This is to certify that the

thesis entitled

Network System Reliability Analysis

presented by

Feng Hsu

has been accepted towards fulfillment
of the requirements for

Master (MS) degree in Operations Research - Statistics

Martin Fox

Major professor

Date 2/5/88

NETWORK SYSTEM RELIABILITY ANALYSIS

By

FENG HSU

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Statistics and Probability

December, 16 1987

ABSTRACT

NETWORK SYSTEM RELIABILITY ANALYSIS

By

Feng Hsu

The reliable performance of a network system for a mission under various conditions is of utmost importance in many industrial, military, and everyday life situations. Main effort of this thesis research have been devoted to develop algorithms and optimization methods for network reliability problems. In chapter 1, two algorithms so-called MDT and DBM methods are presented respectively for qualitative and quantitative evaluation of large networks and the relevant theorems are proved. The three sections in chapter 2 are spent on the optimility problems of network reliability, in which a useful optimal redundancy allocation method is presented and in addition, two useful mesures called the maintenance importance ($MI_{s,t}(X_i)$) and the diagnosis importance ($DI_{s,t}(X_i)$) are presented for deriving optimal policies used in network reliability maintenance and failure diagnosis. Examples and comments are included correspondingly to each of the topics in the thesis.

To my parents, Mr. Ziwei Xu, and Mrs. Shensu Xie,

and a special gift on the 60'th birthday of my father

ACKNOWLEDGMENT

I would like to express my heartly thanks to my major advisor, professor Martin Fox for his greatly appreciated guidance, suggestions, comments, carefully readings and crrections, which has made it possible for me to complete this research project. Thanks are also expressed to my graduate committee members, professor R. Schlueter and professor Soumen Ghosh for thier very helpful comments and suggestions.

CONTENTS

Chapter 1 Network Reliability Analysis

Section 1 Qualitative Analysis of Network Reliability Through Fault-tree and MDT Method

In this section an efficient approach is presented for qualitative analysis of network reliability through FTA (Fault-tree Technique) and finding MCS (Minimal Cut Sets) by the MDT (Modified Dual-graph Transformation) method. It is generally accepted that for the purpose of qualitative analysis for networks (or any complex system) one must establish its Fault-tree by first finding its MCS. Therefore the enumeration of all MCS seperating a specified node pair is a fundamental step in reliability evaluation of network systems which are frequently encountered in communications, electronics, computers and power systems, etc. It is well know in graph theory [1-2] that the algorithms existing for enumerating MCS are not satisfactory so far, while the algorithms for finding MPS (Minimal Path Sets) are much more time efficient. Hence, based on the principle of graph theory, a so called MDT method is presented below for enumerating the MCS of networks by employing any of the existing efficient MPS algorithms [2, 3-5].

1.1 Theorems and Definitions

Definition 1

If, for any two vertices in a graph G, there is at least one path between them, then G is said to be a connected graph.

Definition 2

A network is a connected graph, which has a finite number of vertices and a finite number of edges.

Let G(V, E) be the notation of a network, where V and E are the sets of vertices and edges in the network, respectively.

Definition 3

Consider a network G(V,E) drawn in the plane in such a way that each vertex $v_i \in V$ is represented by a point, each edge is represented by a continous curve connecting the two points which represent its end vertices and no two curves, which represent edges, share any points except in their ends. Such a drawing is called a Plane Network.

Throughout the reminder of this chapter, we assume that all networks discussed are plane networks.

Definition 4

Let G(V,E) be a connected plane graph, K be an edge set. Then $K \subset E$ is called a cutset if it is a minimal seperating set of edges, ie, the removal of K from G interrupts its connectivity, but no proper subset of K has this property. In other words, a cutset seperates G into two connected components.

Definition 5

The network $\tilde{G}(\tilde{V}, \tilde{E})$ is said to be the Dual of a connected graph G(V, E), if there is a 1 - 1 correspondence $f: E \to \tilde{E}$, such that a set of edges T forms a simple circuit in G if and only if f(T) (the corresponding set of edge in $\tilde{G}$) froms a cut set in $\tilde{G}$.

Definition 6

A point v_i is a Boundary Point if there exists a curve connecting v_i to the exterior of the convex hull of the network which does not intersect the network.

Definition 7

An edge of a network G is a Boundary Edge if each point on the edge is a boundary point.

Definition 8

A cycle C of a network G is a Boundary Cycle if at least one boundary edge exists in C.

Theorem 1 Let G be a network and x be a boundary vertex of G and assume that $\exists$ at least one cycle through x. Then, $\exists$ at least one boundary cycle through x.

Proof Suppose the theorem is false. Take an arbitary cycle C_0 through x and let e_0 be an edge in C_0. Then $\exists$ a point $y_0 \in e_0$ which is not a boundary point. Every curve connecting y_0 to the exterior of the convex hall of the network intersects the network.

Then, $\exists$ a cycle $C_1 \neq C_0$ through x which includes e_0 and a subset of this set of intersections. This construction can be repeated, each time creating a new cycle by taking e_n an edge of C_n and C_1, C_2, $\ldots C_n$ distinct.

Thus, $\exists$ an infinite sequence of cycles leading to a contradiction.

Definition 9

Let $G(V, E)$ be a plane network and let $\tilde{G}(\tilde{V}, \tilde{E})$ be a dual of G. If s and t are two boundary vertices of G, $(s, t \in V)$ then, by drawing two half curves from s and t respectively, the exterior of the network G is divided into two parts.

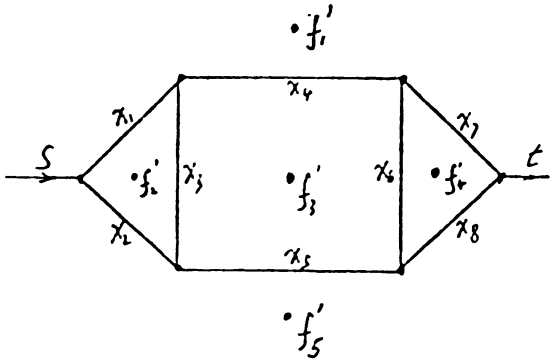
Let s' and t' be points, one on each side of the two parts of the exterior of G given above, and let $\tilde{G}(\tilde{V}, \tilde{E})$ be a dual of G with $s' \in \tilde{V}$. Form a network $G^*(V^*, E^*)$ as as follows:

1 $V^* = \tilde{V} \cup \{t'\}$

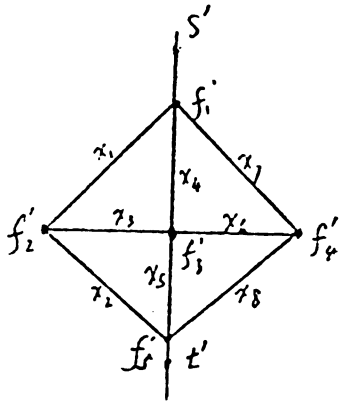
2 Find a boundary cycle through s' with a boundary edge $x \overset{e'}{-} s'$, $e' \in \tilde{E}$ and replace such $x \overset{e'}{-} s'$ by $x \overset{e'}{-} t'$, so that the new graph is a plane graph.

3 Repeat 2 for as many cycles of $\tilde{G}$ as possible without destroying any path from s' to t' which already existed. The new network G^* is called a Modified Dual Network of G.

Figure 2 G^* is a modified dual graph of G (Figure 1):

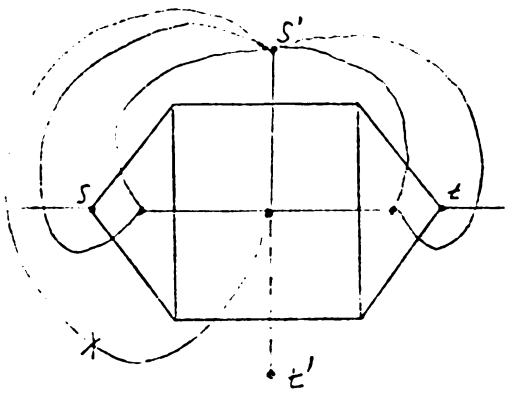


G Figure 1

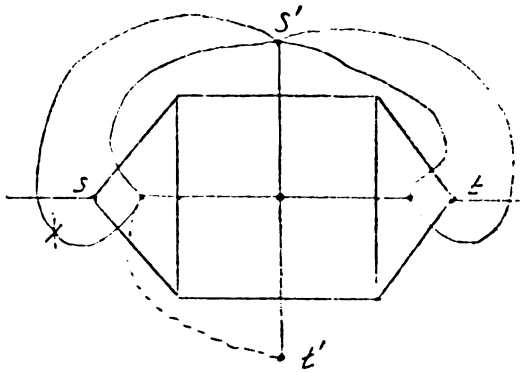


G^* Figure 2

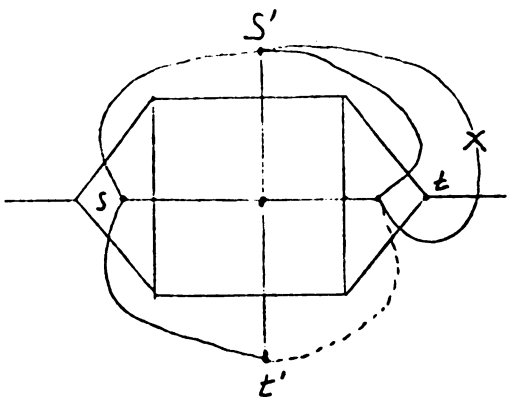
Construction of the above modified network G^* from G is illustrated in Figure 3.



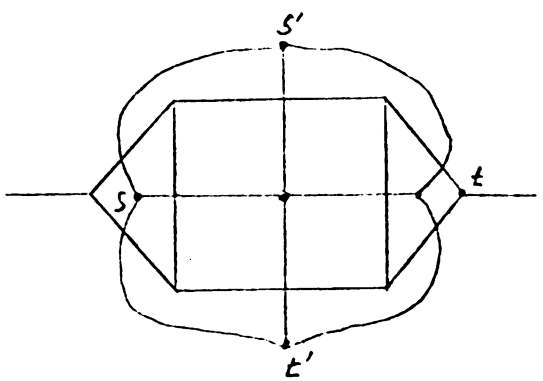
(a) $\tilde{G}$ and G with 1st stage (dotted)



(b) 2nd stage dotted



(c) 3rd stage



(d) G and G^*

Figure 3

In the following theorem a notation $\equiv$ will be used to represent the equivalency between any two edge sets.

Theorem 2.

Let S be the set of all MCS seperating s and t in the network G. Let G^* be a modified dual network of G and P' be the set of all MPS from s' to t' in G^*. Then we have $S \equiv P'$.

Proof

Let S be the set of all MCS's and $\tilde{T}$ be the set of all cycles in a usual dual network $\tilde{G}$ of G.

Thus, each $S_k \in S$ is equivalent to some $r_k \in \tilde{T}$, That is, $S_k \equiv r_k \in \tilde{T}$, where r_k passes through at least one vertex v' with $v' \in V$ and v' in the interior of G.

Morever, S and t are also seperated by r_k. see (Figure 4).

If we suppose that M is the set of all such $r_k \in \tilde{T}$, we then have:

$$M \equiv S \qquad (1)$$

Let P'_k to be any minimal path of G^* which goes from s' to t'. If s' and t' are merged into one point say s', then any path from s' to t' will become a cycle of $\tilde{G}$ through s' and all edges containing s' are edges contained in each MPS of G^*. That is P'_k is equivalent to a cycle r_k of $\tilde{G}$.

Thus, $P'_k \equiv r_k \in \tilde{T}$ (2)

Since $r_k \equiv S_k \in S$, it follows that

$\Rightarrow \quad P'_k \in M$

For $\forall\ P'_k \in P'$, it follows that

$P' \subseteq M$ (3)

On the contrary, if each $r_k \in M$ and s' and t' are the source and sink of G^*, respectively, then we can break each cycle r_k such that both s' and t' are to be connected by edges of each r_k without disturbing the 1 - 1 correspondence of edges between E and E' as given in Definition 5, and meanwhile, either s or t of G is no longer inside any cycle r_k of $\tilde{G}$.

Hence, r_k is a minimal path of G^* by which s and t is seperated.

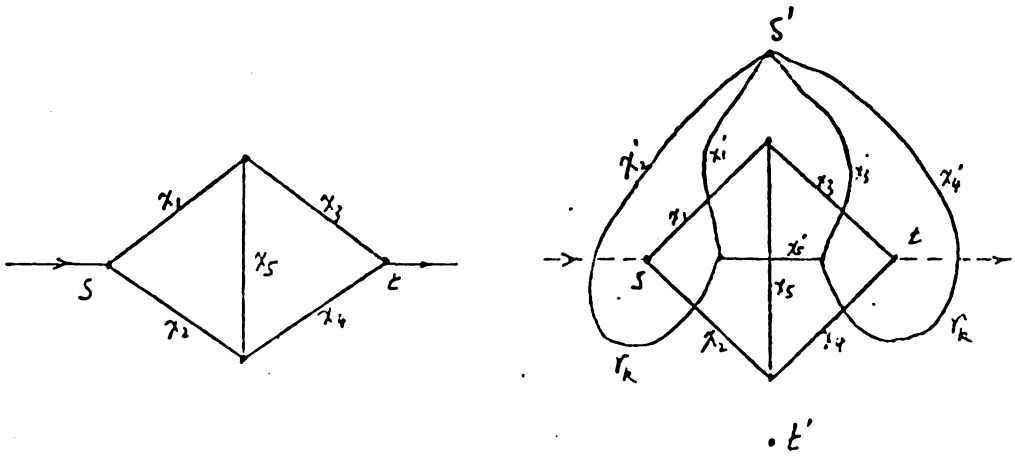
Hence, $r_k \in M$

and $P'_k \in P' \Rightarrow r_k \equiv P'_k \subseteq P$

$\Rightarrow P'_k \in M \subseteq P'$

So, $M \subseteq P'$ (4)

is proved to be true. the theorem then follows from (1), (3), (4) #



(Figure 4)

Definition 10 A Fault-tree is a system logical diagram composed of logical operational symbols which explicitly shows the logical interactions between each basic components of the system, and gives all the possible system failure imformation that may be existed in the system. [6-10]

By applying the above Theorems, the problem of finding a given network's MCS simply becomes a problem of enumerating the minimal path of its modified dual graph. Therefore, any of the efficient path-finding algorithms can be used to find the minimal cut sets of a given network and, then, the Fault-tree of such a network can be easily obtained by simply plugging all the MCS into a tree with each of its leaves represent a possible failure event, which equivalently corresonding to a MCS. (see Figure 5)

1.2 Algorithm and Examples

Algorithm

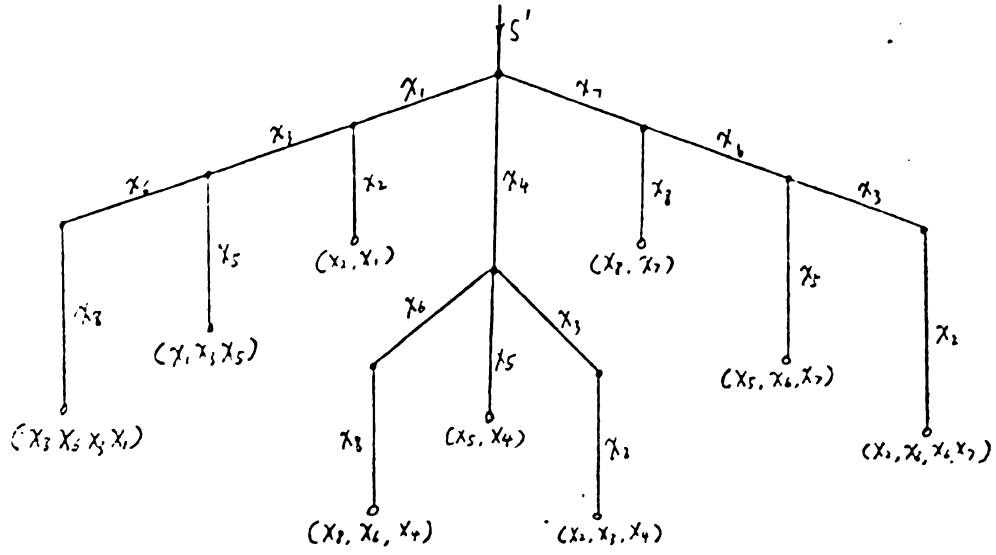
(1) Obtain the modified dual network G^* of G.

(2) Enumerate all the path sets P' between s' and t' in G^*. and then get the MCS S of G by theorem 2, ie, S = P'.

(3) Drawing the fault-tree of the network by applying all the MCS obtained in step (2).

The construction of fault tree stated in step (3) can be very time consuming if a large complex network is encountered. In recent years much effort has been spent to develop algorithms and techniques, for computer-aided automatic synthesis of fault trees of large networks or any complex systems, and several algorithms have been presented [11-13]. However, for a network with small number of components, the corresponding fault tree can be easily drawn. That is, as is shown in Figure 5 we can first start out from the source (vertex s') and let s' be the 'root' of the tree, and then draw a tree rooted at s' by representing all the edges in each of the MCS as an individual path from s' to a 'leaf' (ie, vertex t') respectively.

For example, the MCS and the fault-tree (Figure 5) of network in Figure 2 is obtained by using above algorithm:



(Figure 5)

The above fault-tree is a simple version which uses node and 'branches' instead of using logical OR , AND symbols respectively. A formal fault-tree of the same network is given below. see (Figure 5'):

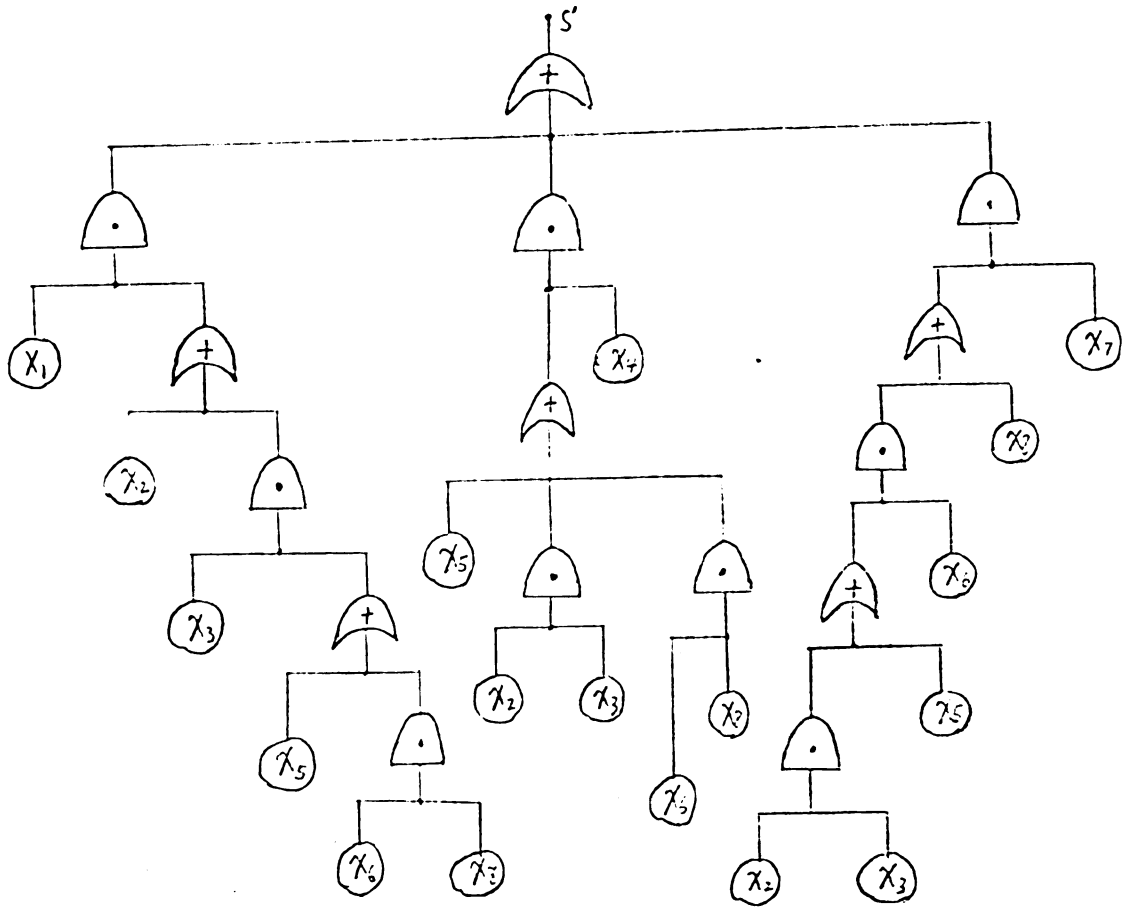
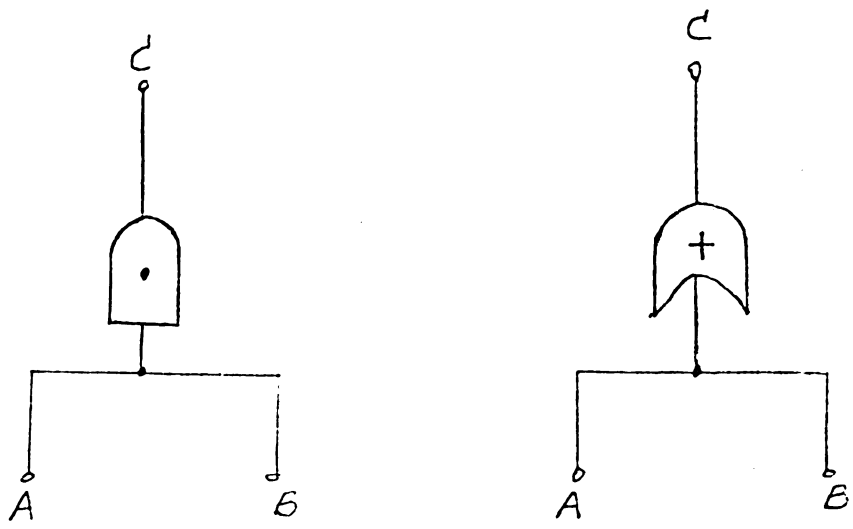


Figure 5'

The above fault tree (Figure 5') is logically equivalent to the simple version (Figure 5), the only difference is that Figure 5' uses the logical AND, OR symbols (see Figure 5'') to represent the relationship between components instead of using nodes and 'branches' in Figure 5. Since the formal fault tree is easier to

construct by using a computer, it is more widely used than the simple version.



(A ∩ B = C)

AND gate

(A ∪ B = C)

OR gate

Figure 5"

Section 2 Quantitative Analysis by DBM Method

A meaningful masure of the reliability and availability of a network system is the terminal reliability between a given pair of vertices, defined as the probability that there exists at least one path between these two nodes. This parameter depends on both the network topology and the reliability of all elements composing the network and is relevant not only for network analysis but also for synthesis of reliable networks.[14]

In recent years much effort has been spent to develop algorithms [15-17] for computating the network reliability. However the difficulty is that the complexity of the algorithms remains exponential with the increase of number of elements, therefore the need remains to widen the class of tractable networks.

It is well known that for the quantitative evaluation of system failure probability (reliability), one must transfer the structure function (a Boolean S-O-P form) into the disjoint-S-O-P (sum of product) form. J. M Cargal has shown [18] that the crucial problem of determining the network reliability is to assess the desired probability with much less effort than is currently extended and higher algebra is most likely the avenue for such a task. To attain the same object by trying the DBM - a

Disjoint Boolean Manipulation method in this section, we have the very helpful approach of calculating the network reliability. The substance of the DBM method is the minimization principle of a Boolean function in disjoint sense. In other words, the advantage of using the DBM method is that it minimizes the number of Boolean terms of sytem's logical expression while eliminating any logical redundencies. The ordinary minimization principle of Boolean functions is to minimize the total number of symbols of intersection and union manipulations. Using DBM we can not only simplyfy the computation of network reliability but also make the design of switching circuits much efficient and easier.

For example, by the ordinary Boolean principle $A \cup B$ is minimal for an OR operations with two inputs, but $A \cup A'B$ or $B \cup B'A$ is much more convenient for probability computations. The difference between these principles seems to be very little when n is small, but it may become very large eventually when n is large. In fact the expression of system reliability given by the sum of events $X_1 + X_2 + \ldots + X_n$ (these X_i can express a MCS or MPS in the network or any compex system that can be constructed logically by Fault-tree technique) may contain 2^n-1 terms, but that of disjoint sum event $X_1 + X_1'X_2 + \ldots + X'_1 \ldots X'_{n-1}X_n$ contains only n terms.

Rosenthal has shown [19] that computing the probability of a desired event (system success or failure) for an arbitry system structure function, like computing the reliability of a general network, is NP difficult, ie, the amount of computation increases

exponentially with the number of components in the network.

2.1 DBM Method and Its Application

(1) Nomenclature

$UB(X)=UB(X_1,\ldots X_n)$ denotes a usual Boolean expression, an event-expression for system success or failure that contains only Boolean variables and operations.

$DB(X)=DB(X_1,\ldots X_n)$ denotes a disjoint Boolean expression that can be used directly for computing the system reliability.

X_i denotes an event.

x_i denotes the indicator of X_i.

(2) Assumptions:

1. A network system S which consists of n-independent elements could be expressed as:

$$S = \{e_1, e_2, \ldots e_n\}.$$

We assume that each of these elements has two states:

functioning and failure. The state of e_i could be characterized by the binary variable x_i as following:

$$x_i = \begin{cases} 0 & \text{(failure)} \\ 1 & \text{(functioning)} \end{cases}$$

Thus, x_i is the indicator of the event that the ith component functions.

Definition 11 The system structure function denoted by $\phi(x_1, x_2, \ldots x_n)$ is a Boolean function which is the indicator of the event of system functioning.

Definition 12 a system failure function denoted by $\Psi(x_1, x_2, \ldots x_n)$ is a Boolean function which is the indicator of the event of system failure.

Let x_{ri} be the indicator of the event that the ith component of the rth MPS is functioning. The x_{ri} are $x_1, \ldots, x_n$ and, for $r \neq s$, it is possible to have $x_{ri} = x_{sj}$ for some i and j.

If we enumerate the MCS and MPS of S, the structure function and the failure function will be written, respectively, as:

$$\phi(x) = \phi(x_1, x_2, \ldots . x_n) = \sum_r \prod_i x_{ri} \qquad x_{ri} \in \text{rth MPS}$$

$$\Psi(x) = \Psi(x_1, x_2 \ldots . x_n) = \sum_q \prod_j x_{qj} \qquad x_{qj} \in \text{qth MCS}$$

Let $P = (p_1, p_2, \ldots p_n)$ denote the reliability vector of components

$e_1, e_2, \ldots e_n$. Because of the assumption that the components are independent of one another, the domain of ϕ and of Ψ are extended by setting:

$$\phi(P) = E[\phi(x)]$$

$$\Psi(P) = E[\Psi(x)]$$

Thus,

$\phi(P) = \phi(p_1, p_2, \ldots . p_n) = \sum_r \prod_i p_{ri}$ is the probability of the system functioning.

$\Psi(P) = \Psi(p_1, p_2 \ldots . p_n) = \sum_q \prod_j p_{qj}$ is the probability of the system failing.

Where, $\phi(P) = 1 - \Psi(P)$ since $\phi(x) = 1 - \Psi(x)$.

2 In graphs only arcs may be faulty. Nodes have 0 failure probability. The reliability of each arc is given.

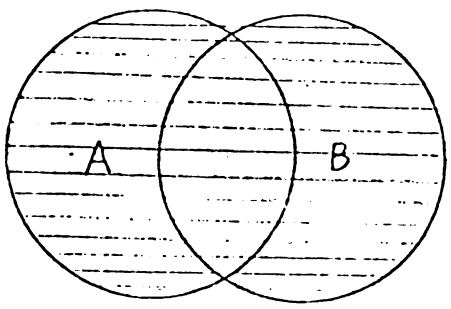
3. Component (arc) failures are independent.

4. Given a network and two of its particular nodes, s and t, all simple paths from (or all minimal cut sets seperating) s and t are known or can be found.

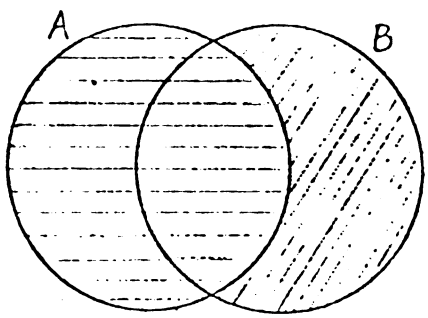
(3) **Rules & comparision between DBM and usual operations (UBM):**

Logical interaction	OBM rules	DBM rules
1. AND operation	AB	AB
2. OR operation	A+B	A+A'B or B+B'A (Fig 6)
3. NAND operation	(AB)'= A'+B'	(AB)'= A'+AB' or (AB)'= B'+BA' (Fig 7)
4. NOR operation	(A+B)'= A'B'	(A+B)'= (A+A'B)'= A'B' or (A+B)'= (B+B'A)'= A'B'

Using Venn diagrams the OBM and UBM rules can be interpreted as following:



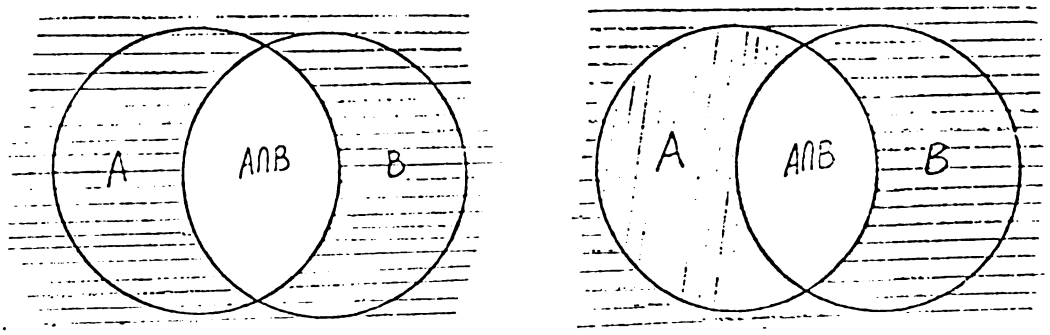
A+B by UDM



A+B by DBM

($S_a + S_b - S_{ab}$ shows the relation of area)

(Figure 6)



(AB)' by UDM (AB)' by DBM

(Figure 7)

According to the addition and multiplication theorems of probability, it is easy to see that:

a. The rules of the two types are equivalent completely for the calculation of probability.

b. Althrough the expressions of DB algebra are non-unique, they are equivalent to each other.

c. Using DBM to decompose the logical structure of system, the disjoint Boolean function can be obtained directly from the logical structure of the network (or any systems).

d. By extending, absorbing and summarizing the structure functions using the logical identities such as:

$$A + AB = A \qquad AB + AB = AB$$

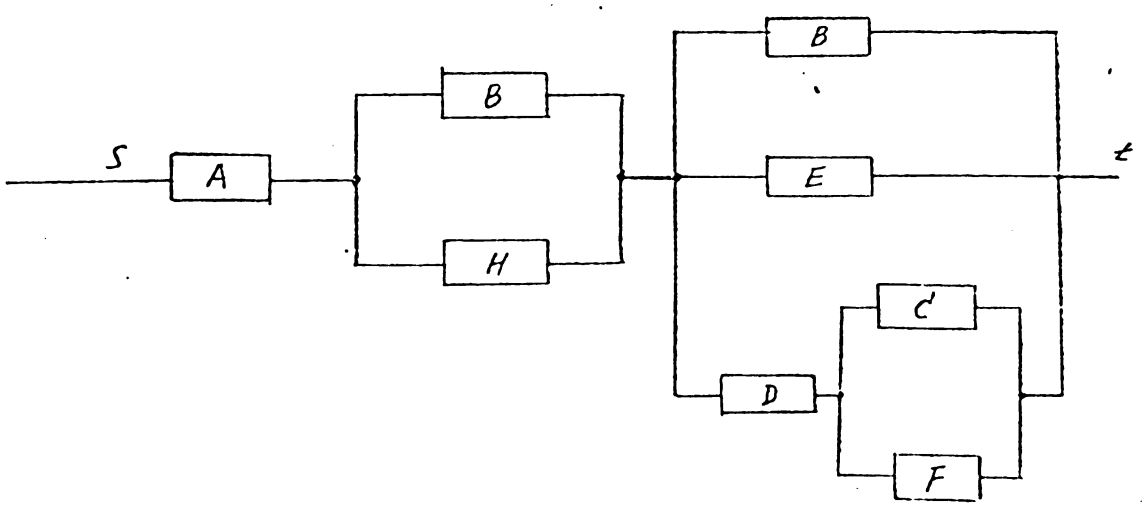
$$AA = A \qquad AB + AB' = A \qquad AA' = 0$$

the disjoint S-O-P expression of the system can be eventually obtained.

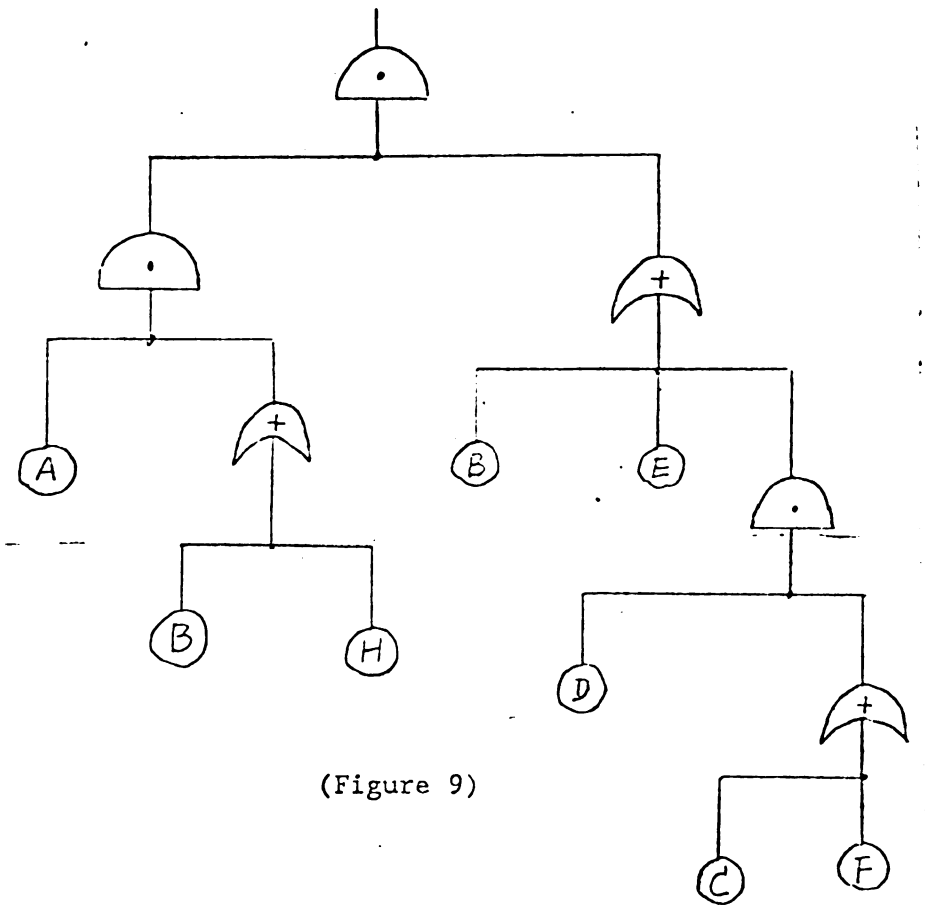
2.2 Algorithm and Examples

(4) Examples:

A given network is shown in Figure 8 and its fault-tree shown in Figure 9:



(Figure 8)



(Figure 9)

1. Using Fratta's algorithm: [20]

$$OB(X) = ABB + ABE + ABCD + ABDF + AHB + AHE + AHDC + AHDF$$
$$= AB + AHE + AHDC + AHDF.$$

Simplify the above structure function into the disjoint-S-O-P form by the following steps:

a. $(AB)'(AHE + AHDC + AHDF) = (A' + B')(AHE + AHDC + AHDF)$
$$= B'AHE + B'AHDC + B'AHDF$$

b. $(B'AHE)'(B'AHDC + B'AHDF) = (B + A' + H' + E')(B'AHDC + B'AHDF)$

$= E'B'AHDC + E'B'AHDF$

c. $(E'B'AHDC)'(E'B'AHDF) = (E + B + A' + H' + D' + C')E'B'AHDF$

$= C'E'B'AHDF$

Finally, the disjoint S-O-P form is:

$$UB(X) = AB + B'AHE + E'B'AHDC + C'E'B'AHDF \quad (1)$$

2. Using the DBM method:

The disjoint S-O-P form can be obtainted directly (in one step) from the system structure diagram (in either network or Fault-tree form):

$$DB(X) = A(B+B'H)[B+B'E + B'E'D(C+C'F)]$$

$$= ABB + ABB'E + ABB'E'DC + ABB'E'DC'F + AB'HB$$

$$+ AB'HB'E + AB'HB'E'DC + AB'HB'E'DC'F$$

$$= AB + AB'HE + AB'HE'DC + AB'HE'DC'F \quad (2)$$

Obviously, (1) = (2)

It is easy to see that in the DBM approch the amount of computation is reduced significantly, and the computer implementation of calculating the reliabilities between any given pair of boundary vertices of a network, can be more easily performed in a shorter & unique program. Suppose the element reliabilities of the given network are as follows:

$p_a=0.9$ $p_b=0.7$ $p_c=0.8$ $p_d=0.97$

$p_e=0.95$ $p_h=0.7$ $p_f=0.8$

Then the system reliability of the network is:

$$R_s = 0.9*0.7 + 0.9*0.3*0.7*0.95 + 0.9*0.3*0.7*0.05*0.97*0.8$$
$$+0.9*0.3*0.7*0.05*0.97*0.2*0.8$$
$$= 0.818 \qquad \#$$

Combining section 1 and section 2 we can form an algorithm for reliability evaluation (quantitatively & quantitatively) of the network systems:

Algorithm

1. Apply the MDT algorithm to given network and get G^* of G and all the MCS in the system.

2. Construct the system structre function $\phi(x)$ using all the MCS information obtained in step 2 (or by applying the DBM method).

3. Reduce $\phi(x)$ first by general boolean identities ie:

$X+XY=X \quad XY+XY=XY \quad XY+XY'=X \quad XX=X \quad XX'=0$.

4. Apply the DBM rules directly to the simpfied $\emptyset(x)$ from 3.

5. Any more redundent terms? if yes, go to 4. if no go to 6.

6. Input all the failure probabilities of every component X_i.

7. Calculate the reliability directly for the whole system and then stop.

Chapter 2 Some Optimal Policies for System Reliability

Section 1 Redundency Policy for Series-paralell System Under Resource Constraints Using L-M Method

This section is devoted to discuss the problem of allocating redundent components subject to resource constraints so as to optimize some measure of system performance. Since the 1950s, many models and solution procedures have been developed for various of these problems. The solution procedures can be divided into two categries: heuristic and exact methods. While useful heuristics are easy to implement and have modest computational requirements, they do not guarantee optimal solutions. All known methods which guarantee optimal solutions are enumerative, such as integer and dynamic programming. These exact methods have computational requirements that grow exponentially with the size of conponents. Tillman & Frank [21] provide an excellent survey of research on the problem.

The optimization problem presented here is a rather simple and useful method in practice, which gives approximately the optimal solution for minimizing the whole system cost under the constraint that the whole system must meet the designed reliability requirement. This approach rather than just

maximazing the system reliability subject to element constraints, is most commonly used [22-24].

1.1 Problem Formulation and Solution

Suppose there are n different stages (subsystems) conected serialy with each other, where the ith (i=1.....n) subsystem consist of m_i identical conponents connected paralelly with each other. Assume also the compoent failures are independent. Given that the failure probability is q_i for any element in the ith subsystem and that its cost is c_i, the problem is to find an optimal allocation of the elements for each subsystem which will minimizing the total system cost under the constraint that the whole system must be at least as reliable as is required by the designer. Thus let R_0, be the minimum reliability requirement the whole system must meet. That is:

Evaluate the decision variables m_i such that

$$\text{Minimizes} \quad f(m_i) = \min\left\{ \sum_{i=1}^{n} C_i m_i \right\} \qquad (1)$$

$$\text{Subject to} \quad R = \left| \prod_{i=1}^{n} \right| (1 - q_i^{mi}) \geq R_0 \qquad (2)$$

and $m_1, m_2, \ldots m_n$ are integers.

where c_i, q_i and R_0 are given. Since, the higher the system reliability required, the more element redundencies need to be added on to each of the subsystems. so the solution $(m_1, m_2, \ldots m_n)$ which minimizes the total cost must also meet the requirement that $R \geq R_0$. Howover, a little increase in R_0 will result in a big cost for the system. Therefore, to make the problem simpler we can get an approximation for the problem by making the following two assumptions:

a. Use equality ($=$) in (2) rather than inequality ($\geq$)

b. Take each M_i as a continous variable (vector) and the logrithm in both sides of (1) and round off solutions at the very end.

c. Following assumptions (a) & (b) we can easily apply the Lagrange multiplier method to solve the problem.

Let $H(m_1, m_2 \ldots\ldots m_n, \gamma)$

$$= \sum_{i=1}^{n} C_i M_i + \gamma \left(\sum \ln(1 - q_i^{mi}) - \ln R_0 \right)$$

$$\frac{\partial H}{\partial m_i} = C_i + \gamma \left(\frac{-q_i^{m_i} \ln q_i}{1 - q_i^{m_i}} \right) = 0 \quad (i = 1, 2, \ldots n)$$

$$q_i^{m_i} = \frac{C_i}{C_i + \gamma \ln q_i} \qquad (3)$$

$$\text{Then, } m_i = \ln \left(\frac{C_i}{C_i + \gamma \ln q_i} \right) / \ln q_i \qquad (4)$$

From (2) & (3) and (4) we get:

$$\gamma^n - \{ R_0 \, |\overline{}|_i (C_i + \gamma \ln q_i) / |\overline{}|_i \ln q_i \} = 0 \qquad (5)$$

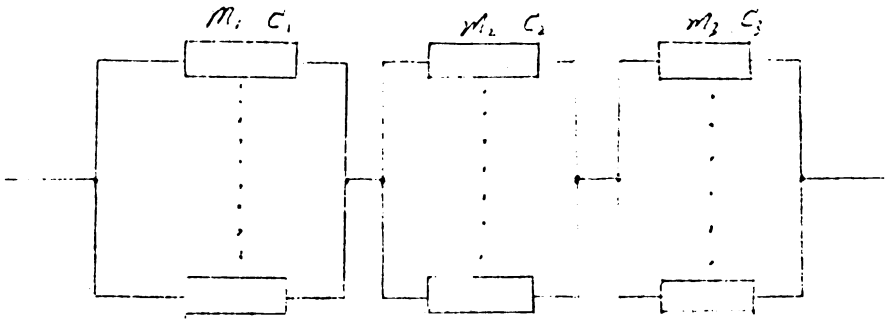
we can get the solution γ_0 of (5) and then by (4) we get:

$$m_i = \ln \left(\frac{C_i}{C_i + \gamma_0 \ln q_0} \right) / \ln q_i \qquad (6)$$

1.2 Example

The system is given as shown in (Figure 10), it consists of three subsystems, and each of which is composed of m_1, m_2, m_3 different elements parallel. (components in a

subsystem assumed all the same)



(Fig 10)

The failure probabilities of the three kinds elements are assumed to be 0.02, 0.01, 0.03 and their cost are \$3000, \$4000, \$5000 respectively. The best policy is sought for allocating those elements so as to minimize the total cost under the condition that the system reliability must no less than 0.99.

Solution: $q_1 = 0.02$, $q_2 = 0.01$, $q_3 = 0.03$. $c_1 = 3$ $c_2 = 4$, $c_3 = 5$ (unit)

find (m_1, m_2, m_3)

$$\text{st.} \qquad \min\{3m_1 + 4m_2 + 5m_3\}$$

$$\text{subject to} \qquad \prod_{i=1}^{3} (1 - q_i^{m_i}) \geq 0.99$$

By (5),

$$\gamma^3 + 303\gamma^2 - 297\gamma + 94 = 0$$

solving the equation yields $\gamma_0 = -99.806$.

Then, by (6),

$m_1 = 1.245$, $m_2 = 1.032$, $m_3 = 1.215$

Rounding up, we finally get:

$$m_1 = m_2 = m_3 = 2$$

system reliability is :

$[1-(0.02)^2][1-(0.01)^2][1-(0.03)^2] = .9986 > .99$

So $(m_1, m_2, m_3) = (2, 2, 2)$ is the optimal policy for allocating the three different kinds of elements to the system, which yields a total cost of $(3 + 4 + 5)*2 = 24$ (1000 dollars) to the system.

This can be verified as following that reduction any of the elements m_i will violate the system reliability constraint :

(1) If $m_2 = 1$, system reliability is

$[1-(0.02)^2][1-(0.01)^1][1-(0.03)^2] = 0.988 < 0.99$

(2) If $m_3 = 1$, system reliability is

$[1-(0.02)^2][1-(0.01)^2][1-(0.03)^1] = 0.969 < 0.99$

On the other hand, suppose we want the total cost be \$22 which is less than \$24. let $m_1 = 3$, $m_2 = 2$ $m_3 = 1$.

Then the system reliability becomes

$[1-(0.02)^3][1-(0.01)^2][1-(0.03)^1] = 0.9699 < 0.99$

which violates the R_0 constraint. This has verified that the privous solution for the problem is truly a optimal solution.

Section 2 Some Optimal Polices Used for Reliability Maintenance And Failure Diagnosing Without Cost Constraint

Based on the results from chapter 1, once we have the disjoint S-O-P form of system failure function (or success function), we can also develop some useful mesures which can be used in determining the optimal polices for system reliabiliy maintenance and repairing diagnosis. In this section two measures that may be useful in network reliability maintainence and failure diagnosis will be discussed [25-26].

2.1 Maintenance Importance: ($MI_{s,t}(X_i)$)

For a functioning network system, the primary interest for the system operator and the maintenance engineer is to know which one of the components among the system is more important to the performance of the whole system. In other words, we want to know the optimal policy

which gives a best time scheduling (or priorities) for each of the components in a routine maintenance project which will substantially reduce the risk of system break-downs. However, without considering the cost, the main effort is to determine the dependencies of the network reliability on the reliability of a individul element. Therefore it is quite natural that the maintenance importance of the components should be defined as follows:

$$MI_{s,t}(X_i) = Pr(\text{ system failed } | \text{ component } X_i \text{ failed })$$

This is the conditional probability of system failure (no path from s to t exists anymore) given that component X_i has failed. As already described in chapter 1 (section 2), $MI_{s,t}(X_i)$ is the expectation of the failure function given that $x_i=0$, ie,

$$MI_{s,t}(X_i) = E[\Psi(x) \mid x_i=0] \qquad (2)$$

Similarly, $\phi(x) = 1 - \Psi(x)$ so that,

$$E[\phi(x) \mid x_i=0] = 1 - MI_{s,t}(X_i) \qquad (3)$$

2.2 Diagnosing Importance: ($DI_{s,t}(X_i)$).

For repairing a failed network system, one must first find out which one of the elements is most likely to be the failed one which has caused the break-down of the whole system. In other words, in order to fix the system in a lowest cost and shortest time, one must know the optimal priority ordering for cheking those components. Therefore, a so-called Diagnosing Importance $DI_{s,t}(X_i)$ should be defined as:

$$DI_{s,t}(X_i) = Pr\{\text{component } X_i \text{ failed} \mid \text{system failed}\} \qquad (4)$$

This is the conditional probability that component X_i failed given the whole sytem has failed already. Likewise, we can derive $DI_{s,t}(X_i)$ by Bayes' rule and conditioning to the failure of component x_i and finally conditioning to system failure, ie:

$$\begin{aligned} DI_{s,t}(X_i) &= Pr\{X_i \text{ failed, system failed}\} / Pr\{\text{system failed}\} \\ &= P\{\text{system failed} \mid X_i \text{ failed}\}P\{Xi \text{ failed}\} / \Psi(p) \\ &= (1-p_i)MI_{s,t}(X_i) / \Psi(p) \qquad (5) \end{aligned}$$

2.3 Example

For the network system given in chapter 1 (Figure 8) the component's reliabilities are given as:

$p_a = 0.9$ $\quad p_b = 0.7$ $\quad p_c = 0.8$ $\quad p_e = 0.95$

$p_f = 0.8$ $\quad p_d = 0.97$ $\quad p_h = 0.7$.

From (3), $MI_{s,t}(X_i) = 1 - E[\phi(x) \mid x_i=0]$ and,

$$\Psi(x) = 1 - \phi(x)$$

So, $\Psi(P) = 1 - \phi(P) = 1 - (p_a p_b + p_a q_b p_h p_e + p_a q_b p_h q_e p_d p_c + p_a q_b p_h q_e p_d q_c p_f)$

$= 1 - 0.818 = 0.182.$

Let $\Psi(x_i=0) = E[\Psi(x) \mid x_i=0]$, which is the expectation of the system failure function given that component X_i is failed.

Then, from (2) we get:

$MI_{s,t}(A) = \Psi(x_a = 0) = 1 - 0 = 1$

$MI_{s,t}(B) = \Psi(x_b = 0) = 1 - (0.9*1*0.7*0.95 + 0.9*1*0.7*0.05*0.97*0.8 + 0.9*1*0.7*0.05*0.97*0.2*0.8)$

$= 1 - (0.5985 + 0.02444 + 0.004888)$

$= 0.372$

$$MI_{s,t}(C) = \Psi(x_c = 0) = 1 - \{\ 0.9*0.7 + 0.3*0.9*0.7*0.95 + 1*0.05*0.3*0.9*0.7*0.97*0.8\ \}$$

$$= 1 - \{\ 0.63 + 0.17955 + 0.0073332\}$$

$$= 0.183$$

Similarly,

$MI_{s,t}(D) = \Psi(x_d = 0) = 0.1905$

$MI_{s,t}(H) = \Psi(x_h = 0) = 0.37$

$MI_{s,t}(E) = \Psi(x_e = 0) = 0.358$

$MI_{s,t}(F) = \Psi(x_f = 0) = 0.183$

From (5) and the above results we get:

$$DI_{s,t}(A) = MI_{s,t}(A)(1 - p_a) / \Psi(p) = 1*(1-0.9)/0.182 = 0.549$$

$$DI_{s,t}(B) = MI_{s,t}(B)(1 - p_b) / \Psi(p) = 0.372(1-0.7)/0.182 = 0.613$$

$$DI_{s,t}(C) = MI_{s,t}(C)(1 - p_c) / \Psi(p) = 0.183(1-0.8)/0.182 = 0.201$$

Similarly,

$DI_{s,t}(D) = MI_{s,t}(D)(1 - p_d) / \Psi(p) = 0.0314$

$DI_{s,t}(E) = MI_{s,t}(E)(1 - p_e) / \Psi(p) = 0.0984$

$DI_{s,t}(H) = MI_{s,t}(H)(1 - p_h) / \Psi(p) = 0.609$

$DI_{s,t}(F) = MI_{s,t}(F)(1 - p_f) / \Psi(p) = 0.201$

From the above results:

(a) If the system is functioning, the maintenance importances for the components are listed as follows:

(ie, list of $MI_{s,t}(X_i)$ from largest to smallest)

1st :	component A	2nd :	component B
3rd :	component H	4th :	component E
5th :	component D	6th :	component C or F

Obviously, for the functioning system the optimal maintenance scheduling for each of the components should follow the priorities as below:

1st $\max_i\{MI_{s,t}(X_i)\}$ ie, component A

After finishing examination and repair of component A the next one will be:

$\max_i\{MI_{s,t}(X_i)$, given A is ok$\}$ which can be evaluated as $\max_i\{\Psi(x_i=0, x_a=1)\} = 0.3025$

ie, component B

After finishing component B, and so on we finally get

3rd	component E	4th	component H
5th	component D	6th	component C or F

It can be noticed that this actual optimal maintenance scheduling for the given system is just slightly different from the list of maintenance importances obtained before.

(b). If the system has failed, the diagnosis importances for each of the components are listed below:

(ie, list of $DI_{s,t}(X_i)$ from largest to smallest)

1st :	coponent B	2nd :	component H
3rd :	component A	4th :	component C or F
5th :	component E	6th :	component D

From above list, we can decide that component B (with largest $DI_{s,t}(X_i)$) should be the one to check first. However, the optimal diagnosis policy for checking out the failed component is more complicated than that of the maintenance cases described above, and which will depend on different conditions of the system.

Suppose that on checking component B it was found that it did not fail. Then the next one (or the only one for this particular example) to check is:

$\max_i \{ DI_{s,t}(X_i, \text{ given B is ok}) \}$ which is obviously component A . That is, for this particular example if system has failed and first found out B did not failed, then we can immediately decide that component A must be the failed one which has causted the failure of the system.

However, for diagnosing the failure of a general network, if we first find out component X_j did not failed, then the next component needs to check will be:

$$\max_i \{ DI_{s,t}(X_i, \text{ given } X_j \text{ is ok}) \} \qquad (i \neq j)$$

$$= \max_i \{ DI_{s,t}(X_i \mid x_j=1) \}$$

$$= \max_i \{ \Pr(X_i \text{ failed} \mid \text{system failed}, X_j \text{ ok}) \}$$

Let F_{xi} denote the event that component X_i failed

F_s denote the event that system failed

Then, $DI_{s,t}(X_i \mid x_j=1)$

$$= \Pr\{ F_{xi} \mid F_s, x_j=1 \}$$

$$= \frac{\Pr\{ F_{xi}, F_s, x_j = 1 \}}{\Pr\{ F_s, x_j = 1 \}}$$

$$= \frac{\Pr\{ F_s \mid F_{xi}, x_j = 1 \} P\{ F_{xi}, x_j = 1 \}}{\Pr\{ F_s , x_j = 1 \}}$$

$$= \frac{\Pr\{ F_s \mid F_{xi}, x_j = 1 \} \Pr(F_{xi})}{\Pr\{ F_s \mid x_j = 1 \}}$$

$$= \frac{MI_{s,t}(X_i \mid x_j = 1) (1 - p_i)}{\Psi(x_j = 1)}$$

$$= \frac{\Psi(X_i = 0, x_j = 1) (1 - p_i)}{\Psi(x_j = 1)} \qquad (5)$$

Thus, by repeatedly using formula (5) we can always find a priority list for the purpose of system failure diagnosis. That is:

$$\max_i \{ DI_{s,t}(X_i, \text{ given } X_j \text{ is ok}) \} \qquad (i \neq j)$$

$$= \max_i \{ \frac{\Psi(x_i = 0, x_j = 1) (1 - p_i)}{\Psi(x_j = 1)} \} \qquad (i \neq j).$$

2.4 Comments and More Examples on Failure Diagnosis Policy

Since the profit related optimal diagnosis policy for a certain system depends on the actual conditions of the system, it is not easy to determine an optimal diagnosis strategy which yields a minimal expected total cost. The following two examples are employed to show that no simple optimal diagnosis strategy exists for general network systems.

Example 1

Consider a two components (A_1 and A_2) series system given below (Figure 11):

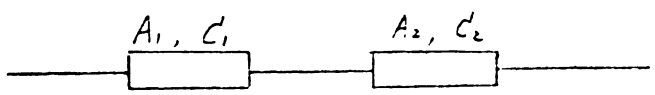


Figure 11

Here, p_i, c_i denote the reliability and diagnosing cost of component A_i respectively. Let F denote the event that

system failed and A_i denote the event that component A_i failed. Then, we have

$$P(A_1 \mid F) = \frac{q_1}{q_1 + p_1 q_2}$$

$$P(A_2 \mid F) = \frac{q_2}{q_1 + p_1 q_2}$$

Suppose that the system has failed and let the possible diagnosis policies be:

Policy 1: check component A_1 first

Policy 2: check component A_2 first

Let CP_1 and CP_2 denote the expected total costs involved in policy 1 and policy 2 respectively. Then we have:

$$CP_1 = c_1 + \frac{c_2 q_2}{q_1 + p_1 q_2}$$

$$CP_2 = c_2 + \frac{c_1 q_1}{q_1 + p_1 q_2}$$

and, $$CP_1 - CP_2 = c_1 - c_2 + \frac{c_2 q_2 - c_1 q_1}{q_1 + p_1 q_2}$$

$$= (c_1 p_1 q_2 - c_2 p_2 q_1) / [q_1 + p_1 q_2].$$

Obviously, if $\frac{c_1}{c_2} < \frac{p_2 q_1}{p_1 q_2}$, then $CP_1 < CP_2$ and policy 1 is better. Otherwise, policy 2 is better.

Example 2

For the three component system given below (Figure 12),

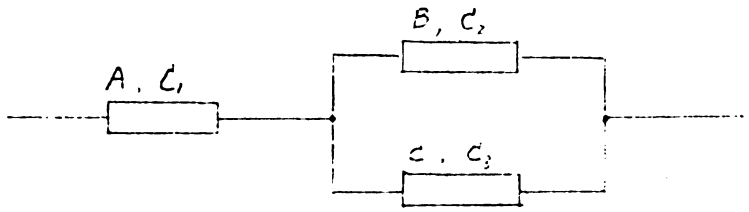


Figure 12

$$P(A \mid F) = \frac{q_1}{q_1 + p_1 q_2 q_3};$$

$$P(B \mid F) = \frac{q_2(q_1 + p_1 q_3)}{q_1 + p_1 q_2 q_3};$$

$$P(C \mid F) = \frac{q_3(q_1 + p_1 q_2)}{q_1 + p_1 q_2 q_3};$$

$$P(BC \mid F) = \frac{q_2 q_3}{q_1 + p_1 q_2 q_3}.$$

Suppose that the system has failed and let the possible diagnosis policies be:

Policy 1: Check component A first. Then,

$$CP_1 = c_1 + \frac{(c_2 + c_3)q_2q_3}{q_1 + p_1q_2q_3}.$$

Policy 2: Check component B first. Then,

$$CP_2 = c_2 + \frac{c_1q_1}{q_1 + p_1q_2q_3}.$$

Policy 3: Check component C first. Then,

$$CP_3 = c_3 + \frac{c_1q_1}{q_1 + p_1q_2q_3}.$$

Let $C_m = \min(c_2, c_3)$, and

$$CP_m = \min(CP_2, CP_3) = C_m + \frac{c_1q_1}{q_1 + p_1q_2q_3}.$$

Then, the numerator of $CP_1 - CP_m$ is:

$$c_1p_1q_2q_3 + (c_2 + c_3)q_2q_3 - C_m(q_1 + p_1q_2q_3).$$

If $C_m = c_2$, the numerator becomes:

$$c_1p_1q_2q_3 - c_2q_1(1 - q_2q_3) + c_3q_2q_3.$$

Set $c_3 = kc_2$ $(k \geq 1)$ Then, this quantity is:

$$c_1p_1q_2q_3 - c_2[q_1(1 - q_2q_3) - kq_2q_3] = f(c_i, p_i, k).$$

If $f < 0$ then policy 1 is optimal.

Otherwise, policy 2 is optimal.

These examples show that we cannot find a general easy algorithm for the failure diagnosis of general networks.

REFERENCES

[1] Barlow R. E., Fussell J.B., Singpuwalla N.D. (ed) "Reliability and fault tree analysis", SIAM, 1975

[2] Shimon Even, Graph Algorithms, Computer science press, Inc. Rockville, Maryland 20850,1985

[3] K.K. Aggorwal, Suresh Rai, "Reliability evaluation in computer communication network", IEEE Trans. Reliability, R-30, 1981

[4] Nakagawa. H. Baysian, "Decomposition method for computing the reliability of an orented network", IEEE Trans. Reliability, R-25, No.2, 1976

[5] John A. Buzacott, Clement C. Feng, "An algorithm for symbolic reliability computation with path-sets or cut-sets", IEEE Trans. Reliability, vol R-36, No.1, 1987

[6] R. G. Bennetts, "On the analysis of fault trees", IEEE Trans. Reliability, vol R-24, No.3, 1975

[7] Dean B. Wheeler, "Fault tree analysis using bit manipulation",

IEEE Trans. Reliability, vol R-26, No.2, 1977

[8] Barlow R. E., Proschan F., Statistical Theory of Reliability and Life Testing, N.Y. Holt, Rinehart and Winston, 1975

[9] Shi Dinghua, "A unified algorithm for computer-aided fault-tree analysis", Scientia Sinica (Series A) vol 27, No.1, 1984

[10] Feng Hsu, Shi Dinghua, "Problems and development of fault tree analysis", Report on 4th National China Symposium of Mathematical Theory of Reliability, Shanghai, Nov. 1-6,1983, Acta Automatica Sinica, No.4, 1984

[11] P. Camarda, F. Corsi, "An efficient simple algorithm for fault tree automatic synthesis from the reliability graph", IEEE Trans. Reliability, vol R-27, No.3, 1978

[12] S.A. Lapp, G.J. Powers, "Computer aided synthesis of fault-trees", IEEE Trans. Reliability, vol R-26, 1977

[13] Feng Hsu, Shi Dinghua,"Computer-aided fault tree construction" A selected Paper on 2nd China national conference on reliability, China science press, vol.2, May, 1985

[14] H. Frank, I.T. Frisch, "Analysis and design of survivable networks", IEEE Trans. Communication Technol., vol COM-18, pp501-519, October, 1970

[15] C. Singh, S. Asgarpoor, "Reliability evaluation of flow networks using delta-star transformations", IEEE Trans. Reliability, vol R-35, No.4, 1986

[16] Luig, Fratta, Ugo G. Montanari, "A recursive method based on case analysis for computing network terminal reliability", IEEE Trans. Communication, vol COM-26, NO.8, 1978

[17] Liao Jiongsheng, "A new approach for fault tree analysis", Scientia Sinica (Series A), vol. 25, No.9, 1982

[18] Cargal, J.M., "An alternative fault-tree algebra", IEEE Trans. Reliability, R-29, No.33, 1980

[19] Rosenthal, A., "Reliability and fault tree analysis", SIAM, 1975, pp135-152

[20] Luigi Fratta, "A boolean algebra method for computing the terminal reliability in a communication network", IEEE Trans. Circuit Theory, vol CT-20, No.3, May, 1973

[21] Frank A. Tillman, "Optimization techniques for system reliability with redundancy - a review", IEEE Trans. Reliability, vol R-26, No.3, 1977

[22] Yuji Nakagawa, Kyoichi Nakashima, "A heuristic method for determining optimal reliability allocation", IEEE Trans. Reliability, vol R-26, No.3, 1977

[23] Robert L. Bulfin, "Optimal allocation of redundant components for large systems", IEEE Trans. Reliability, vol R-34, No.3, 1985

[24] Yoshio Hattori, "Reliability optimization with multiple properties and integer variables", IEEE Trans. Reliability, vol R-28, No.1, 1979

[25] Martin Fox, Dorian Feldman, NOTES ON PROBABILITY, Department of Statistics and Probability, Michigan State University, August, 1986

[26] Frederick S. Hillier, Gerald J. Lieberman, Introduction to Operations Research, Stanford University, January,1980