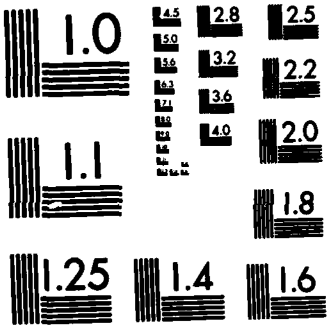
AD-A123 055   SYSTEM RELIABILITY ANALYSIS: FOUNDATIONS(U) CALIFORNIA   1/1
UNIV BERKELEY OPERATIONS RESEARCH CENTER R E BARLOW
JUL 82 ORC-82-8 DAAG29-81-K-0160

UNCLASSIFIED   F/G 9/3   NL

END
FILMED
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

SYSTEM RELIABILITY ANALYSIS: FOUNDATIONS[†]

Operations Research Center Research Report No. 82-8

Richard E. Barlow

July 1982

U. S. Army Research Office - Research Triangle Park

DAAG29-81-K-0160

Operations Research Center
University of California, Berkeley

APPROVED FOR PUBLIC RELEASE;
DISTRIBUTION UNLIMITED.

[†] This paper was presented at the NATO Advanced Study Institute on "Electronic Systems Effectiveness and Life Cycle Costing," Norwich, England, July 1982 and will appear in the Conference Proceedings.

THE VIEW, OPINIONS, AND/OR FINDINGS CONTAINED IN THIS REPORT ARE THOSE OF THE AUTHOR(S) AND SHOULD NOT BE CONSTRUED AS AN OFFICIAL DEPARTMENT OF THE ARMY POSITION, POLICY, OR DECISION, UNLESS SO DESIGNATED BY OTHER DOCUMENTATION.

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		**READ INSTRUCTIONS BEFORE COMPLETING FORM**
1. REPORT NUMBER ORC 82-8	2. GOVT ACCESSION NO. AD-A123	3. RECIPIENT'S CATALOG NUMBER 055
4. TITLE *(and Subtitle)* SYSTEM RELIABILITY ANALYSIS: FOUNDATIONS		5. TYPE OF REPORT & PERIOD COVERED Research Report
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Richard E. Barlow		8. CONTRACT OR GRANT NUMBER(s) DAAG29-81-K-0160
9. PERFORMING ORGANIZATION NAME AND ADDRESS Operations Research Center University of California Berkeley, California 94720		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS P-18195-M
11. CONTROLLING OFFICE NAME AND ADDRESS U. S. Army Research Office P.O. Box 12211 Research Triangle Park, North Carolina 27709		12. REPORT DATE July 1982 13. NUMBER OF PAGES 26
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Network Reliability
Logic Trees
Marginal Importance
Availability Formulas

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

(SEE ABSTRACT)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

S/N 0102-LF-014-6601

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

ABSTRACT

The philosophical and computational foundations of a system reliability analysis are discussed. Recent advances in network and logic tree computational methods are reviewed. Long run performance formulas for systems subject to preventive maintenance are given.

SYSTEM RELIABILITY ANALYSIS: FOUNDATIONS

Richard E. Barlow
Operations Research Center
University of California
Berkeley, CA 94720

1. INTRODUCTION

System reliability analysis problems arise in many practical engineering areas. Some of these include communication networks, electrical power systems, water transmission systems, nuclear power reactors, and transportation systems. We will illustrate some of the ideas basic to a system reliability analysis via our experience in analyzing a proposed Satellite X-ray Test Facility (SXTF). This facility would test space satellites relative to an electromagnetic radiation environment.

The purpose of a system reliability analysis is to acquire *information* about a system of interest relative to *making decisions* based on considerations of availability, reliability, and safety as well as any inherent engineering risks. The philosophy and guidelines for a system analysis have been discussed in several excellent introductory chapters by David Haasl in a *Fault Tree Handbook* (1981). Broadly speaking, there are two important aspects to a system analysis: (1) An INDUCTIVE ANALYSIS stage and (2) A DEDUCTIVE ANALYSIS stage. In the inductive analysis stage we gather and organize available information on the system. We define the system, describe its functional purpose and determine its critical components. At this stage, we ask the question: WHAT can happen to the system as a result of a component failure or a human error? We hypothesize and guess possible system failure scenarios as well as system success modes. A Preliminary Hazard Analysis is often performed at the system level. A Failure Modes and Effects Analysis is conducted at the component level.

The DEDUCTIVE ANALYSIS aspect of a system reliability analysis answers the question: HOW can a system fail (or succeed) or be unavailable? A logic tree (or fault tree if we are failure oriented) is often the best device for deducing how a major system failure event could possibly occur. However, its construction depends on a thorough understanding of the system and the results of the system inductive analysis. A block diagram or a network graph is a useful device for representing a successfully functioning system. Since the network graph is close to a system functional representation, it cannot capture abstract system failure and human error events as well as the logic tree representation. However, from the point of view of mathematical probability analysis, the network graph representation seems to be correspondingly easier to analyze.

The Operations Research Center at Berkeley has completed two projects so far involving extensive system reliability analysis of a proposed X-ray test facility. One subsystem providing the photon source is composed of 192 individual modules. They are attached at one end to the Marx capacitor bank, and the other end penetrates into the vacuum chamber and terminates in the X-ray producing diode (see Figure 1). Each module is filled with water that is separated from the oil in the Marx tank by an epoxy diaphragm and from the vacuum chamber by a styrene insulator plate.

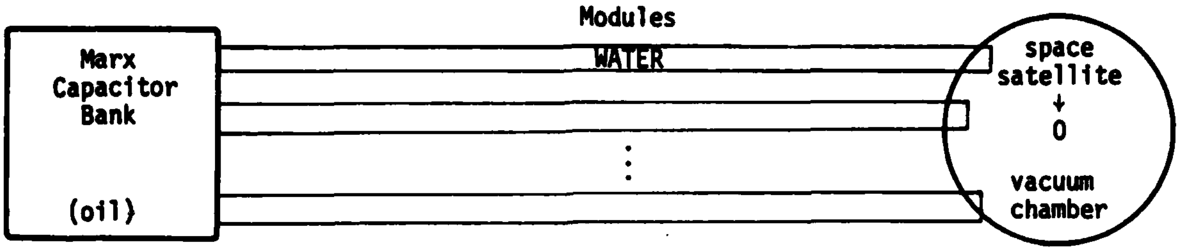


FIGURE 1. PROPOSED X-RAY TEST FACILITY

In the inductive phase of our system analysis we listed all possible mechanical and electrical failure modes that we could envision. This led to a critical components list including assessed failure rates. For each member of the list, a detailed failure modes and effects form was filled out by engineers concerned with the project. This, together with a detailed discussion of possible system faults, constituted our "inductive analysis."

It is well-known that system failures often occur at subsystem *interfaces*. In the deductive phase of our analysis we were most concerned with the oil-water and also the water-vacuum interfaces. Fault trees were constructed for water leakage from the tube into the vacuum chamber, for oil/water mixing and also for satellite contamination. These fault trees pinpointed failure modes which might have been otherwise overlooked. In particular, as a result of these fault trees, certain components were redesigned to prevent potential failures. The fault trees provided useful visual tools for describing the logic leading up to possible serious system failure events. They provided the basis for contending that all likely critical failure events have been found and studied. Finally, a simple block diagram of our system was used to implement a system availability analysis. In the next section we show how to analyze, probabilistically, more complex networks.

2. CALCULATION OF SYSTEM RELIABILITY

The logical relationship between component events and system events is best represented by a network graph or a logic tree. A Boolean expression can be derived from either representation which can then be used to calculate the probability of system events of interest. However, recent research on the computational complexity of network reliability problems has shown that Boolean computational methods are not efficient. Chang (1981) in Chapter 3 of his Ph.D. thesis discusses the Boolean algebra approach and Backtrack algorithms in this regard.

Networks

Suppose we consider a network graph representation such as the undirected network in Figure 2.

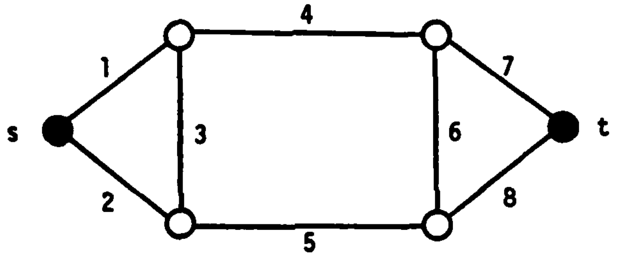


FIGURE 2. UNDIRECTED TWO TERMINAL NETWORK

In this case, system success occurs if there is at least one working path of nodes and arcs from source s to terminal t. Let the Boolean indicator

$$x_i = \begin{cases} 1 & \text{if arc } i \text{ works} \\ 0 & \text{otherwise.} \end{cases}$$

For convenience, suppose nodes are perfect so that $(x_1, x_2, \ldots, x_8)$ is a state vector for our network. Let

$$\phi(x_1, x_2, \ldots, x_8) = \begin{cases} 1 & \text{if } s \text{ and } t \text{ can communicate} \\ 0 & \text{otherwise.} \end{cases}$$

Such systems are called *coherent systems* in Barlow and Proschan (1981). Basically, ϕ is coherent if it is nondecreasing coordinatewise. All coherent systems, ϕ , can be represented as two terminal networks with possible replication of some arcs.

A minimal path set for the network in Figure 2 is $P_1 = \{1,4,7\}$ for example. There are 8 such min path sets $\{P_1, P_2, \ldots, P_8\}$. Hence, ϕ can be represented as

$$\phi(x_1, x_2, \ldots, x_8) = 1 - \prod_{r=1}^{8} \left(1 - \prod_{i \epsilon P_r} x_i\right) \overset{\text{def}}{=} \coprod_{r=1}^{8} \prod_{i \epsilon P_r} x_i \ . \tag{2.1}$$

By expanding this expression, using the usual arithmetic (*not* Boolean arithmetic) and replacing x_i^n by x_i, we can obtain an expression suitable for computing the system success probability. Assuming arc failure events are statistically independent we need only replace x_i by p_i, the probability arc i works, in the resulting expression.

However, there is a far more efficient method for doing this calculation - called the *factoring algorithm*. The idea is to first perform all possible series and parallel probability reductions and then pivot on an arc. Let $\underline{p} = (p_1, p_2, \ldots, p_8)$ and $h(\underline{p})$ denote the probability that s and t communicate. If we "pivot" on arc i then we obtain the "pivotal" decomposition of $h(\underline{p})$, namely

$$h(\underline{p}) = p_i h(1_i, \underline{p}) + (1 - p_i) h(0_i, \underline{p}) \tag{2.2}$$

where $(1_i, \underline{p}) = (p_1, p_2, \ldots, p_{i-1}, 1_i, p_{i+1}, \ldots, p_8)$. This, together with series and parallel reductions, is the mathematical basis for the factoring algorithm.

In Figure 2 no series or parallel reductions are possible, so we pivot on arc 1. That is, we short arc 1 on the left and delete arc 1 on the right. Series and parallel reductions are now possible on the two modified graphs. After performing these reductions, we again pivot. In our binary computational tree in Figure 3 there are 4 leaves at the bottom of the tree. Neglecting parallel and series reductions except at the last stage, we have performed only $2(4) - 1 = 7$ operations to achieve our reliability computation. If each arc i has probability p of working, it is easy to see that the system reliability in this case is

$$\begin{aligned}
&P\{s \text{ can communicate with the terminal } t\} \\
&= h(p) \\
&= p^2(((((p \amalg p)p) \amalg p)p) \amalg p) + p(1-p)(((p \amalg p)p)(p^2 \amalg p)) \\
&\quad + p(1-p)((p(p \amalg p)) \amalg (p^2))p + (1-p)^2((p^3 \amalg p)p^2) \\
&= (p^3 + p^4 + p^5 - 5p^6 + 4p^7 - p^8) + (2p^4 - p^5 - 4p^6 + 4p^7 - p^8) \\
&\quad + (3p^4 - 4p^5 - p^6 + 3p^7 - p^8) + (p^3 - 2p^4 + 2p^5 - 3p^6 + 3p^7 - p^8) \\
&= \underline{2p^3 + 4p^4 - 2p^5 - 13p^6 + 14p^7 - 4p^8} \ .
\end{aligned}$$

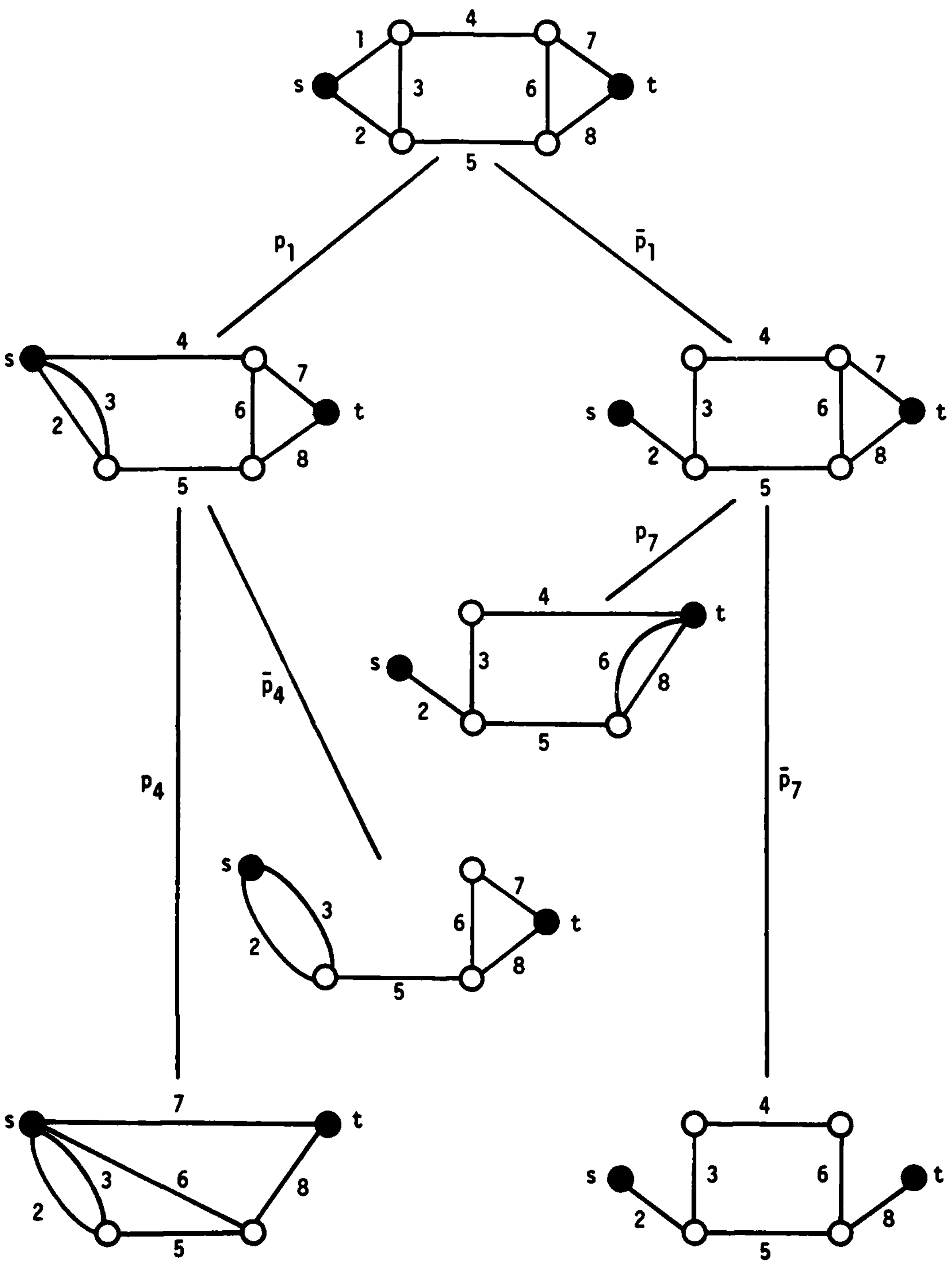


FIGURE 3. BINARY COMPUTATIONAL TREE USING THE FACTORING ALGORITHM

The lower case "ip" operator, $\amalg$, corresponds to calculating the reliability of parallel arcs; i.e.,

$$p_i \amalg p_j = p_i + p_j - p_i p_j \ .$$

In Figure 3, $\bar{p}_i = 1 - p_i$.

Linear and polynomial time algorithms are now available for computing network reliability when the underlying graph has a series-parallel topology. For example, the graph in Figure 4 is called a topologically series-parallel graph even though the same graph in Figure 2 with distinguished nodes s and t is *not* series-parallel with respect to reliability computation.

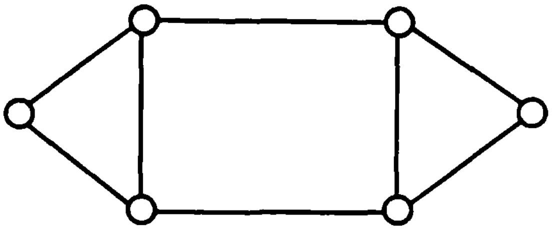


FIGURE 4. A TOPOLOGICAL SERIES-PARALLEL GRAPH (NO DISTINGUISHED NODES)

For *undirected* networks, the basic reference is A. Satyanarayana and Kevin Wood (1982). For *directed* networks, the basic reference is Avinash Agrawal and A. Satyanarayana (1982).

If the arc reliability, p , is unknown but there is data available, then we may assess our uncertainty about p by a probability density. If a Beta prior density is used, then the posterior density is also Beta and is of the form

$$\pi(p|a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} p^{a-1}(1-p)^{b-1} \ . \tag{2.3}$$

Our final system reliability assessment is now

$$R = \int_0^1 h(p)\pi(p|a,b)dp \ . \tag{2.4}$$

A common mistake is to compute the expected arc reliability

$$\int_0^1 p\pi(p|a,b)dp = \frac{a}{a+b}$$

and compute $h\left(\frac{a}{a+b}\right)$. However, $R \neq h\left(\frac{a}{a+b}\right)$.

Logic Trees

Logic tree (or fault tree) analysis is a detailed deductive analysis that usually requires considerable system information. It is best applied during the design stages of a system. At that point, it can identify hazardous conditions and potential accidents in a system design and thus can help eliminate costly design changes and retrofits that would otherwise have to be made later in the system life cycle. Undesired events requiring logic tree analysis are identified either by inductive analysis or by intuition. These events are usually undesired system states that can occur as a result of subsystem functional faults.

A logic tree is a model that graphically and logically represents the various combinations of possible events, both fault and normal, occurring in a system that lead to the top undesired event. The logic tree is so structured that the undesired event appears as the top event in the logic tree. The sequences of events that lead to the undesired event are shown below the top event and are logically linked to the undesired event by standard OR and AND gates. The input events to each logic gate that are also outputs of other logic gates at a lower level are shown as rectangles. (Rectangles are called gate events.) These events are developed even further until the sequences of events lead to basic causes. The basic events appear as circles and diamonds on the bottom of the fault tree and represent the limit of resolution. The circle represents an internal or primary failure of a system element when exercised within the design envelop of the system. The diamond represents a failure, other than a primary failure, that is purposely not further developed. Gate nodes correspond to intermediate events while the top node usually corresponds to a very serious system failure event. In Figure 6, all arcs are regular with the exception of the complementing arc joining nodes 6 and 4, and this arc is distinguished by the symbol "$\sim$."

Associated with each gate is a logic symbol: OR gates have a plus symbol (for set union) while AND gates have a product (·) symbol (for set intersection). For example, output event 3 occurs if either input event 4 or 5 (or both) occur. Likewise output event 5 occurs only if *both* input events 7 and 8 occur. Since the arc connecting gate events 4 and 6 is complemented, gate event 4 occurs only if basic event 11 occurs and gate event 6 does *not* occur.

A complete reliability analysis on an extensive system such as the SXTF System normally requires three levels of fault tree development, as shown in Figure 5. The upper level, called the top structure, includes the top undesired event and the sub-undesired events that are potential accidents and hazardous conditions that are immediate causes of the top event. The next level of the logic tree divides the operation of the system into phases, subphases, etc., until the system environment remains

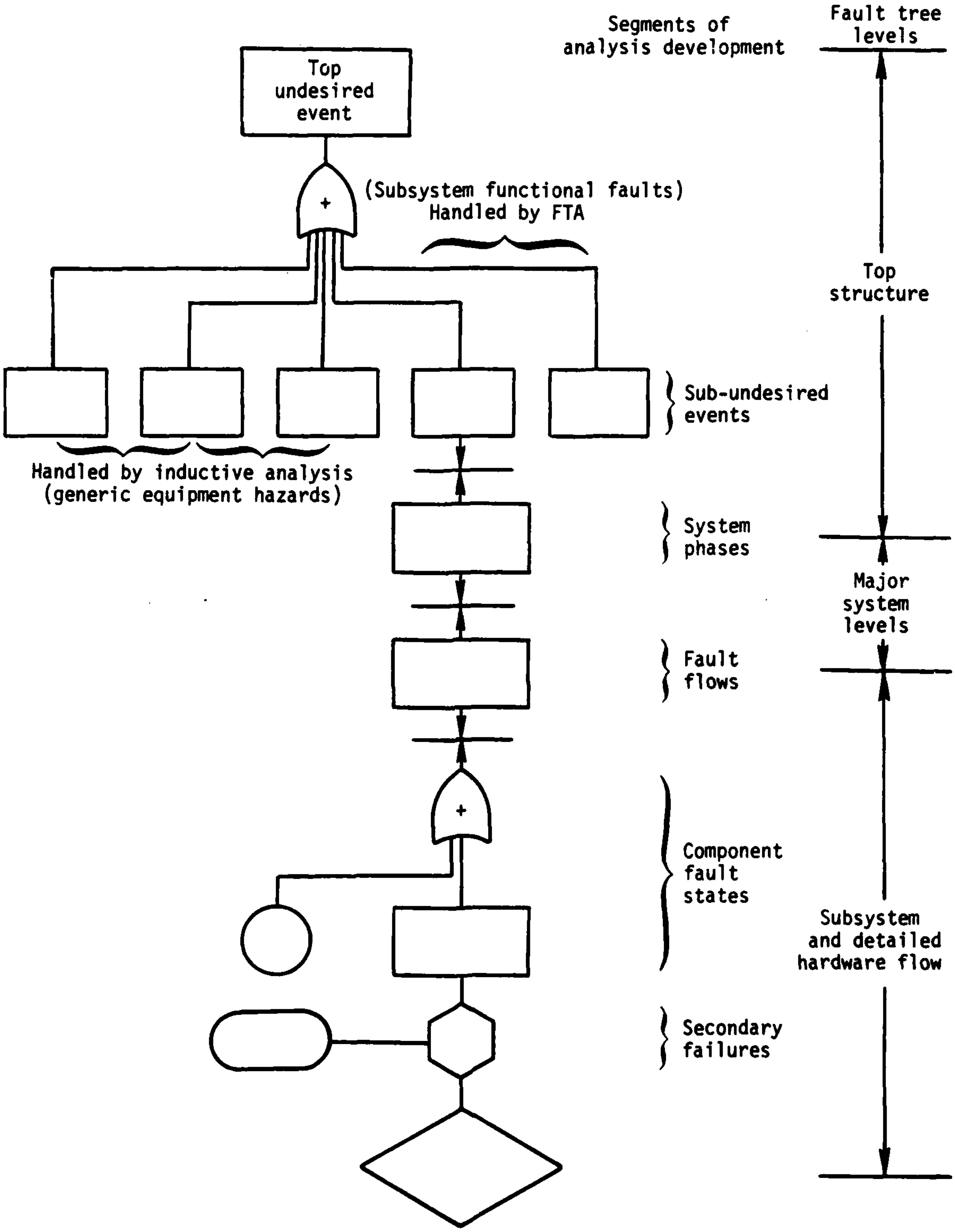

Segments of analysis development
Fault tree levels
Top undesired event
+
(Subsystem functional faults) Handled by FTA
Top structure
Sub-undesired events
Handled by inductive analysis (generic equipment hazards)
System phases
Major system levels
Fault flows
+
Component fault states
Subsystem and detailed hardware flow
Secondary failures

FIGURE 5. LEVELS OF LOGIC TREE DEVELOPMENT

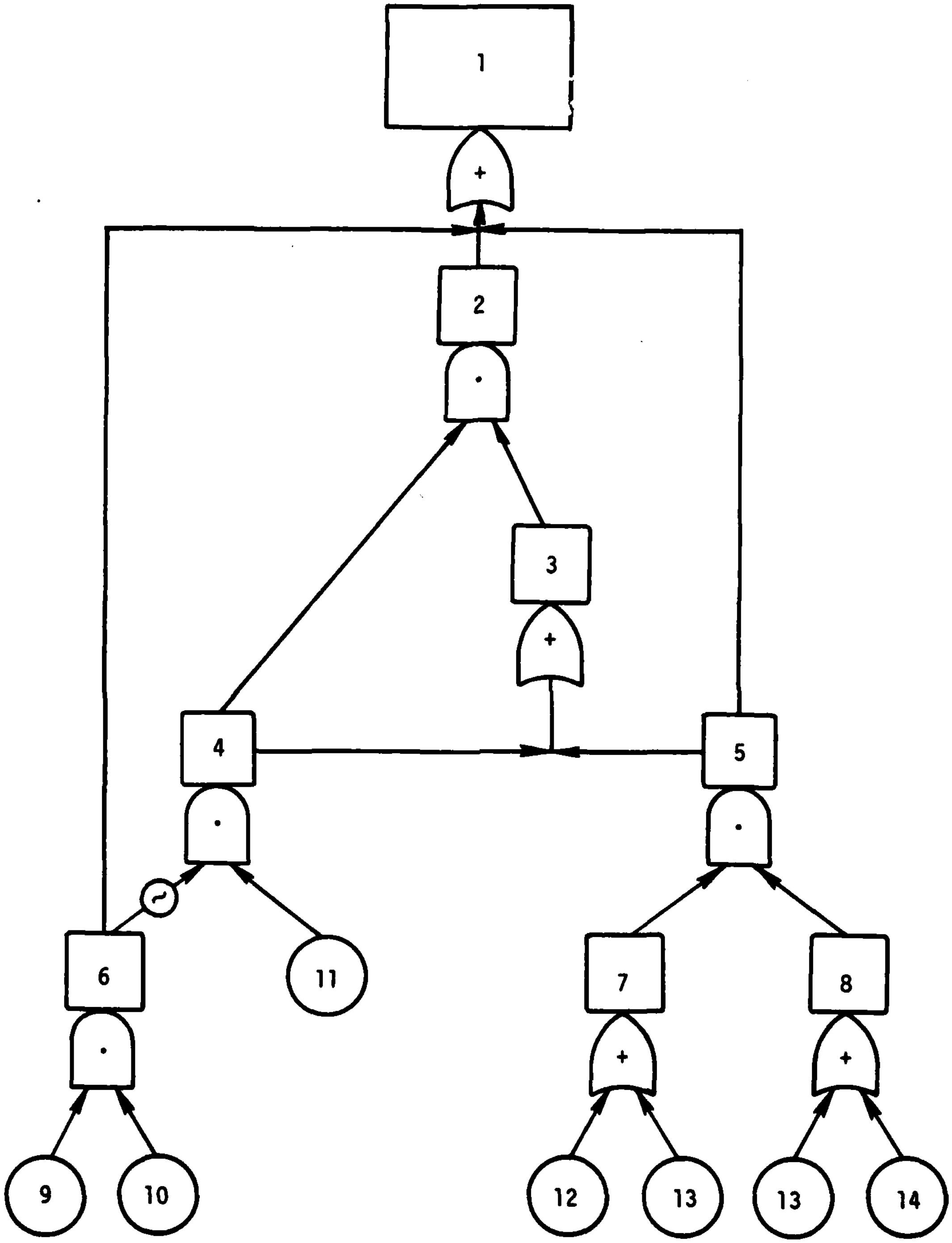


FIGURE 6. EXAMPLE LOGIC TREE

constant and the system characteristics do not change the fault environment. In this second level of fault tree development, the analyst examines system elements from a functional point of view. He uses a structuring process to develop fault flows within the system that deductively lead to a subsystem and detailed hardware flow, which is the third level of the fault tree. At the third level, the analyst is faced with one of the most difficult aspects of logic tree analysis. He must determine if basic events are statistically independent. He then focuses his attention on common events that can simultaneously fail two or more system elements. The effects of any common environmental or operational stresses are studied, as well as the effects of the human factor in the testing, maintenance, and operation of the system.

Once the logic tree is constructed, all logically possible accident scenarios (called minimal cuts) can be obtained. There are many algorithms and computer programs for finding minimal cuts (or prime implicants for general logic trees). One of the best of these is a computer program called FTAP due to Randall Willie (1978). The minimal cuts can then be used to compute the probability of gate events including the TOP event. A sensitivity analysis can be performed using a so-called marginal importance measure which is essentially the partial derivative of system reliability with respect to component reliability.

Mathematics of Fault Tree Analysis

Boolean switching theory is basic for the mathematics of fault tree analysis. For the fault tree node set $U = [1,2, \ldots, q]$, let $x_1, x_2, \ldots, x_q$ be Boolean variables assuming values 0 or 1 and let $\underline{x} = (x_1, x_2, \ldots, x_q)$. (In Figure 6, $q = 14$.) For any u in U , let $x_{-u} \equiv 1 - x_u$. The index set for complements is $-U \equiv [-1,-2, \ldots, -q]$ and $(u,-u)$ is a complementary pair of indices.

Expressions may be formed using $x_1, \ldots, x_q$, $x_{-1}, \ldots, x_{-q}$ and the ordinary Boolean relations of product and sum. An arbitrary nonempty family $\mathcal{I}$ of subsets of $U \cup (-U)$ (not necessarily distinct) is identified with the Boolean sum-of-products expression

$$\sum_{I \varepsilon \mathcal{I}} \prod_{i \varepsilon I} x_i \ , \tag{2.5}$$

where I is a member of the family $\mathcal{I}$. (Remember, the arithmetic is Boolean.) The notation $/\mathcal{I}/\underline{x}$ denotes the value of this expression for a given vector $\underline{x}$ of 0's and 1's, that is,

$$/\mathcal{I}/\underline{x} \equiv \max_{I \varepsilon \mathcal{I}} \left(\min_{i \varepsilon I} x_i \right) = \sum_{I \varepsilon \mathcal{I}} \prod_{i \varepsilon I} x_i \ . \tag{2.6}$$

Given nonempty families $\mathcal{I}$ and $\mathcal{J}$ of subsets of $U \cup (-U)$, $/\mathcal{I}/ = /\mathcal{J}/$ means that for all $\underline{x}$ $/\mathcal{I}/\underline{x} = /\mathcal{J}/\underline{x}$. It is further assumed that *no set of a family contains a complementary pair*. Whenever a new family is constructed, any set containing complementary pairs is simply eliminated.

A family is said to be *minimal* if all sets are distinct and for any two sets of the family, neither is a subset of the other. For any family $\mathcal{I}$, let $m(\mathcal{I})$ (the "minimization" of $\mathcal{I}$) be the minimal family obtained by eliminating duplicate sets and those which contain another set of $\mathcal{I}$. For instance, $m([\{2,3\}, \{1,2,3\}]) = [\{2,3\}]$. Of course, for any $\mathcal{I}$, $/m(\mathcal{I})/ \equiv /\mathcal{I}/$. The first task of a fault tree analysis is to obtain a certain minimal family of sets of $U \cup (-U)$ called a *prime implicant family*. We are only interested in prime implicant families for fault tree nodes which we wish to analyze since such families are unique and determine the Boolean expression for the node indicator. For Figure 6 and node 1,

$$\mathcal{P} = [\{9,10\},\{12,14\},\{13\},\{11\}]$$

is a prime implicant family and

$$x_1 = \sum_{P\epsilon\mathcal{P}} \prod_{i\epsilon P} x_i , \qquad (2.7)$$

where P is a member of the family $\mathcal{P}$ and x_1 is the indicator for the top event in Figure 6. The first task of a fault tree analysis is to obtain the prime implicant families for fault tree nodes of special interest.

For trees without complemented arcs, the prime implicants are called minimal cut sets. The minimal cut set family for a large fault tree (having, say, more than 100 gate nodes) may consist of millions of sets, if the tree has an appreciable number of OR-type gates. A. Rosenthal (1975) has shown that the general problem of finding the complete minimal cut set family associated with a fault tree is a member cf the class of NP-complete problems. (A class of problems for which it is conjectured that no algorithm exists which will always run on a computer within a polynomial time bound.) Hence we cannot expect to devise an algorithm whose running time is bounded for all fault trees by a polynomial in, say, the number of fault tree nodes. The serious analyst should probably not rely on the same method for every fault tree.

Sensitivity Analysis for Coherent Systems and Logic Trees

Often the relative importance ranking of components in a coherent system (or of basic events in a logic tree) is more useful than the probability of system success or failure. We will use coherent system terminology to illustrate the concept of *marginal importance*. We define the marginal importance, $I_h(i)$, of component i to be

$$I_h(i) = \frac{\partial h(\underline{p})}{\partial p_i} \qquad (2.8)$$

when components are statistically independent and $h(\underline{p})$ is the system reliability. From the pivotal decomposition in (2.2), it is clear that

$$\frac{\partial h(\underline{p})}{\partial p_i} = h(1_i,\underline{p}) - h(0_i,\underline{p}) \ . \qquad (2.9)$$

This is also valid for general logic trees where $h(\underline{p})$ is the probability of the Top Event. If, in addition, ϕ is nondecreasing coordinatewise,

$$I_h(i) = \frac{\partial h(\underline{p})}{\partial p_i} = P\{\phi(1_i,\underline{x}) - \phi(0_i,\underline{x}) = 1 \mid \underline{p}\} \qquad (2.10)$$

so that $I_h(i)$ is the probability that component i is "critical" at a given time instant. This means that with i working the system works, but with i failed, the system is failed.

The reliability importance of components may be used to evaluate the effect of an improvement in component reliability on system reliability, as follows. By the chain rule for differentiation,

$$\frac{dh}{dt} = \sum_{j=1}^{n} \frac{\partial h}{\partial p_j} \frac{dp_j}{dt} \ ,$$

where t is a common parameter, say, the time elapsed since system development began. Using (2.8), we have

$$\frac{dh}{dt} = \sum_{j=1}^{n} I_h(j) \frac{dp_j}{dt} \ . \qquad (2.11)$$

Thus the rate at which system reliability grows is a weighted combination of the rates at which component reliabilities grow, where the weights are the reliability importance numbers.

From (2.11), we may also obtain

$$\Delta h \cong \sum_{j=1}^{n} I_h(j)\Delta p_j \ , \qquad (2.12)$$

where Δh is the perturbation in system reliability corresponding to perturbations

Δp_j in component reliabilities. As in (2.11), the reliability importance numbers enter as weights. Thus small improvements Δp_j in component reliabilities lead to a corresponding improvement Δh in system reliability in accordance with (2.12).

Examples:

Assume components have been labeled so that component reliabilities are ordered as follows:

$$p_1 \leq p_2 \leq \cdots \leq p_n \; .$$

(a) *Series System.* If $h(p) = \prod_{i=1}^{n} p_i$, then

$$I_h(j) = \prod_{i \neq j} p_i$$

and $I_h(1) \geq I_h(2) \geq \cdots \geq I_h(n)$, so that the component with lowest reliability is the most important to the system. This reflects the well-known principle that "a chain is as strong as its weakest link."

(b) *Parallel System.* If $h(p) = \coprod_{i=1}^{n} p_i$, then

$$I_h(j) = \prod_{i \neq j} (1 - p_i)$$

and $I_h(1) \leq I_h(2) < \cdots \leq I_h(n)$, so that the component with highest reliability is the most important to the system. This, too, is intuitively reasonable, since if just *one* component functions, the system functions.

The concept of marginal importance plays a very key role in a computer program called PAFT, [T. Barlow and K. Wood (1982)] for analyzing logic trees. This program calculates the probability of all gate events given the probabilities for basic events. It then calculates the marginal importances of all gate and basic events relative to the Top Event. Given failure rates for basic events and using the marginal importances, the program also calculates marginal occurrence rates for basic events relative to the Top Event. The Top Event occurrence rate for selected time points is then calculated as the sum of basic event marginal occurrence rates.

This program attempts to take optimum advantage of the tree structure for the probability calculation. It neither finds nor uses minimal cut sets for this purpose.

Another approach not based on minimal cut sets is due to S. Arnborg (1978). His algorithm uses the concept of domination in order to achieve "reduced state enumeration." According to Arnborg, it can give good results, but apparently the fault trees on which it is used must be carefully screened for the right characteristics.

3. SYSTEM AVAILABILITY ANALYSIS

In most system reliability analyses, it is necessary to evaluate the effect of maintenance procedures on overall system availability and performance. For example, the following questions are of interest:

1. What is the long run expected time average of the number of system failures?
2. What is the long run expected average of system up (down) times?
3. How often do we expect a specific component to "cause" system failure? (We say that a component "causes" system failure if system failure coincides with that component's failure.)

We will consider two system models of general interest.

MODEL A: Coherent Systems with Separately Maintained Components

For this model, failure-repair processes in different system component positions are assumed to be statistically independent. This is somewhat unrealistic since we also suppose that functioning components continue to operate (and perhaps fail) even when the system is down.

MODEL B: Series Systems Whose Functioning Components Suspend Operation During Repair

This model represents the other extreme relative to MODEL A. In this case functioning components are in "suspended animation" so to speak when the system is down. While in "suspended animation" components do not age and cannot fail.

For both models, we assume continuous failure and repair distributions. Let $\tilde{N}_i(t)$ be the number of times in $[0,t]$ that component i "causes" system failure. Let

$$\tilde{N}(t) = \sum_{i=1}^{n} \tilde{N}_i(t)$$

be the number of system failures in $[0,t]$ where n is the number of system components. We call $\frac{E[\tilde{N}(t)]}{t}$ the expected time average of the number of system failures in $[0,t]$. In general, it will be time dependent. When

$$\lim_{t\to\infty} \frac{E\tilde{N}(t)}{t}$$

exists, we call this the long run expected time average of the number of system failures.

MODEL A: LONG RUN PERFORMANCE FORMULAS

Most of the formulas which answer the previous three questions involve computing the reliability, function, $h(\underline{p})$, discussed in Section 2. Although this function is based on the binary case (components are either working or failed at a specific time), it also plays a crucial role in the dynamic, time dependent case. Under Model A, we have an alternating renewal process for each component position of our coherent system, ϕ . Let component type i have mean life μ_i and mean repair time ν_i . Let

$$x_i(t) = \begin{cases} 1 & \text{if component } i \text{ is working at time } t \text{ ,} \\ 0 & \text{otherwise.} \end{cases}$$

Then the system indicator function is

$$X(t) = \phi[X_1(t), X_2(t), \ldots, X_n(t)]$$

and

$$\begin{aligned} A(t) &= P[X(t) = 1] \\ &= E\phi[\underline{X}(t)] = h[A_1(t), A_2(t), \ldots, A_n(t)] \end{aligned} \qquad (3.1)$$

where $A_i(t)$ is the probability that component i is available at time t . The long run availability is

$$\lim_{t\to\infty} A(t) = h\left[\frac{\mu_1}{\mu_1 + \nu_1}, \frac{\mu_2}{\mu_2 + \nu_2}, \ldots, \frac{\mu_n}{\mu_n + \nu_n}\right] \underline{\underline{\text{def}}} \; A \text{ .} \qquad (3.2)$$

The number of failures in component position i , $N_i(t)$, generates a (delayed) renewal counting process $\{N_i(t) \; ; \; t \geq 0\}$. Let $M_i(t) = EN_i(t)$. It is proved in Barlow and Proschan (1975) that the expected number of system failures in $[0,t]$ caused by component i is

$$E\tilde{N}_i(t) = \int_0^t [h(1_i, \underline{A}(u)) - h(0_i, \underline{A}(u))] dM_i(u) \text{ .} \qquad (3.3)$$

From this result it can be shown that

$$\lim_{t\to\infty} \frac{1}{t} E\tilde{N}_i(t) = [h(1_i, \underline{A}) - h(0_i, \underline{A})]/(\mu_i + \nu_i) \qquad (3.4)$$

where $\underline{A} = (A_1, A_2, \ldots, A_n)$ and $A_i = \mu_i/(\mu_i + \nu_i)$. The long run expected time average of the number of system failures is then

$$\lim_{t\to\infty} \frac{E\tilde{N}(t)}{t} = \sum_{i=1}^{n} [h(1_i,\underline{A}) - h(0_i,\underline{A})]/(\mu_i + \nu_i) \ . \tag{3.5}$$

If $U_1, U_2, \ldots, U_k$ are successive system uptimes, then it can be shown that

$$\lim_{k\to\infty} \frac{E[U_1 + U_2 + \ldots + U_k]}{k} = \frac{h(\underline{A})}{\sum_{i=1}^{n} [h(1_i,\underline{A}) - h(0_i,\underline{A})]/(\mu_i + \nu_i)} \ . \tag{3.6}$$

If $D_1, D_2, \ldots, D_k$ are successive system downtimes, then

$$\lim_{k\to\infty} \frac{E[D_1 + \ldots + D_k]}{k} = \frac{1 - h(\underline{A})}{\sum_{i=1}^{n} [h(1_i,\underline{A}) - h(0_i,\underline{A})]/(\mu_i + \nu_i)} \tag{3.7}$$

the long run average of system downtimes.

<u>Example</u>: Series System

In this case, the long run expected time average of the number of system failures is

$$\left[\prod_{j=1}^{n} \frac{\mu_j}{\mu_j + \nu_j}\right] \sum_{i=1}^{n} \frac{1}{\mu_i} = A \sum_{i=1}^{n} \frac{1}{\mu_i} \tag{3.8}$$

while

$$\lim_{k\to\infty} \frac{E[U_1 + \ldots + U_k]}{k} = \left[\sum_{i=1}^{n} \frac{1}{\mu_i}\right]^{-1} \tag{3.9}$$

and letting $\mu = \left[\sum_{i=1}^{n} \frac{1}{\mu_i}\right]^{-1}$,

$$\lim_{k\to\infty} \frac{E[D_1 + \ldots + D_k]}{k} = \frac{1 - \prod_{j=1}^{n} \frac{\mu_j}{\mu_j + \nu_j}}{\prod_{j=1}^{n} \left[\frac{\mu_j}{\mu_j + \nu_j}\right] \sum_{i=1}^{n} \frac{1}{\mu_i}} = \left(\frac{1 - A}{A}\right) \mu \ . \tag{3.10}$$

MODEL B: LONG RUN PERFORMANCE FORMULAS

Under this model, functioning components suspend operation during repair of non-functioning components. If any component in this series system fails, the remaining components are shut off and remain in suspended animation until the failed component is fixed.

Let $U(t)$ $[D(t)]$ be the cumulated system uptime [downtime] by time t. In Barlow and Proschan (1975), Chapter 7, it is shown that the long run average system availability is, in this case

$$A_{av} \overset{\text{def}}{=} \lim_{t\to\infty} \frac{1}{t}\int_0^t A(u)du = \left(1 + \sum_{j=1}^{n} \frac{\nu_j}{\mu_j}\right)^{-1} . \qquad (3.11)$$

If $\lim_{t\to\infty} A(t)$ exists, then it is the same as (3.11).

The limiting expected time average of the number of system failures caused by component i is

$$\lim_{t\to\infty} \frac{E\tilde{N}_i(t)}{t} = \frac{A_{av}}{\mu_i} . \qquad (3.12)$$

Hence, the long run expected time average of the number of system failures is

$$\lim_{t\to\infty} \frac{E\tilde{N}(t)}{t} = A_{av} \sum_{i=1}^{n} \frac{1}{\mu_i} . \qquad (3.13)$$

The long run average of system uptimes is

$$\lim_{k\to\infty} \frac{E[U_1 + U_2 + \dots + U_k]}{k} = \left(\sum_{i=1}^{n} \frac{1}{\mu_i}\right)^{-1} \overset{\text{def}}{=} \mu . \qquad (3.14)$$

The long run average of system downtimes is

$$\lim_{k\to\infty} \frac{E[D_1 + D_2 + \dots + D_k]}{k} = \mu \sum_{i=1}^{n} \nu_i/\mu_i = \left(\frac{1 - A_{av}}{A_{av}}\right)\mu . \qquad (3.15)$$

Compare (3.8) and (3.13); also (3.9) and (3.14); also (3.10) and (3.15).

Availability of Series Systems with Preventive Maintenance

Most systems are subject to planned maintenance. In calculating system availability, it seems unfair that planned maintenance downtime should count against good system performance. Hence, we define $A_{failure}$ as long run system availability when downtime due to routine maintenance is *not* considered as contributing to system unavailability. Conversely, system failure unavailability is, in the long run, the fraction of time the system is down due to a component or subsystem failure. The following discussion shows how this fraction (or percentage) may be computed.

The Pulse Radiation Source (an X-ray system) is basically a series system of five major subsystems: Marx, waterline, tube, source, and source shield. Each subsystem has a prescribed time between scheduled maintenance and a maintenance downtime (see Table 1). In addition, system failures may cause additional unscheduled maintenance downtime. When scheduled or unscheduled maintenance is performed on a subsystem, the other subsystems are said to be in suspended animation. When the subsystem is maintained or repaired, all subsystems resume normal operation. For the purpose of availability analysis, we assume that maintained or repaired subsystems are "like new."

A table of maintenance downtimes, failure repair downtimes, and subsystems mean times to failure follows (Table 1). Since there are four shots per 8-hour work day, we let 1 system shot equal 2 hours and all times in the table are expressed in hours.

TABLE 1

PULSE RADIATION SOURCE
MAINTENANCE AND FAILURE INFORMATION

Subsystem	Maintenance Frequency	Maintenance Downtime	Mean Time To Failure μ_i	Failure Repair Mean Downtime ν_i
Marx	After 50 shots or *100* h	*8* h	400 shots or *800* h	*8* h
Waterline	After 50 shots or *100* h	*8* h	500 shots or *1000* h	*16* h
Tube	After 50 shots or *100* h	*8* h	200 shots or *400* h	*24* h
	After 5 shots or *10* h	*4* h		
Source/ Shield	After every shot or *2* h	*1* h	---	---

Since the source/shield is repaired after every shot, no mean time to failure is assessed. Since the Marx, Waterline and Tube are periodically maintained, the failure rate is considered constant (one divided by the mean time to failure).

There is a natural 100h operating cycle for the maintenance regime. Since after 10h, 100h, etc. more than one subsystem is serviced, each downtime corresponds to the longest required service time. In a 100h operating cycle, the total maintenance downtime accumulated is

$$T = 1.0[50 - 10] + 4[10 - 1] + 8[1] = 84 \text{ hours.}$$

Let t be calendar time (in units of working hours) and $U(t)$ the cumulated system uptime in calendar time t. Let D_{ir} be the r-th downtime to repair a failure of subsystem i. Then overall system availability, A_o, in the long run is

$$A_o = \lim_{t\to\infty} \frac{U(t)}{U(t) + \frac{U(t)}{100}T + \sum_{i=1}^{k} \sum_{r=1}^{N_i[U(t)]} D_{ir}} \qquad (3.16)$$

since $\frac{U(t)}{100}T$ will be approximately the downtime due to preventive maintenance in calendar time t. In our example, $k = 3$ corresponding to the Marx, waterline and the tube. Hence

$$A_o = \frac{1}{1 + \frac{T}{100} + \sum_{i=1}^{k} \frac{\nu_i}{\mu_i}} = 0.519 \text{ ,}$$

which looks bad! However, if we only count downtime due to failures, then long run availability in this case, called $A_{failure}$, is

$$A_{failure} = \lim_{t\to\infty} \frac{U(t) + \frac{U(t)}{100}T}{U(t) + \frac{U(t)}{100}T + \sum_{i=1}^{k} \sum_{r=1}^{N_i[U(t)]} D_{ir}}$$

$$= \frac{1 + T/100}{1 + \frac{T}{100} + \sum_{i=1}^{k} \frac{\nu_i}{\mu_i}} = 0.955. \qquad (3.17)$$

Let $A_{maint.}$ be the long run system availability with respect to planned maintenance (i.e., the fraction of time the system is *not* down due to planned maintenance). Then

$$A_{\text{maint.}} = \lim_{t\to\infty} \frac{U(t)}{U(t) + \frac{U(t)}{100}\,T} = \left[1 + \frac{T}{100}\right]^{-1}$$

$$= 0.543. \qquad (3.18)$$

Note that from (3.16), (3.17) and (3.18) we have

$$A_o = A_{\text{maint.}} \times A_{\text{failure}} .$$

This is valid assuming failure occurrence is independent of scheduled maintenance.

If we *neglect* planned maintenance downtimes, then from (3.11), we have

$$\lim_{t\to\infty} \frac{E[U(t)]}{t} = \left[1 + \sum_{i=1}^{k} \frac{\nu_i}{\mu_i}\right]^{-1} \qquad (3.19)$$

and from (3.13) the long run expected time average of the number of system failures (neglecting planned maintenance downtimes) is

$$\left[1 + \sum_{i=1}^{k} \frac{\nu_i}{\mu_i}\right]^{-1} \sum_{i=1}^{k} \frac{1}{\mu_i} .$$

In 100 operating hours, we expect

$$100\left[1 + \sum_{i=1}^{k} \frac{\nu_i}{\mu_i}\right]^{-1} \sum_{i=1}^{k} \frac{1}{\mu_i}$$

system failures so that for a planned operating and maintenance cycle of $100 + T$ hours we expect the long run average number of failures per hour to be

$$\lambda = \frac{100\left[1 + \sum_{i=1}^{k} \frac{\nu_i}{\mu_i}\right]^{-1} \sum_{i=1}^{k} \frac{1}{\mu_i}}{100 + T} = 2.377 \times 10^{-3}/h \qquad (3.20)$$

where $\lambda = \lambda_{\text{Marx}} + \lambda_{\text{Waterline}} + \lambda_{\text{Tube}}$ and

$$\lambda_{\text{Marx}} = 6.256 \times 10^{-4}/h$$

$$\lambda_{\text{Waterline}} = 5.005 \times 10^{-4}/h$$

$$\lambda_{\text{Tube}} = 1.25 \times 10^{-3}/h .$$

It is clear that the Tube, the Marx and the Waterline are the most critical subsystems and in that order.

REFERENCES

1. Agrawal, A. and A. Satyanarayana (1982). "An O(|E|) Time Algorithm for Computing the Reliability of a Class of Directed Networks." ORC 82-7, Operations Research Center, University of California, Berkeley, CA 94720.

2. Arnborg, S. (1975). "Reduced State Enumeration - Another Algorithm for Reliability Evaluation." IEEE Trans. on Reliability, Vol. R-27, pp. 101-105, June 1975.

3. Barlow, R. E. and F. Proschan (1975). "Importance of System Components and Fault Tree Events." Stochastic Processes and Their Applications, Vol. 3, No. 2, April 1975.

4. Barlow, R. E., J. B. Fussell and N. D. Singpurwalla, editors (1975). Reliability and Fault Tree Analysis, Society for Industrial and Applied Mathematics, Philadelphia, PA. (An edited conference volume containing many basic papers on fault tree analysis.)

5. Barlow, R. E., P. Chao, Z. Khalil, J. Gerbino and G. S. Subramanian (1980). "Reliability Analysis of the MBS/SXTF System." Operations Research Center Report, University of California, Berkeley, CA 94720.

6. Barlow, R. E., P. Chao, P. B. Candela and M. Verret (1981). "Reliability Analysis of the PRS/SXTF System." Operations Research Center Report, University of California, Berkeley, CA 94720.

7. Barlow, R. E. and F. Proschan (1981). Statistical Theory of Reliability and Life Testing. TO BEGIN WITH, Silver Spring, MD.

8. Barlow, T. and K. Wood (1982). "PAFT - A Computer Program for Fault Tree Analysis," Operations Research Center Report, University of California, Berkeley, CA 94720.

9. Chang, M. K. (1981). "A Graph Theoretic Appraisal of the Complexity of Network Reliability Algorithms." Ph.D. thesis. Operations Research Center, University of California, Berkeley, CA 94720.

10. Rosenthal, A. (1975). "A Computer Scientist Looks at Reliability Computations." In Reliability and Fault Tree Analysis, Barlow, Fussell and Singpurwalla (editors), pp. 133-152. (Discusses the computational complexity of fault tree analysis problems.)

11. Satyanarayana, A. and M. K. Chang (1981). "Network Reliability and the Factoring Theorem." ORC 81-12, Operations Research Center, University of California, Berkeley, CA 94720.

12. Satyanarayana, A. and R. Kevin Wood (1982). "Polygon-to-Chain Reductions and Network Reliability." ORC 82-4, Operations Research Center, University of California, Berkeley, CA 94720.

13. Vesely, W. E., F. F. Goldberg, N. H. Roberts and D. F. Haasl (1981). Fault Tree Handbook, Office of Nuclear Regulatory Research, NUREG-0492, Washington, D.C. 20555.

14. Willie, R. R. (1978). "Computer-Aided Fault Tree Analysis." ORC 78-14, Operations Research Center, University of California, Berkeley, CA 94720.

END

FILMED

2-83

DTIC