUM2
C
C*****NEXT IS LOCATION OF FIRST ENTRY IN FILE OF ACTIVITIES WITH START
C*****NODE- NODE. NT IS THE NODE TYPE.
C
      III=0
```

```
      NT=NTYPE(NODE)
      NEXT=MFE(NODE)
      IF (NEXT.LE.0) GO TO 80
      GO TO (20,10,20,10,20,10,20,10), NT
C
C*****GENERATE A RANDOM NUMBER RNUM.
C
   10 RNUM=RANF(ISED)
C
C*****TEST IF DETERMINISTIC OR PROBABILISTIC NODE.
C
   20 GO TO (30,90,30,90,30,90,30,90), NT
   30 DO 40 I=1,IM1
         INDX=NEXT+I-1
         JTRIB(I)=NSET(INDX)
   40 CONTINUE
C
C*****OBTAIN SAMPLE FOR ARC.
C
      CALL SAMPL (DEV)
      ATRIB(1)=TNOW+DEV
      INDXQ=NEXT+IM1+2
      ATRIB(3)=QSET(INDXQ)
      ATRIB(4)=QSET(INDXQ+1)
      SUMCT=SUMCT+ATRIB(3)+DEV*ATRIB(4)
      IF (JTRIB(6).GT.0) ATRIB(6)=TNOW
C
C*****FILE END OF ARC EVENT IN EVENT FILE.
C
      CALL FILEM (1)
      IF (JTRIB(1).LE.0) GO TO 50
      NSUB=JTRIB(1)
      NTI=NTYPE(NSUB)
      IF (NTI.LE.4) GO TO 50
      NSET(NEXT)=-NSET(NEXT)
   50 III=1
      GO TO (60,80,60,80,60,80,60,80), NT
C
C*****DETERMINE IF OTHER ACTIVITIES ARE IN FILE.
C
   60 INDX=NEXT+IM1+IM2+1
      NEXT=NSET(INDX)
      IF (NEXT-77777) 20,80,70
   70 CALL ERROR (23)
   80 IF (III.NE.0) GO TO 100
      WRITE (IPRT,110) TNOW,NODE
      IF (NEXT.EQ.0) CALL EXIT
      IF (NEXT.NE.0) GO TO 30
      CALL EXIT
   90 INDX=NEXT+IM1
C
C*****TEST RNUM AGAINST PROBABILITY (CUM.) OF REALIZING THE ARC.
C
      IF (QSET(INDX).GT.RNUM) GO TO 30
      GO TO 60
```

```
  100 CONTINUE
      RETURN
C
C
C
  110 FORMAT (/12X,20HSCHAT CALLED AT TIME,F8.2,42H   ARC    COULD NOT B
     1E SCHEDULED FROM NODE,I5/)
      END
      SUBROUTINE RMOVE (KCOL,JQ)
      DIMENSION NSET(5600)
      COMMON QSET(5600)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      EQUIVALENCE (NSET(1),QSET(1))
      DOUBLE PRECISION SUM1,SUM2
      IF (KCOL.GT.0) GO TO 10
      IF ((MAXQS-MFA).LT.MAXQS) CALL ERROR (11)
      CALL ERROR (97)
C
C*****PUT VALUES OF KCOL IN ATTRIB
C
   10 INDX=KCOL
      DO 20 I=1,IM1
         JTRIB(I)=NSET(INDX)
         INDX=INDX+1
   20 CONTINUE
      DO 30 I=1,IM2
         ATRIB(I)=QSET(INDX)
         INDX=INDX+1
   30 CONTINUE
      INDX=KCOL+IM1+IM2
      JL=NSET(INDX+1)
      JK=NSET(INDX)
      IF (JL.EQ.77777) GO TO 70
      IF (JK.EQ.99999) GO TO 60
      INDX=JK+IM1+IM2+1
      NSET(INDX)=JL
      INDX=JL+IM1+IM2
      NSET(INDX)=JK
   40 INDX=KCOL+IM1+IM2+1
      NSET(INDX)=MFA
      NSET(INDX-1)=99999
      IF (MFA.GE.88888) GO TO 50
      INDX=MFA+IM1+IM2
      NSET(INDX)=KCOL
   50 MFA=KCOL
      NQ(JQ)=NQ(JQ)-1
      RETURN
   60 INDX=JL+IM1+IM2
      NSET(INDX)=99999
      MFE(JQ)=JL
      GO TO 40
   70 IF (JK.EQ.99999) GO TO 80
```

```
      INDX=JK+IM1+IM2+1
      NSET(INDX)=77777
      MLE(JQ)=JK
      GO TO 40
   80 MFE(JQ)=0
      MLE(JQ)=0
      GO TO 40
      END
C
      FUNCTION NFIND (NVAL,JQ,JATT,IRNK)
      DIMENSION NSET(5600)
      COMMON QSET(5600)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      EQUIVALENCE (NSET(1),QSET(1))
      DOUBLE PRECISION SUM1,SUM2
      NEXTK=MFE(JQ)
      K=1
      IF (NEXTK) 10,20,30
   10 CALL ERROR (89)
   20 NFIND=0
      RETURN
   30 INDX=NEXTK+JATT-1
      IF (NSET(INDX).EQ.NVAL) GO TO 40
      INDS=NEXTK+IM1+IM2+1
      NEXTK=NSET(INDS)
      IF (NEXTK.LT.77777) GO TO 30
      GO TO 20
   40 NFIND=NEXTK
      IF (K.EQ.IRNK) RETURN
      K=K+1
      GO TO 30
      END
C
      SUBROUTINE HISTC (X1,N)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G3/ KST1(100),XLOW(200),NREL1(300),NREL2(300),NREL(300),
     1  MREL(300),KST4(100),KST3(100),NTYPE(300),WIDTH(200),NSTS,NCTS
      COMMON /G6/ NCEL(10),PROB(32,10),VAL(33,10),ISEC(3,10),MAX,NHIS,
     1  KHIST,NPNCH
      NHIST=NSTS*2+2
      IF (N.LE.NHIST) GO TO 20
   10 WRITE (IPRT,90) N
      CALL EXIT
   20 IF (N) 10,10,30
C
C*****TRANSLATE X1 BY SUBTRACTING A IF X.LE.A THEN ADD 1 TO FIRST CELL
C
   30 A=XLOW(N)
      W=WIDTH(N)
      IF (W) 80,70,40
   40 X=X1-A
```

```
      IF (X.GE.0.) GO TO 50
      IC=1
      GO TO 60
C
C*****DETERMINE CELL NUMBER IC. ADD 1 FOR LOWER LIMIT CELL AND 1 FOR
C*****TRUNCATION
C
   50 IC=X/W+2.+.0001
C
      IF (IC.LE.MXC) GO TO 60
      IC=MXC
   60 JCELS(N,IC)=JCELS(N,IC)+1
   70 RETURN
   80 WRITE (NHIS) X1,N
      KHIST=KHIST+1
      RETURN
C
C
   90 FORMAT (19H ERROR IN HISTOGRAM,I4//)
      END
C
      SUBROUTINE COLCT (X,N)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      COMMON /G3/ KST1(100),XLOW(200),NREL1(300),NREL2(300),NREL(300),
     1  MREL(300),KST4(100),KST3(100),NTYPE(300),WIDTH(200),NSTS,NCTS
      COMMON /G4/ XSTUS(100),SUMCT,CSTUS(100),NCND,NCNL,NCNU,T2,C2,
     1  ITFLG,ICFLG,TYYY,TCCC,TIMTD,COSTD
      DOUBLE PRECISION SUM1,SUM2
      NCLT=NSTS*(NCTS+2)+2
      IF (N.GT.0) GO TO 20
   10 CALL ERROR (90)
   20 IF (N.GT.NCLT) GO TO 10
      SUM1(N)=SUM1(N)+X
      SUM2(N)=SUM2(N)+X*X
      SUM3(N)=SUM3(N)+1.0
      IF (X.LT.SUM4(N)) SUM4(N)=X
      IF (X.GT.SUM5(N)) SUM5(N)=X
      IF (T2.GE.0.) GO TO 30
      IF (C2.LT.0.) GO TO 40
   30 CALL COLCC
   40 RETURN
      END
C
      SUBROUTINE COLCC
      DIMENSION NSET(5600)
      COMMON QSET(5600)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      COMMON /G3/ KST1(100),XLOW(200),NREL1(300),NREL2(300),NREL(300),
     1  MREL(300),KST4(100),KST3(100),NTYPE(300),WIDTH(200),NSTS,NCTS
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      COMMON /G4/ XSTUS(100),SUMCT,CSTUS(100),NCND,NCNL,NCNU,T2,C2,
     1  ITFLG,ICFLG,TYYY,TCCC,TIMTD,COSTD
      EQUIVALENCE (NSET(1),QSET(1))
      DOUBLE PRECISION SUM1,SUM2
      TESTT=0.0
      TESTC=0.0
      IF (T2.LT.0.) GO TO 30
      KFLAG=ITFLG+1
      GO TO (10,30,10,30), KFLAG
C
C*****CALCULATE ACCUMULATED TIMES.
C
   10 DO 20 K=NCNL,NCNU
         KK=K+(K-1)*NCTS
         TESTT=TESTT+SUM1(KK)
         IF (XSTUS(K).LE.0.) GO TO 20
         YYY=TNOW-XSTUS(K)
         TESTT=TESTT+YYY
   20 CONTINUE
      TESTT=TESTT+TYYY-TIMTD
C
C*****TEST ACCUMULATED TIME AGAINST THE T2 VALUE.
C
      IF (TESTT.LT.T2) GO TO 30
      ITFLG=ITFLG+1
      NT2R=NT2R+1
      IF (ICFLG.EQ.1.OR.ICFLG.EQ.3) NT2C2=NT2C2+1
   30 IF (C2.LT.0.) GO TO 90
      KFLAG=ICFLG+1
      GO TO (40,90,40,90), KFLAG
C
C*****CALCULATE PRESENT COST IN C-NODES.
C
   40 DO 50 K=NCNL,NCNU
         L=NCLT-NSTS-2+K
         TESTC=TESTC+SUM1(L)
   50 CONTINUE
C
C*****CALCULATE AND ADD COSTS FROM STILL ACTIVE COST ACTIVITIES.
C
C*****ATRIB(3)=QSET(INDQX+2)      SETUP COST OF ARC
C*****ATRIB(4)=QSET(INDQX+3)      VARIABLE COST OF ARC
C*****ATRIB(6)=QSET(INDQX+5)      TIME ARC WAS STARTED
C
      LINE=MFE(1)
      IF (LINE.LE.0) GO TO 80
   60 INDQX=LINE+IM1
      IF (QSET(INDQX+5).LE.0.) GO TO 70
      TESTC=TESTC+QSET(INDQX+2)+QSET(INDQX+3)*(TNOW-QSET(INDQX+5))
   70 ISUB=LINE+IM1+IM2+1
      LINE=NSET(ISUB)
      IF (LINE-77777) 60,80,80
C
C*****TEST ACCUMULATED COSTS AGAINST THE C2 VALUE.
C
```

```
 80 TESTC=TESTC-COSTD+TCCC
    IF (TESTC.LT.C2) GO TO 90
    ICFLG=ICFLG+1
    NC2R=NC2R+1
    IF (ITFLG.EQ.1.OR.ITFLG.EQ.3) NT2C2=NT2C2+1
 90 CONTINUE
    RETURN
C
    END
    FUNCTION CSTUP (DUM)
    DIMENSION NSET(5600)
    COMMON QSET(5600)
    COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
   1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
    COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
   1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
    COMMON /G4/ XSTUS(100),SUMCT,CSTUS(100),NCND,NCNL,NCNU,T2,C2,
   1  ITFLG,ICFLG,TYYY,TCCC,TIMTD,COSTD
    EQUIVALENCE (NSET(1),QSET(1))
    DOUBLE PRECISION SUM1,SUM2
    LINE=MFE(1)
    CSTUP=SUMCT
    IF (LINE.LE.0) GO TO 20
 10 INDX=LINE+IM1
    CSTUP=CSTUP+(TNOW-QSET(INDX))*QSET(INDX+3)
    ISUB=LINE+IM1+IM2+1
    LINE=NSET(ISUB)
    IF (LINE.LT.77777) GO TO 10
 20 CONTINUE
    RETURN
C
    END
    SUBROUTINE SUMRY
    DIMENSION GRAF(50), XLET(6)
    COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
   1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
    COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
   1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
    COMMON /G3/ KST1(100),XLOW(200),NREL1(300),NREL2(300),NREL(300),
   1  MREL(300),KST4(100),KST3(100),NTYPE(300),WIDTH(200),NSTS,NCTS
    COMMON /G4/ XSTUS(100),SUMCT,CSTUS(100),NCND,NCNL,NCNU,T2,C2,
   1  ITFLG,ICFLG,TYYY,TCCC,TIMTD,COSTD
    COMMON /G5/ MSTN(2000,2),MST(300),KST2(300),IGRF,JGRAF,SCAL
    DATA AA,BB,CC/1H*,1H ,1HC/
    DATA XLET/1HF,1HA,1HB,1HI,1HD,1HC/,T,CT/4HTIME,4HCOST/
    DOUBLE PRECISION SUM1,SUM2
    SFLAG=0.0
    NHIST=NSTS*2+2
    HFLAG=0.0
    NCLT=NSTS*(NCTS+2)+2
    MCLT=NCLT-NSTS-2
    CALL ADJST
    WRITE (IPRT,340)
    WRITE (IPRT,350) NAME,NRUN
    K=1
```

```
   10 JJ=KST3(K)
      J=1
      IF (JJ.EQ.0) GO TO 110
      CODE=XLET(JJ)
   20 I=(K-1)*(NCTS+1)+J
      JJ=J-1
   30 IF (SUM3(I).GT.0.) GO TO 40
      IF (JJ.NE.0) GO TO 100
      IF (I.GT.MCLT) GO TO 100
      WRITE (IPRT,360) KST1(K),JJ
      GO TO 100
   40 XS=SUM1(I)
      XSS=SUM2(I)
      XN=SUM3(I)
      AVG=XS/XN
      IF (XN*XSS.GT.XS*XS) GO TO 50
      STD=0.0
      GO TO 60
   50 STD=SQRT(((XN*XSS)-(XS*XS))/(XN*(XN-1.0)))
C
C*****TEST IF C-NODE RUN STATS ARE BEING PROCESSED.
C
   60 IF (SFLAG.EQ.1.0) GO TO 280
      IF (HFLAG.EQ.1.0) GO TO 290
      IF (J.GT.1) GO TO 90
      IF (KST3(K).LE.1) GO TO 70
      XN=KST4(K)
   70 PROB=XN/NRUN
      IF (I.GT.MCLT) GO TO 80
      WRITE (IPPT,370) KST1(K),PROB,AVG,STD,SUM3(I),SUM4(I),SUMS(I),CODE
      I=MCLT+K
      GO TO 30
   80 IF (JGRAF.EQ.0) GO TO 100
      WRITE (IPRT,380) AVG,STD,SUM3(I),SUM4(I),SUM5(I)
      GO TO 100
   90 WRITE (IPRT,390) KST1(K),JJ,AVG,STD,SUM3(I),SUM4(I),SUM5(I)
  100 J=J+1
      IF (J.LE.(NCTS+1)) GO TO 20
  110 K=K+1
      IF (K.LE.NSTS) GO TO 10
      WRITE (IPRT,400)
      IF (T2.GE.0.) WRITE (IPRT,410) NT2R
      IF (C2.GE.0.0) WRITE (IPRT,420) NC2R
      IF (NT2C2.GT.0) WRITE (IPRT,430) NT2C2
      IF (IGRF.EQ.1) GO TO 150
      WRITE (IPRT,440)
      DO 130 I=1,NSTS
         NCL=MXC
         IF (WIDTH(I).EQ.0.) GO TO 120
         WID=ABS(WIDTH(I))
         WRITE (IPRT,450) KST1(I),XLOW(I),WID,(JCELS(I,J),J=1,NCL)
  120    IF (JGRAF.EQ.0) GO TO 130
         J=NSTS+I
         IF (WIDTH(J).EQ.0.) GO TO 130
         WID=ABS(WIDTH(J))
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      WRITE (IPRT,460) KST1(I),XLOW(J),WID,(JCELS(J,K),K=1,NCL)
  130 CONTINUE
      IF (NCND.GT.0) GO TO 270
  140 CONTINUE
      RETURN
  150 NNN=0
  160 KKK=1
  170 I=KKK
      II=I+NNN
      IF (WIDTH(II).EQ.0.) GO TO 250
      INDEX=II+(II-1)*NCTS
      IF (NNN.GT.0) INDEX=MCLT+I
      IF (SUM3(INDEX).LE.0.) GO TO 250
      IF (NNN.GT.0) GO TO 180
      WRITE (IPRT,470) KST1(I)
      GO TO 190
  180 WRITE (IPRT,480) KST1(I)
  190 WRITE (IPRT,490)
      WRITE (IPRT,500)
      CUML=0.
      WID=ABS(WIDTH(II))
      DO 240 J=1,MXC
         RELA=JCELS(II,J)
         RELA=RELA/SUM3(INDEX)
         CUML=CUML+RELA
         MR=RELA*50.+.000001
         MC=CUML*50.+.000001
         DO 200 N=1,50
            GRAF(N)=BB
  200    CONTINUE
         GRAF(50)=XLET(4)
         IF (MC.LE.0) GO TO 220
         GRAF(MC)=CC
         IF (MR.LE.0) GO TO 220
         DO 210 N=1,MR
            GRAF(N)=AA
  210    CONTINUE
  220    IF ((J-1).GT.0) GO TO 230
         WRITE (IPRT,510) JCELS(II,J),RELA,CUML,GRAF
         BLOW=XLOW(II)-WID
         GO TO 240
  230    BLOW=BLOW+WID
         WRITE (IPRT,520) JCELS(II,J),RELA,CUML,BLOW,GRAF
  240 CONTINUE
      KOBS=SUM3(INDEX)+.001
      WRITE (IPRT,530) KOBS
C
C*****TEST IF C-NODE HISTGRAMS BEING PROCESSED AND IF DONE.
C
      IF (HFLAG.EQ.1.0) GO TO 330
      IF (SFLAG.EQ.1.0) GO TO 140
  250 KKK=KKK+1
      IF (KKK.LE.NSTS) GO TO 170
      IF (NNN.GT.0) GO TO 260
      IF (JGRAF.LE.0) GO TO 260
```

```
      NNN=NSTS
      GO TO 160
  260 IF (NCND) 140,140,270
C
C*****WORKING ON C-NODE STATS. SET FIRST FLAG AND PRINT OUT HEADINGS.
C
  270 SFLAG=1.0
      WRITE (IPRT,540) NRUN
C
C*****PREPARE TO PROCESS C-NODE TIME DATA.
C
      I=NCLT-1
      CODE=T
      GO TO 40
C
C*****PRINT OUT C-NODE RUN TIME STATS.
C
  280 WRITE (IPRT,550)
      WRITE (IPRT,560) CODE,AVG,STD,SUM3(I),SUM4(I),SUM5(I)
C
C*****PREPARE TO PROCESS C-NODE COST DATA.
C
      HFLAG=1.0
      SFLAG=0.0
      I=NCLT
      CODE=CT
      GO TO 40
  290 WRITE (IPRT,560) CODE,AVG,STD,SUM3(I),SUM4(I),SUM5(I)
C
C*****PREPARE TO GET HISTOGRAM OF C- NODE DATA.
C
      IF (IGRF.EQ.1) GO TO [illegible]
C
C*****STANDARD HISTOGRAMS FOLLOW. FIRST TIME DATA.
C
      I=NHIST-1
      NCL=MXC
      IF (WIDTH(I).NE.0..OR.WIDTH(NHIST).NE.0.) WRITE (IPRT,440)
      IF (WIDTH(I).EQ.0.) GO TO 300
      WID=ABS(WIDTH(I))
      WRITE (IPRT,570) XLOW(I),WID,(JCELS(I,J),J=1,NCL)
C
C*****COST HISTOGRAMS.
C
  300 IF (WIDTH(NHIST).EQ.0.) GO TO 140
      WID=ABS(WIDTH(NHIST))
      WRITE (IPRT,580) XLOW(NHIST),WID,(JCELS(NHIST,K),K=1,NCL)
      WRITE (IPRT,340)
      GO TO 140
C
C*****PLOTTED HISTOGRAMS.
C
  310 II=NHIST-1
      INDEX=NCLT-1
      IF (WIDTH(II).EQ.0.) GO TO 330
```

```
      WRITE (IPRT,590)
C
C*****PREPARE TO PROCESS TIME DATA.
C
  320 I=II
      GO TO 190
C
C*****PREPARE TO PROCESS COST DATA.
C
  330 IF (WIDTH(NHIST).EQ.0.) GO TO 140
      WRITE (IPRT,600)
      SFLAG=1.0
      HFLAG=0.0
      II=NHIST
      INDEX=NCLT
      GO TO 320
C
C
C
  340 FORMAT (1H1)
  350 FORMAT (1X,26HGRASP SIMULATION PROJECT :,1X,20A4,////,31X,
     1  '**FINAL RESULTS FOR',I8,' SIMULATION RUN(S)**',//,16X,'NODE',
     2  4X,'PROB./',8X,'MEAN',9X,'STAND.',7X,'NO OF',6X,'MIN',12X,'MAX',
     3  9X,'NODE',/,24X,'COUNT',23X,'DEV.',9X,'OBS.',33X,'TYPE')
  360 FORMAT (/1X,I17,I13,9X,18HNO VALUES RECORDED)
  370 FORMAT (/1X,I17,3X,F9.4,4X,E12.6,3X,E12.6,F8.0,2(3X,E12.6),4X,A3)
  380 FORMAT (17X,10HCOST      ,7X,E12.6,3X,E12.6,F8.0,2(3X,E12.6))
  390 FORMAT (1X,I17,I13,3X,E12.6,3X,E12.6,F8.0,2(3X,E12.6))
  400 FORMAT (1H0)
  410 FORMAT (1H0,16X,47HNUMBER OF TIMES SYSTEM EXCEEDED T2 CRITERION IS
     1,I6)
  420 FORMAT (1H0,16X,47HNUMBER OF TIMES SYSTEM EXCEEDED C2 CRITERION IS
     1,I6)
  430 FORMAT (1H0,16X,58HNUMBER OF TIMES SYSTEM EXCEEDED BOTH T2 AND C2
     1CRITERIA IS,I6)
  440 FORMAT (//43X,14H**HISTOGRAMS**//17X,5HLOWER,7X,4HCELL/6X,4HNODE,
     1  7X,5HLIMIT,7X,5HWIDTH,31X,11HFREQUENCIES)
  450 FORMAT (/1X,I9,2E12.4,4X,11I6/(38X,11I6))
  460 FORMAT (/1X,4HCOST,I5,2E12.4,4X,11I6/(38X,11I6))
  470 FORMAT (1H1///20X,23HSTAT HISTOGRAM FOR NODE,I5)
  480 FORMAT (1H1///20X,23HCOST HISTOGRAM FOR NODE,I5)
  490 FORMAT (///20X,4HOBSV,5X,4HRELA,5X,4HCUML,5X,11HLOWER BOUND,8X,
     1  1H0,8X,2H20,8X,2H40,8X,2H60,8X,2H80,7X,3H100)
  500 FORMAT (20X,3(4HFREQ,5X),2X,7HOF CELL,8X,1HI,10(5H....I))
  510 FORMAT (/19X,I5,2(4X,F5.3),9X,4H-INF,9X,1HI,50A1)
  520 FORMAT (19X,I5,2(4X,F5.3),4X,E12.4,6X,1HI,50A1)
  530 FORMAT (21X,3H---,40X,1HI,10(5H....I)/18X,I6)
  540 FORMAT (1H1///24X,51H**FINAL RESULTS FOR RUN DATA OF ACCUMULATOR N
     1ODES**,////,31X,'**FINAL RESULTS FOR',I8,' SIMULATION RUN(S)**'/)
  550 FORMAT (//17X,4HTYPE,18X,4HMEAN,5X,8HSTD.DEV.,5X,5HNO OF,9X,
     1  4HMIN.,9X,4HMAX./62X,4HOBS.//)
  560 FORMAT (17X,A4,11X,2E12.4,3X,F7.0,2E13.4)
  570 FORMAT (//12X,4HTIME,2E12.4,4X,11I6/(44X,11I6))
  580 FORMAT (//12X,4HCOST,2E12.4,4X,11I6/(44X,11I6))
  590 FORMAT (1H1///20X,42HSTAT HISTOGRAM FOR ACCUMULATION NODE TIMES)
```

```
  600 FORMAT (1H1///20X,42HSTAT HISTOGRAM FOR ACCUMULATION NODE COSTS)
      END
      SUBROUTINE SAMPL (DEV)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NDQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      COMMON /G5/ MSTN(2000,2),MST(300),KST2(300),IGRF,JGRAF,SCAL
      DOUBLE PRECISION SUM1,SUM2
      JP=JTRIB(2)
      JD=JTRIB(3)
      GO TO (10,20,30,40,110,120,170,180,170,190,210,220,230), JD
C
C     CONSTANT
C
   10 DEV=PARM(JP,1)
      RETURN
C
C     NORMAL
C
   20 DEV=RNORM(JP)
      RETURN
C
C     UNIFORM
C
   30 A=PARM(JP,2)
      B=PARM(JP,3)
      DEV=A+(B-A)*RANF(ISED)
      RETURN
C
C     ERLANG
C
   40 K=PARM(JP,4)+.001
      RNUM=RANF(ISED)
      K=K-1
      IF (K.LE.0) GO TO 60
      DO 50 I=1,K
         RNUM=RNUM*RANF(ISED)
   50 CONTINUE
   60 DUM=-PARM(JP,1)*ALOG(RNUM)
C
   70 IF (DUM-PARM(JP,2)) 80,100,90
   80 DUM=PARM(JP,2)
      GO TO 100
   90 IF (DUM.LE.PARM(JP,3)) GO TO 100
      DUM=PARM(JP,3)
  100 DEV=DUM
      RETURN
C
C     LOGNORMAL
C
  110 DEV=EXP(RNORM(JP))
      RETURN
C
C     POISSON
```

```
C
  120 NPSSN=0
      P=PARM(JP,1)
      IF (P.GT.9.) GO TO 140
      Y=EXP(-P)
      X=1.
  130 RNUM=RANF(ISED)
      X=X*RNUM
      IF (X.LT.Y) GO TO 150
      NPSSN=NPSSN+1
      GO TO 130
  140 TEMP=PARM(JP,4)
      PARM(JP,4)=SQRT(PARM(JP,1))
      PARM(JP,1)=PARM(JP,1)+PARM(JP,2)
      DEV=AINT(RNORM(JP)+0.5)
      PARM(JP,1)=PARM(JP,1)-PARM(JP,2)
      PARM(JP,4)=TEMP
      RETURN
  150 KK=PARM(JP,2)
      KKK=PARM(JP,3)
      NPSSN=KK+NPSSN
      IF (NPSSN.LE.KKK) GO TO 160
      NPSSN=PARM(JP,3)
  160 DEV=NPSSN
      RETURN
C
C     BETA
C
  170 A=PARM(JP,1)
      B=PARM(JP,4)
      X=GAM(A,ISED)
      DUM=X/(X+GAM(B,ISED))
      DEV=DUM*(PARM(JP,3)-PARM(JP,2))+PARM(JP,2)
      RETURN
C
C     GAMMA
C
  180 A=PARM(JP,4)
      DUM=GAM(A,ISED)/PARM(JP,1)
      GO TO 70
C
C     SCALE FACTOR
C
  190 DEV=0.
      IF (JP.EQ.0) GO TO 200
      DEV=FLOAT(JP)/SCAL
  200 RETURN
C
C     TRIANGULAR
C
  210 DEV=TRNGL(JP)
      RETURN
C
C     WEIBULL
C
```

```
  220 DUM=PARM(JP,1)*((-1.*ALOG(RANF(ISED)))**PARM(JP,4))
      GO TO 70
C
C     DISCRETE
C
  230 DEV=DISCR(JP)
      RETURN
C
      END
      FUNCTION RNORM (JD)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      DOUBLE PRECISION SUM1,SUM2
      J=JD
      SUM=0.
      DO 10 I=1,12
         SUM=SUM+RANF(ISED)
   10 CONTINUE
      V=SUM-6.
      RNORM=V*PARM(J,4)+PARM(J,1)
      IF (RNORM-PARM(J,2)) 20,30,40
   20 RNORM=PARM(J,2)
   30 RETURN
   40 IF (RNORM.LE.PARM(J,3)) GO TO 30
      RNORM=PARM(J,3)
      RETURN
C
      END
      FUNCTION TRNGL (JD)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      DOUBLE PRECISION SUM1,SUM2
      JP=JD
      R=RANF(ISED)
      IF (R.GT.PARM(JP,1)) GO TO 10
      R=SQRT(PARM(JP,1)*R)
      GO TO 20
   10 R=1.-SQRT((1.-PARM(JP,1))*(1.-R))
   20 TRNGL=R*PARM(JP,4)
      RETURN
C
      END
      FUNCTION GAM (ALPHA,ISED)
      K=ALPHA
      FK=K
      GAM=0.
      IF (K.LE.0) GO TO 20
      PROD=1.0
      DO 10 I=1,K
         PROD=PROD*RANF(ISED)
   10 CONTINUE
```

```
      GAM=-ALOG(PROD)
   20 D=ALPHA-FK
      IF (D.LE..015) GO TO 60
      IF (D.LT..985) GO TO 30
      W=1.
      GO TO 50
   30 A=1./D
      B=1./(1.-D)
   40 X=RANF(ISED)**A
      Y=RANF(ISED)**B+X
      IF (Y.GT.1.) GO TO 40
      W=X/Y
   50 Y=-ALOG(RANF(ISED))
      GAM=GAM+W*Y
   60 RETURN
C
      END
      SUBROUTINE BETAXF
C
C*****THIS ROUTINE CONVERTS THE GIVEN BETA PARAMETERS, MEAN + VARIANCE
C*****TO (1) THE MEAN AND VARIANCE OF A BETA DISTRIBUTION OVER THE
C*****INTERVAL (0,1), (2) THE K1 + K2 VALUES NEEDED TO OBTAIN A BETA
C*****VARIATE FROM THE RATIO  OF TWO GAMMA VARIATES
C
C     PARAM(J,1) CONTAINS THE MEAN, AT CLOSE IT WILL CONTAIN K1
C     PARAM(J,2) CONTAINS THE LOWER LIMIT
C     PARAM(J,3) CONTAINS THE UPPER LIMIT
C     PARAM(J,4) CONTAINS THE VARIANCE_ AT CLOSE IT WILL CONTAIN K2
C
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      DOUBLE PRECISION SUM1,SUM2
      J=JTRIB(2)
      BMEAN=(PARM(J,1)-PARM(J,2))/(PARM(J,3)-PARM(J,2))
      BVAR=(PARM(J,4)/(PARM(J,3)-PARM(J,2)))**2
      PARM(J,1)=BMEAN*(BMEAN*(1.0-BMEAN)/BVAR-1.0)
      PARM(J,4)=PARM(J,1)*((1.0-BMEAN)/BMEAN)
      RETURN
C
      END
      SUBROUTINE PERTXF
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      DOUBLE PRECISION SUM1,SUM2
      J=JTRIB(2)
      PARM(J,1)=(4.*PARM(J,1)+PARM(J,2)+PARM(J,3))/6.0
      PARM(J,4)=(PARM(J,3)-PARM(J,2))/6.
      CALL BETAXF
      RETURN
C
      END
```

```
      SUBROUTINE GREAD
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NQQ,NSNK,
     1  NPUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G6/ NCEL(10),PROB(32,10),VAL(33,10),ISEC(3,10),MAX,NHIS,
     1  KHIST,NPNCH
   10 READ (ICRD,60) JQ,NCELS,WID
      IF (NCELS.EQ.0) RETURN
      IF (JQ.GT.MAX) GO TO 40
      READ (ICRD,70) (PROB(J,JQ),J=1,NCELS)
      READ (ICRD,80) (VAL(J,JQ),J=1,NCELS)
      WRITE (IPRT,90) JQ,NCELS,WID
      WRITE (IPRT,100) (PROB(J,JQ),J=1,NCELS)
      WRITE (IPRT,110) (VAL(J,JQ),J=1,NCELS)
      NCEL(JQ)=NCELS
      DO 20 J=2,NCELS
         PROB(J,JQ)=PROB(J,JQ)+PROB(J-1,JQ)
   20 CONTINUE
      IF (ABS(PROB(NCELS,JQ)-1.).GT..002) GO TO 50
   30 CALL ASSGN (JQ)
      VAL(33,JQ)=WID
      GO TO 10
   40 WRITE (IPRT,120) JQ,MAX
      CALL EXIT
   50 WRITE (IPRT,130) JQ,PROB(NCELS,JQ)
      PROB(NCELS,JQ)=1.
      GO TO 30
C
C
C
   60 FORMAT (2I5,E10.4)
   70 FORMAT (16F5.4)
   80 FORMAT (8E10.4)
   90 FORMAT (///10H HISTOGRAM,I5,5X,I5,6H CELLS,5X,11HCELL WIDTH=,
     1  E13.4)
  100 FORMAT (/5X,13HPROBABILITIES/(5X,8F10.4))
  110 FORMAT (/5X,11HCELL VALUES/(5X,8E13.4))
  120 FORMAT (/1X,9(1H*),5X,54HWARNING-ATTEMPTED TO READ DISCRETE DISTRI
     1BUTION NUMBER,I4/15X,39HMAXIMUM ALLOWED BY PRESENT DIMENSION IS,
     2  I4)
  130 FORMAT (/1X,9(1H*),47HWARNING-PROBABILITIES FOR DISCRETE DISTRIBUT
     1ION,I4,16H DO NOT SUM TO 1/11X,4HSUM=,F10.5,23H   SUM WILL BE SET
     2TO 1)
      END
      SUBROUTINE GSAVE (NODE,JQ,K,KWORD)
      DIMENSION XCELS(32)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NQQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUMS(300),NT2C2,EPS
      COMMON /G3/ KST1(100),XLOW(200),NREL1(300),NREL2(300),NREL(300),
     1  MREL(300),KST4(100),KST3(100),NTYPE(300),WIDTH(200),NSTS,NCTS
      COMMON /G5/ MSTN(2000,2),MST(300),KST2(300),IGRF,JGRAF,SCAL
      COMMON /G6/ NCEL(10),PROB(32,10),VAL(33,10),ISEC(3,10),MAX,NHIS,
     1  KHIST,NPNCH
      DOUBLE PRECISION SUM1,SUM2
```

```
      IF (JQ.GT.MAX) GO TO 20
      KK=KST2(NODE)
      IF (KST1(KK).NE.NODE) GO TO 20
      KQ=KK+(KK-1)*NCTS
      SNUM=SUM3(KQ)
      IF (SNUM.LE..5) CALL ERROR (14)
      NCEL(JQ)=32
      WID=ABS(WIDTH(KK))
      IF (WID.EQ.0.) GO TO 20
      START=XLOW(KK)-2.0*WID
      IF (K.EQ.7) GO TO 30
      CUM=0.
      DO 10 J=1,32
         VAL(J,JQ)=START+WID*FLOAT(J)
         CUM=CUM+FLOAT(JCELS(KK,J))/SNUM
         PROB(J,JQ)=CUM
   10 CONTINUE
      VAL(33,JQ)=WID
      CALL ASSGN (JQ)
      RETURN
   20 WRITE (IPRT,60) KWORD,NODE,JQ,MAX
      CALL EXIT
   30 WRITE (NPNCH,70) JQ,WID
      DO 40 J=1,32
         XCELS(J)=FLOAT(JCELS(KK,J))/SNUM
   40 CONTINUE
      WRITE (NPNCH,80) XCELS
      DO 50 J=1,32
         XCELS(J)=START+WID*FLOAT(J)
   50 CONTINUE
      WRITE (NPNCH,90) XCELS
      RETURN
C
C
   60 FORMAT (///1X,9(1H*),25HWARNING-ON KEYWORD CARD =,A4,I5,5H   AS,
     1  IS,1H=/10X,67H EITHER NODE IS NOT A STAT NODE, WIDTH OF HISTOGRA
     2M EQUALS ZERO    ,/10X,51H OR DISCRETE DISTRIBUTION REQUESTED IS GR
     3EATER THAN,I5)
   70 FORMAT (I5,3X,2H32,E10.4)
   80 FORMAT (16F5.4)
   90 FORMAT (8E10.4)
      END
C
      FUNCTION DISCR (JP)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G6/ NCEL(10),PROB(32,10),VAL(33,10),ISEC(3,10),MAX,NHIS,
     1  KHIST,NPNCH
      JQ=JP
      IF (JQ.GT.MAX) GO TO 40
      K=ISEC(3,JQ)
      RNUM=RANF(ISED)
      N=3.*RNUM
      IF (N.LE.0) GO TO 10
      K=ISEC(N,JQ)
```

```
   10 NCELS=NCEL(JQ)-1
      DO 20 J=K,NCELS
         IF (RNUM.LT.PROB(J,JQ)) GO TO 30
   20 CONTINUE
      J=NCEL(JQ)
   30 DISCR=VAL(J,JQ)+VAL(33,JQ)*RANF(ISED)
      GO TO 50
   40 WRITE (IPRT,60) JQ
      CALL ERROR (99)
   50 RETURN
C
C
C
   60 FORMAT (/1X,9(1H*),44HREQUESTED DEVIATE FROM DISCRETE DISTRIBUTION
     1,I4)
      END
      SUBROUTINE ASSGN (JD)
      COMMON /G6/ NCEL(10),PROB(32,10),VAL(33,10),ISEC(3,10),MAX,NHIS,
     1  KHIST,NPNCH
      JQ=JD
      NCELS=NCEL(JQ)
      DO 10 J=1,NCELS
         IF (PROB(J,JQ).GT.0.) GO TO 20
   10 CONTINUE
   20 ISEC(3,JQ)=J
      K=1
      DO 50 J=1,NCELS
         N=3.*PROB(J,JQ)
         IF (N-K) 50,30,40
   30    ISEC(K,JQ)=J
         K=K+1
         GO TO 50
   40    ISEC(K,JQ)=J
         ISEC(K+1,JQ)=J
         RETURN
   50 CONTINUE
      RETURN
C
      END
      SUBROUTINE ADJST
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1  NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1  SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      COMMON /G3/ KST1(100),XLOW(200),NREL1(300),NREL2(300),NREL(300),
     1  MREL(300),KST4(100),KST3(100),NTYPE(300),WIDTH(200),NSTS,NCTS
      COMMON /G6/ NCEL(10),PROB(32,10),VAL(33,10),ISEC(3,10),MAX,NHIS,
     1  KHIST,NPNCH
      DOUBLE PRECISION SUM1,SUM2
C
      IF (KHIST.EQ.0) GO TO 40
      KCLCT=NSTS*NCTS
      NHIST=NSTS*2+2
      DO 20 KH=1,NHIST
         IF (WIDTH(KH).GE.0.) GO TO 20
```

```
      KQ=KH+(KH-1)*NCTS
      IF (KH.GT.NSTS) KQ=KCLCT+KH
      IF (SUM3(KQ).LT.1.5) GO TO 20
      XL=SUM4(KQ)
      XH=SUM5(KQ)
C
C***** SCALE THE HISTOGRAM
C
      PT=(XH-XL)/32.
      IF (ABS(PT).LE.1.E-10) GO TO 20
      IN=1
      IF (PT.GE.100.) IN=-1
      I=-1
   10 I=I+1
      P=10.**(I*IN)
      Z=P*PT
      IF (Z.LT.10..OR.Z.GE.100.) GO TO 10
      K=10.*(Z-AINT(Z))
      PT=.5
      IF (K.GT.5) PT=1.0
      PT=AINT(Z)+PT
      WID=PT*10.**(-I*IN)
      WIDTH(KH)=-WID
      K=IFIX(XL/WID)+1
      IF (XL.LT.0.) K=K-1
      XLOW(KH)=WID*FLOAT(K)
   20 CONTINUE
C
C***** PLOT HISTOGRAMS
C
      REWIND NHIS
      DO 30 I=1,KHIST
         READ (NHIS) OBS,KH
         WIDTH(KH)=-WIDTH(KH)
         CALL HISTO (OBS,KH)
         WIDTH(KH)=-WIDTH(KH)
   30 CONTINUE
      REWIND NHIS
   40 CONTINUE
      RETURN
C
      END
      SUBROUTINE ERROR (J)
      DIMENSION NSET(5600)
      COMMON QSET(5600)
      COMMON /G1/ MFA,MXC,IPRT,ICRD,IM1,IM2,MAXQS,NT2R,NC2R,NOQ,NSNK,
     1 NRUN,NRNS,ISED,TNOW,ATRIB(6),JTRIB(6),NAME(20),JCELS(200,32)
      COMMON /G2/ MFE(300),MLE(300),NQ(300),PARM(100,4),SUM1(300),
     1 SUM2(300),SUM3(300),SUM4(300),SUM5(300),NT2C2,EPS
      COMMON /G3/ KST1(100),XLOW(200),NREL1(300),NREL2(300),NREL(300),
     1 MREL(300),KST4(100),KST3(100),NTYPE(300),WIDTH(200),NSTS,NCTS
      COMMON /G4/ XSTUS(100),SUMCT,CSTUS(100),NCND,NCNL,NCNU,T2,C2,
     1 ITFLG,ICFLG,TYYY,TCCC,TIMTD,COSTD
      EQUIVALENCE (NSET(1),QSET(1))
      DOUBLE PRECISION SUM1,SUM2
```

```
      DATA KT,KC/4HTIME,4HCOST/,KN/4HCNOD/
      WRITE (IPRT,130)
      NCLT=NSTS*(NCTS+2)+2
      IF (J.LT.0) GO TO 10
      WRITE (IPRT,140) J
   10 WRITE (IPRT,150) TNOW
      DO 30 L=1,NOQ
         IF (NQ(L).LE.0) GO TO 30
         WRITE (IPRT,160) L
         INDX=MFE(L)
   20    NP=INDX+IM1+IM2
         NS=NP+1
         NF=INDX+IM1-1
         WRITE (IPRT,170) INDX,NSET(NP),NSET(NS),(NSET(K),K=INDX,NF)
         NF=NF+1
         NL=NF+IM2-1
         WRITE (IPRT,180) (QSET(K),K=NF,NL)
         INDX=NSET(NS)
         IF (INDX.GE.77777) GO TO 30
         GO TO 20
   30 CONTINUE
      IF (J.EQ.-1.OR.(J.EQ.-3.AND.NPUN.EQ.1)) GO TO 120
      WRITE (IPRT,130)
      IF (NCLT.LE.0) GO TO 80
      WRITE (IPRT,190)
      MCTS=NCTS+1
      DO 50 I=1,NSTS
         NC=0
         DO 40 M=1,MCTS
            L=(I-1)*MCTS+M
            WRITE (IPRT,200) KST1(I),NC,SUM1(L),SUM2(L),SUM3(L),SUM4(L),
     1         SUM5(L)
            NC=NC+1
   40    CONTINUE
   50 CONTINUE
      MCLT=NCLT-NSTS-2
      NC=MCLT+1
      MCTS=MCLT+NSTS
      DO 60 I=NC,MCTS
         L=I-MCLT
         WRITE (IPRT,210) KST1(L),SUM1(L),SUM2(L),SUM3(L),SUM4(L),
     1      SUM5(L)
   60 CONTINUE
      IF (NCND.EQ.0) GO TO 70
      NC=NCLT-1
      WRITE (IPRT,220) (KN,SUM1(I),SUM2(I),SUM3(I),SUM4(I),SUM5(I),I=NC,
     1  NCLT)
   70 WRITE (IPRT,130)
   80 NHIST=NSTS*2+2
      IF (NHIST.LE.0) GO TO 110
      WRITE (IPRT,230)
      NC=2*NSTS
      DO 90 I=1,NC
         M=I
         K=KT
```

```
      IF (I.LE.NSTS) GO TO 90
      M=I-NSTS
      K=KC
   90 WRITE (IPRT,240) KST1(M),K,(JCELS(I,K),K=1,MXC)
      IF (NCND.EQ.0) GO TO 110
      NC=NHIST-1
      DO 100 I=NC,NHIST
         WRITE (IPRT,250) KN,(JCELS(I,K),K=1,MXC)
  100 CONTINUE
  110 CALL EXIT
  120 RETURN
C
C
C
  130 FORMAT (1H1)
  140 FORMAT (//36X,16HERROR EXIT, TYPE,I3,7H ERROR.)
  150 FORMAT (//20H FILE STATUS AT TIME,E12.4/5X,42HP=PREDECESSOR POINTE
     1R  S=SUCCESSOR POINTER)
  160 FORMAT (//1X,5HFILE ,I4)
  170 FORMAT (/1X,5HCELL=,I6,4X,2HP=,I6,3X,2HS=,I6/1X,5HJTRIB,8X,
     1  7I12/(14X,7I12))
  180 FORMAT (1X,5HATRIB,8X,7E12.4/(14X,7E12.4))
  190 FORMAT (/,' SUM, SUM OF SQ., NUMB. OF OBS., MIN., MAX.',/)
  200 FORMAT (1X,I4,1H(,I3,1H),5E13.4)
  210 FORMAT (1X,I4,5HCOST ,5E13.4)
  220 FORMAT (5X,A5,5E13.4)
  230 FORMAT (/12H ARRAY JCELS/)
  240 FORMAT (/1X,I4,A4,16X,11I6/(25X,11I6))
  250 FORMAT (/5X,A4,16X,11I6/(25X,11I6))
      END
```