SYSTEMS RELIABILITY EVALUATION OF THE COMPLEX AND LARGE SYSTEMS

by

MYOUNG HO LEE

B.S., Seoul National University, Seoul, Korea, 1973
M.B.A., Seoul National University, Seoul, Korea, 1976

A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1980

Approved by:

Co-Major Professor

Co-Major Professor

THIS BOOK CONTAINS NUMEROUS PAGES WITH THE ORIGINAL PRINTING BEING SKEWED DIFFERENTLY FROM THE TOP OF THE PAGE TO THE BOTTOM.

THIS IS AS RECEIVED FROM THE CUSTOMER.

ACKNOWLEDGEMENTS

The author wishes to express his sincere appreciation to his major professors, Dr. C. L. Hwang and Dr. F. A. Tillman, for their valuable guidance and creative advice in the course of this work. He also acknowledges the encouragement and helpful comments from Dr. D. S. Chung.

The author wishes to extend his deep gratitude to his parents for their constant help and inspiration ; to his wife, Heangsup, for her continued support and patience throughout this work.

The author thanks Mrs. M. K. Oppy for typing this thesis.

This study was partly supported by Office of Naval Research (Contract No. N00014-76-C-0842).

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

Reliability engineering is fast becoming an essential part of the expertise of engineers, particularly those engaged in the design of complex systems. It is, in fact, the rapidly increasing complexity of systems in all branches of engineering that is responsible for the awakening of interest in reliability.

Considering the history of reliability, reliability was first recognized as a pressing need during World War II. The preliminary steps taken were to establish joint Army and Navy (JAN) parts standards and to set up the Vacuum Tube Development Committee (VTDC) in June, 1943. At the close of the war, between 1945 and 1950, several studies revealed some startling results[62]:

1) A Navy study made during maneuvers showed that the electornic equipment was operative only 30 percent of the time.
2) An Army study revealed that between two-thirds and three-fourths of their equipment was out of commission or under repairs.
3) An Air Force study conducted over a 5-year period disclosed that repair and maintenance costs were about 10 times the original cost.
4) A study uncovered the fact that for every tube in use there were one on the shelf and seven in transit.
5) Approximately one electronics technician was required for every 250 tubes.
6) In 1937 a destroyer had 60 tubes; in 1952 the number had risen to 3,200.

These findings served as an impetus for further investigations. In this modern day of science, in which complex devices are utilized for military and scientific purposes, a high degree of reliability is an absolute necessity. Reliability studies have been further boosted by the requirements of modern complex systems such as of space research programmes and Air Craft Systems. In such complex systems, the failure of a part or component results not only in the loss of the failed item but most often also results in the loss of some larger assembly or system of which it is a part. Reliability evaluation for such complex systems requires the use of highly analytic techniques in engineering.

To estimate reliability and to perform reliability calculations, we must first define reliability.

The RETMA[1] definition of reliability states: "Reliability is the probability of a device performing its purpose adequately for the period of time intended under the operating conditions encountered." This definition is now accepted by most contemporary reliability authorities and hence is considered to be standard.

It should be observed that the definition stresses four elements, namely, probability, adequate performance, time, and operating conditions. These four factors are very significant since each of them plays an important role.

Using this basic definition, we can only calculate the reliability of simple components constituting complex systems.

To determine the reliability of a complex system, the system may be represented by a reliability logic diagram. Such a diagram shows that which components in the system must operate failure free for the successful operation of the system.

Reliability evaluation techniques will depend upon the nature of this diagram, as will be seen later.

The system reliability can be defined as the probability of successful communication between points s (source) and t (terminal).

And system reliability is a measure of how well a system performs or meets its design objective, and it is usually expressed in terms of the

[1] RETMA, Radio Electronics and Television Manufacturers Association, is now known as EIA, Electronics Industries Association.

reliabilities of the subsystems or components. The following terminologies are defined. A "part" or "element" is the least subdivision of a system, or an item that cannot ordinarily be disassembled without being destroyed. A "circuit" is a collection of parts that has a specific function. A "component" then is a collection of parts and/or circuits, which represents a self-contained element of a complete operating system and performs a function necessary to the operation of that system. "Unit", "component", and "sub-system" are synonymous. A "system" can then be characterized as a group of sub-system especially integrated to perform a specific operational function or functio

Conceptually, the task of determining the system reliability is a simple one. The method is described by most reliability texts [8],[62]. The general procedure of determining the system reliability can be described as follows:

Firstly	Consider each of the possible states of the system
Secondly	Identify which of the states result in successful system operation
Thirdly	Determine the probability of occurrence of each successful state
Finally	Add all these probabilities together

The final sum is the system reliability.

All this is possible for small systems but as the system becomes large or complex the problem one usually faces is computational difficulty and time consuming. Because of the large amount of work and computer time required to determine the reliability of a complex system by exact methods, approximate procedures that consider only a part of the information about system states are frequently used. Here a complex system may be defined as a system which cannot be reduced to a series-parallel system. Keeping this in view, in this study, a thorough discussion of reliability evaluation problems is presented. The contents include a critical review of evaluation techniques for system reliability with small/complex, moderate/complex and large/complex systems.

The objectives of this thesis are:

1) to present a critical review and classification of small to large complex system reliability problems which have been analyzed with various evaluation techinques;

2) to illustrate the theoretical concepts and the practical formulae required to solve reliability problems in the analysis and design of system networks;

3) to do a careful examination of computational procedures of each technique and provide an insight into its strengths and weakness;

4) to propose the use of multiple attribute decision making (MADM) methods for determining a suitable system reliability evaluation technique depending upon the size of the reliability system configuration.

A state-of-the-art review of the literature related to system reliability evaluation techniques for complex and large systems is presented in Chapter 2.

The literature is classified as follows:

System reliability models:

Small Complex Systems

Moderate Complex Systems

Large Systems

Chapter 3 describes the detailed reliability evaluation techniques for the small complex and moderate complex system.

Those methods which can be used for a small/complex system are:

Exhaustive search of successful states
Direct canonical expansion
Probability map method

As the size of the system configuration becomes moderate, such evaluation techniques as mentioned below may be employed. A symbolic reliability expression or simplified reliability expression may be obtained by using the concept of logical signal relations or the concept of exclusive operator.

The methods for the moderate complex system can be elaborated as follows:

Probability calculus
Bayes' Theorem
Flow graph method
Parametric method
Algebraic extraction
Fast algorithm
Algorithm for SYMRAP (Symbolic Reliability Analysis Program)
An efficient method for reliability evaluation of a general network
Symbolic reliability evaluation using logical signal relations

In Chapter 4 the evaluation methods for the large systems reliability are introduced. Those techniques are as follows:

A computer program for approximating system reliability--success paths and cut sets approach

An algorithm to determine the reliability of a complex system--Minimal cuts and coherent systems approach

A Boolean algebra method for computing the terminal reliability in a communication network.

A Monte Carlo method for system reliability calculations

For a large complex system, computer programs provide the set of minimal cuts and calculates the minimal-cut approximation to system reliability. Based on minimal path (tie) sets, reliability approximations for a large/complex system can be obtained. And Monte Carlo method for system reliability evaluation has been found to be efficient when component reliabilities are sampled by Monte Carlo method.

In general, the basic problem is to decide what kind of evaluation technique should be employed depending upon the size and configuration of the system.

All the evaluation techniques employed in the papers surveyed have limited success in solving some large/complex system reliability evaluation problems. Few techniques have been completely effective when applied to large system reliability problems.

Chapter 5 demonstrates the decision making process through the applications of the multiple attribute decision making (MADM) methods in the selection of a suitable system reliability evaluation technique for the corresponding system configuration.

The MADM methods utilized here are:

Conjunctive Constraints
Simple Additive Weighting
Linear Assignment method
ELECTRE
TOPSIS (Technique for Order Preference by Similarity to Ideal Solution)

To cover all the techniques in a discussion of this sort is practically an impossibility, since the system reliability evaluation techniques are still in the process of evolution and we are continually learning more.

We have tried our best to compile and explain all the current and significant works in this area in a systematic and effective manner.

CHAPTER 2. SYSTEMS RELIABILITY EVALUATION TECHNIQUES FOR COMPLEX/LARGE SYSTEMS - A REVIEW

1. INTRODUCTION

In many practical situations, a reliability model has a nonseries-parallel configuration. Here a complex system may be defined as a system which cannot be reduced to a series-parallel system.

A network of nonseries-parallel systems with ten components has $2^{10} = 1024$ states. A computer could, of course, evaluate the reliability of this system in a very short time [36]. But consider a system with twenty components. The computation becomes considerably more difficult and may even be infeasible with modern day computers [41]. This situation has promoted considerable research into methods of approximate terminal-pair reliability analysis of large systems [13, 36, 41, 52]. In this discussion a large system is defined as a system which has more than ten components. To analyze the reliability of large networks, algorithms which are efficient and which can easily be implemented on a computer are needed.

A state-of-the-art review of the literature related to system reliability evaluation techniques for complex and large systems is presented in this paper. In Table 2.1, the literature for the different system configurations is separated into the following model sub-categories; small/complex, moderate/complex and large/complex systems. In Table 2.2, the same literature is reclassified to indicate the variety of evaluation techniques utilized.

We have tried to be reasonably complete; those papers not included were either inadvertently overlooked or considered not to bear directly on the topics of this survey. We apologize to both the readers and the researchers if we have omitted any relevant papers.

Table 2.1.The reference classification for the system reliability evaluation techniques with regard to various system models.

Systems Models	References
Small Complex Systems	5, 8, 14, 18, 34, 45, 54, 62
Moderate Complex Systems	2, 3, 5, 6, 7, 8, 11, 12, 13, 14, 16, 18, 26, 27, 28, 32, 33, 34, 36, 37, 38, 39, 41, 42, 45, 47, 49, 52, 53, 55, 62, 65, 67
Large Systems	8, 13, 18, 25, 27, 28, 36, 39, 46, 47, 52, 62, 67, 71, 74, 76, 77, 78, 79, 81

9

Table 2.2. The reference classification for the evaluation techniques employed for systems reliability.

	Evaluation Techniques	References
1.	Exhaustive search of successful states	5, 8, 18, 54, 62
2.	Direct canonical expansion	5, 45
3.	Probability map method	5, 14, 34, 45
4.	Probability calculus	5, 26, 42, 53
5.	Bayes Theorem	5, 14, 84
6.	Flow graph method	8, 47, 49
7.	Parametric method	11, 12, 36, 52, 62
8.	Algebraic extraction	2, 5, 45
9.	Fast algorithm	2, 5, 6, 34, 67
10.	Algorithm for SYMRAP (Symbolic Reliability Analysis Program)	13, 18, 26, 27, 28, 32, 33, 36, 39, 41, 49, 62
11.	An efficient method for reliability evaluation of a general network	2, 6, 14, 16, 28, 37, 38, 41, 55, 65
12.	Symbolic reliability evaluation using logical signal relations	3, 6, 7, 14, 18, 42, 45, 47, 55, 65, 67
13.	A computer program for approximating system reliability	13, 36, 52, 62
14.	An algorithm to determine the reliability of a complex system	8, 9, 25, 36, 46, 62
15.	A Boolean algebra method for computing the terminal reliability in a communication network.	1, 16, 18, 27, 28, 39, 47, 52, 58, 67
16.	A Monte Carlo method for system reliability calculations	71, 74, 76, 77, 78, 79, 81
17.	Miscellaneous	19, 20, 22, 23, 29, 31, 40, 43, 66, 72, 75, 80, 82

2. SYSTEM MODELS

In this review, we assume that the reader is familiar with the material treated in these models. For a discussion of the definitions and formulations of the basic concepts, see [8, 14, 26, 27, 33, 44, 45, 59, 62, 63]. We now briefly review each of the models considered in this survey.

2.1. Small Complex Systems

a) Three components (each component has two states, eg. operating and failure) of a system with reliabilities R_{13}, R_{12}, R_{32}, as shown in Fig. 2.1, are connected to form the delta configuration. This can be transformed into star equivalent with reliabilities R_{10}, R_{20}, R_{30}.

b) A nonseries-parallel diagram as shown in Fig. 2.2. Component A feeds B only, and, in parallel, C feeds D only. However, E can feed either B or D (can alternate).

c) A 'bridge network' as shown in Fig. 2.3. E represents a two-way bridging element.

2.2. Moderate Complex Systems

a) A communication system with five modes and seven branches (two of these being interconnecting) as shown in Fig. 2.4.

b) The network as shown in Fig. 2.5., where branches with reliabilities P_3 and P_6 are the interconnecting branches.

c) The ARPA network, shown in Fig. 2.6., where there are 13 different minimal paths.

2.3. Large System

2.3.1. Large Series-Parallel System

a) The system shown in Fig. 2.7 is a relatively complex series-parallel network.

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

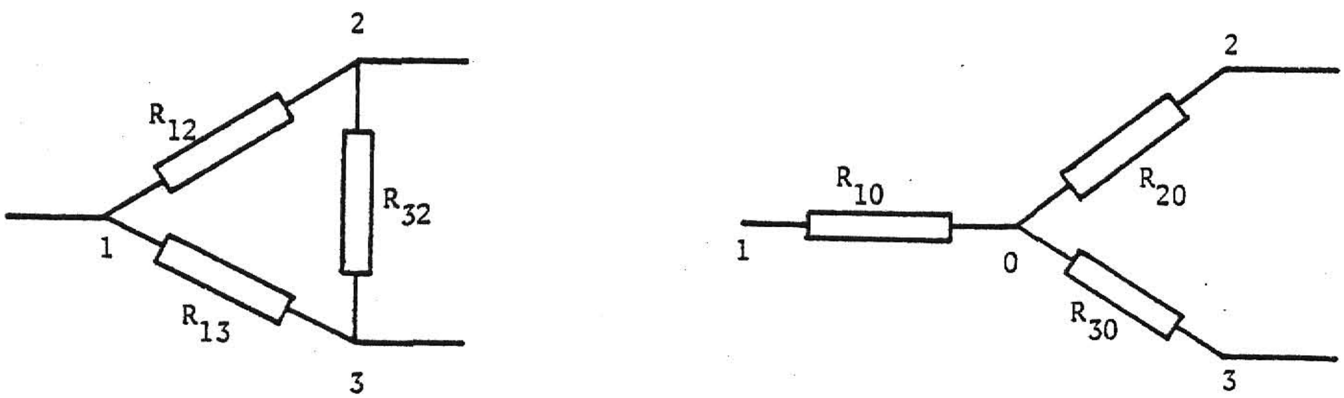


Fig. 2.1. Delta-star configuration

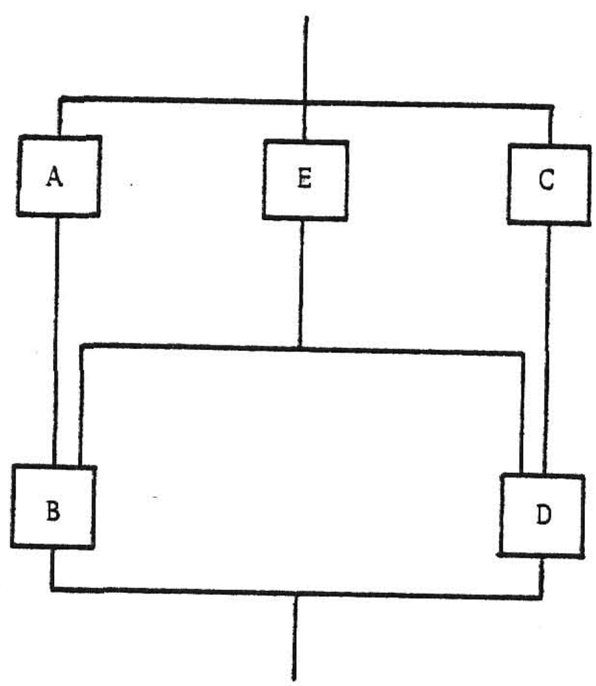


Fig. 2.2. An alternate connection

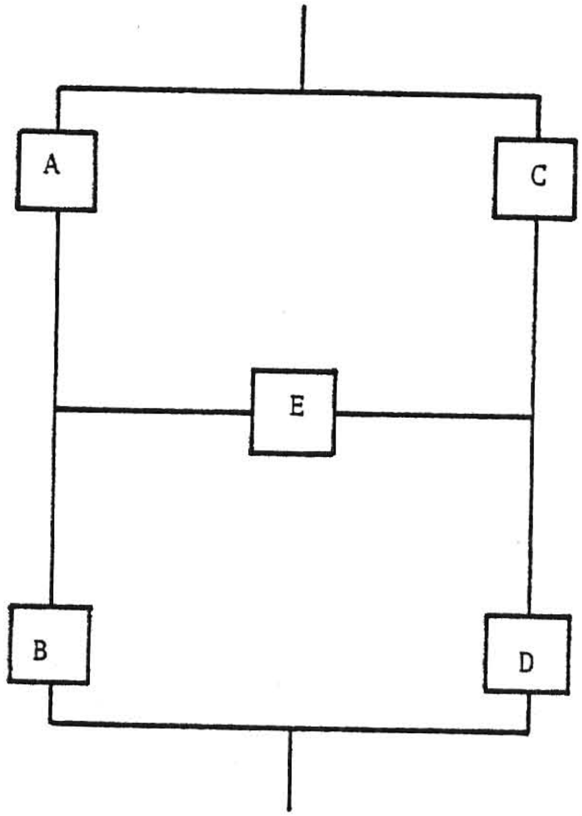


Fig. 2.3, A bridge connection

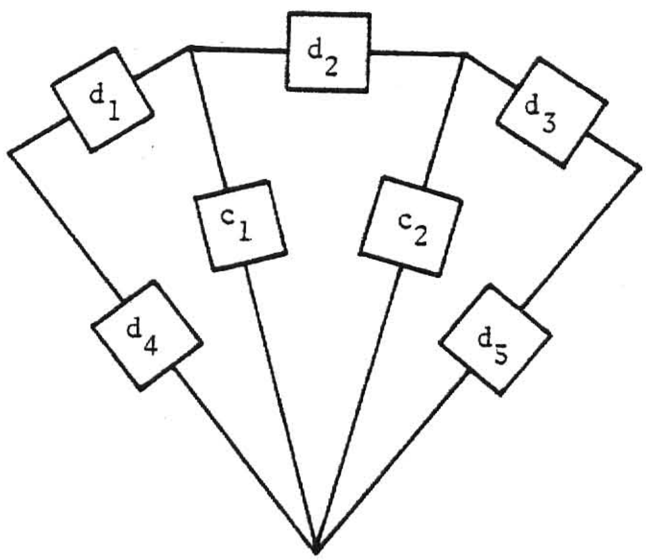


Fig. 2.4. General network [4]

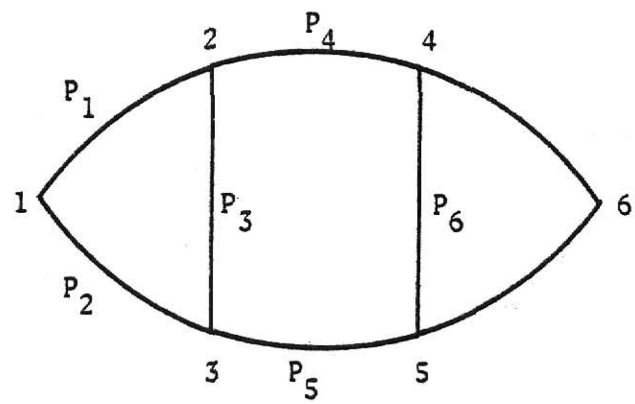


Fig. 2.5. A general network [48]

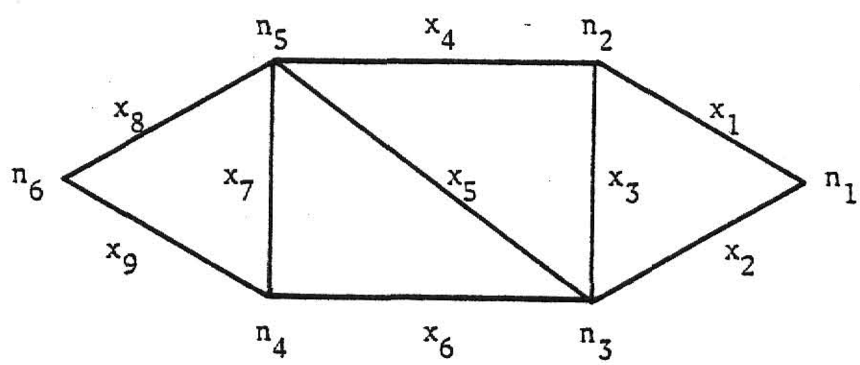


Fig. 2.6. Modified graph of ARPA network [55]

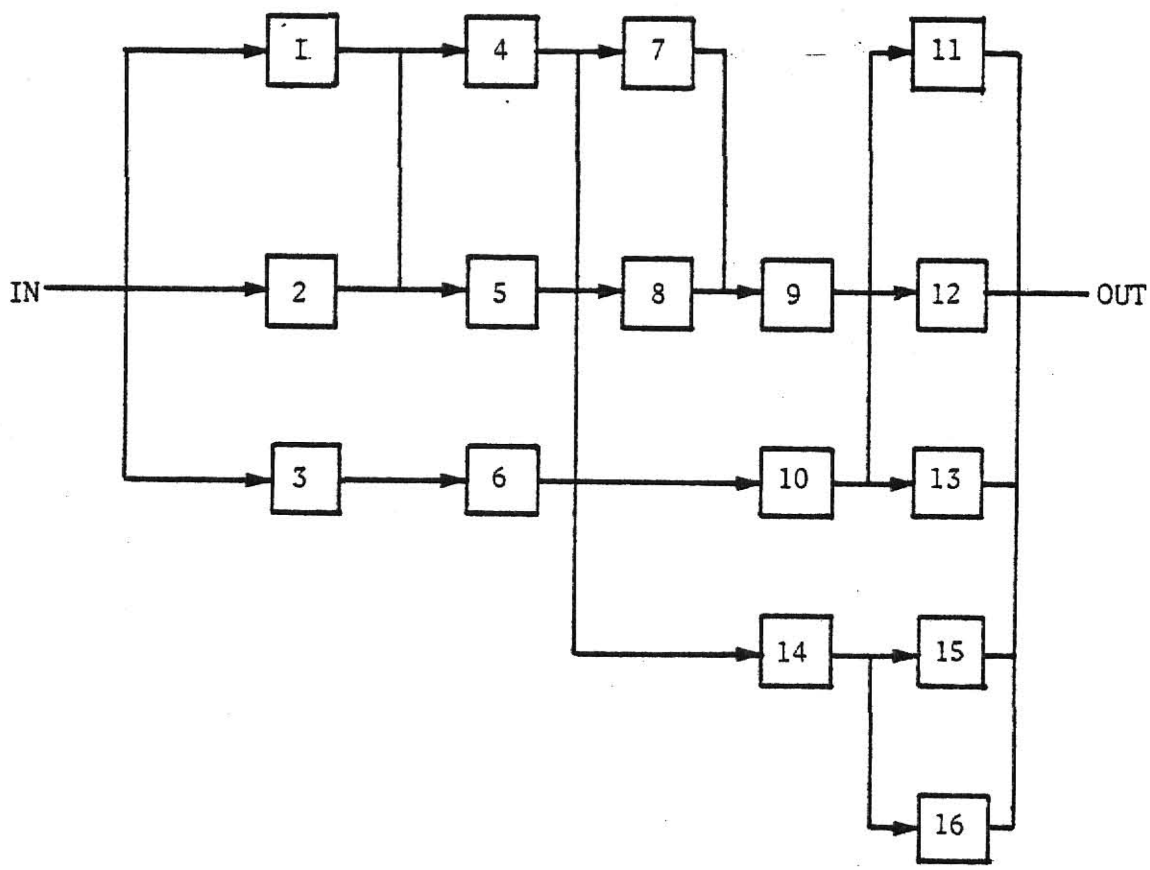


Fig.2.7. System diagram for bounds program [52]

2.3.2 Large Complex System

a) A reliability network shown in Fig. 2.8 is a ten-component system.

b) The ARPA (Advanced Research Project Agency) computer network shown in Fig. 2.9 is to be evaluated for the terminal reliability between UCLA and CMU.

c) A more connected network is shown in Fig. 2.10.

d) A simple long distance telephone network shown in Fig. 2.11 has 18 (S, T) paths.

e) A complicated boiler safety system is shown in Fig. 2.12.

f) A hypothetical 18-components S-coherent complex system is shown in Fig. 2.13.

3. STATEMENT OF THE VARIOUS SYSTEM RELIABILITY EVALUATION PROBLEMS

The structure of the system reliability problems that are relevant to our study are stated below and the literature is identified in Table 2.3

Problem 1

To evaluate the overall reliability of a system which cannot be reduced to a series-parallel model, such as a Delta-Star configuration, an alternate or bridge type connection, when the reliabilities of the elements are known (Figs. 2.1-2.3).

Problem 2

To find the simplified reliability expression, the symbolic reliability expression or the terminal-pair reliability expression of a general network (Figs. 2.6, 2.11, 2.12).

Problem 3

To compute the reliability of the system when its configuration is a moderate size (Figs. 2.4-2.6).

Problem 4

To determine the reliability of a large/complex system which requires a computer program for approximating system reliability from the given reliabilities of its elements (Figs. 2.7, 2.8, 2.13).

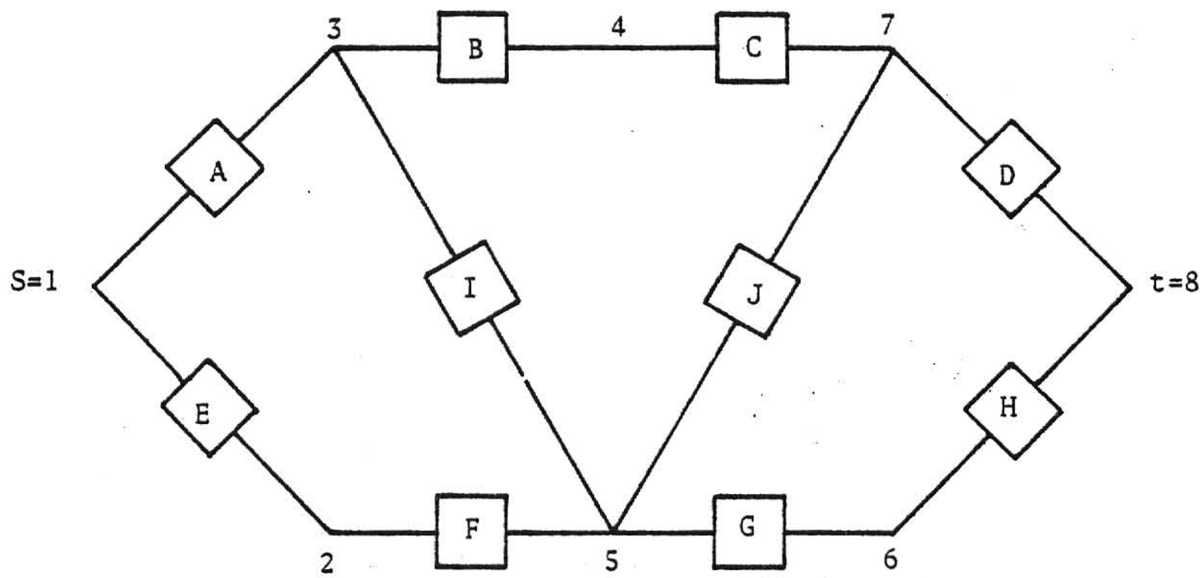


Fig. 2.8. Reliability network [36]

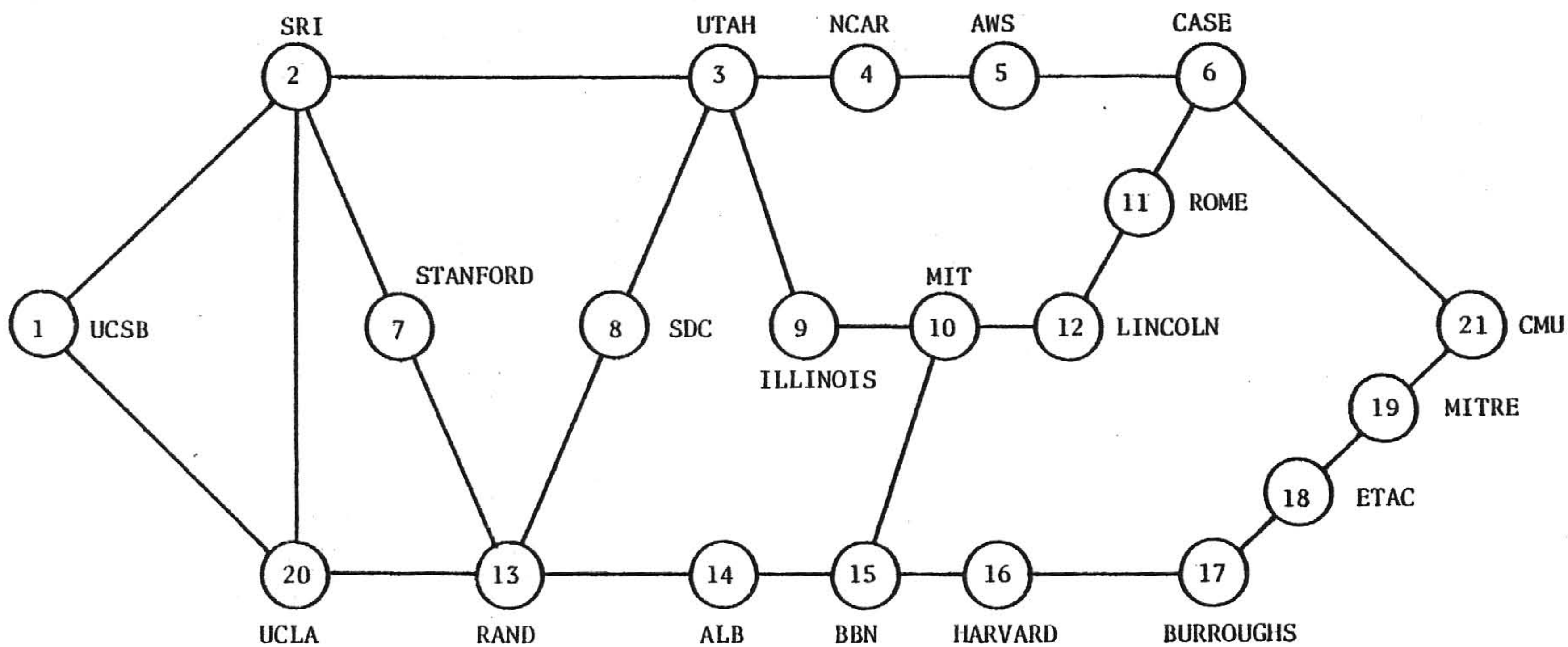


Fig. 2.9. Topology of the ARPA computer network [28]

ILLEGIBLE

THE FOLLOWING DOCUMENT (S) IS ILLEGIBLE DUE TO THE PRINTING ON THE ORIGINAL BEING CUT OFF

ILLEGIBLE

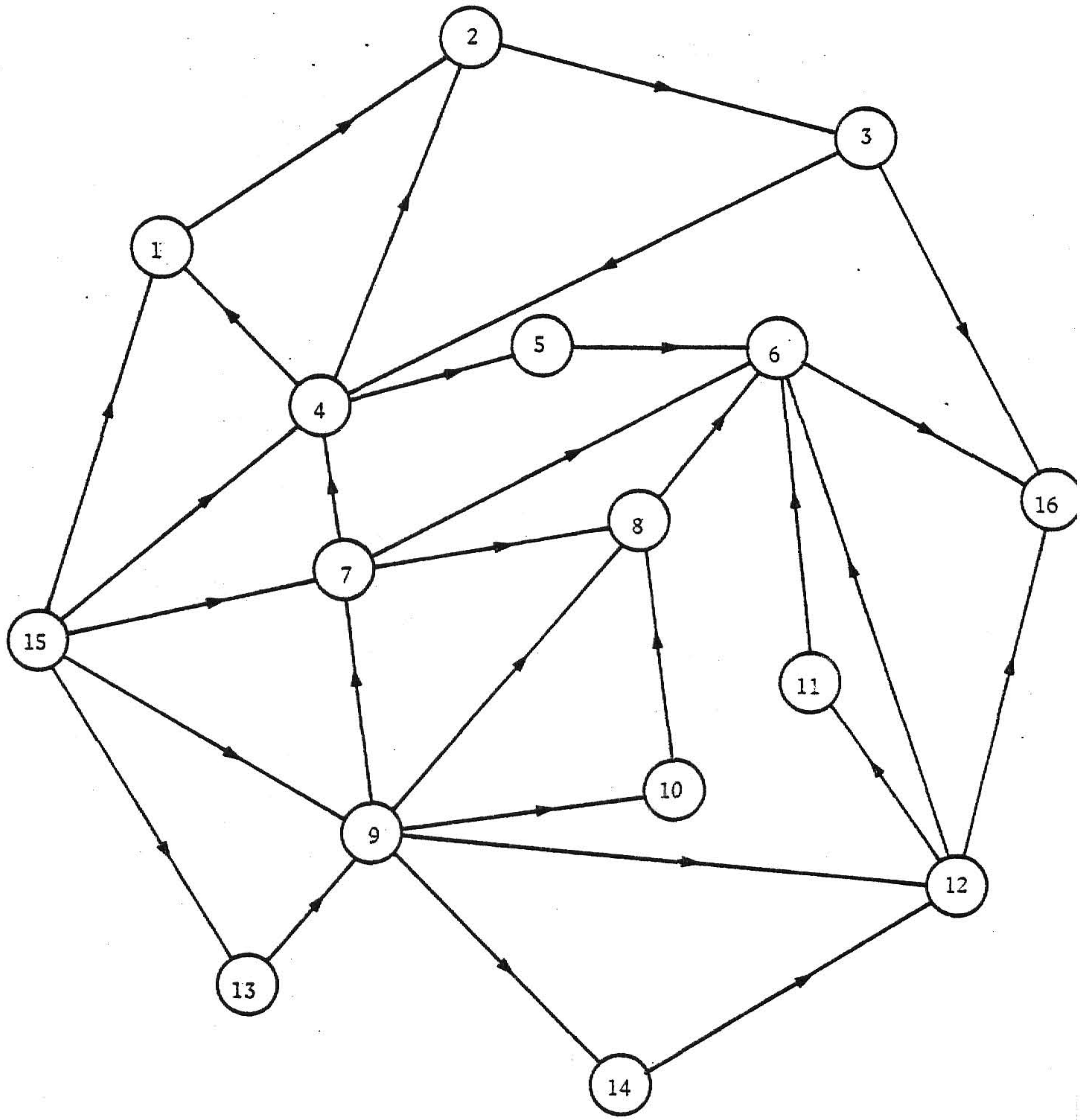


Fig. 2.10. Topology of a more connected network [28]

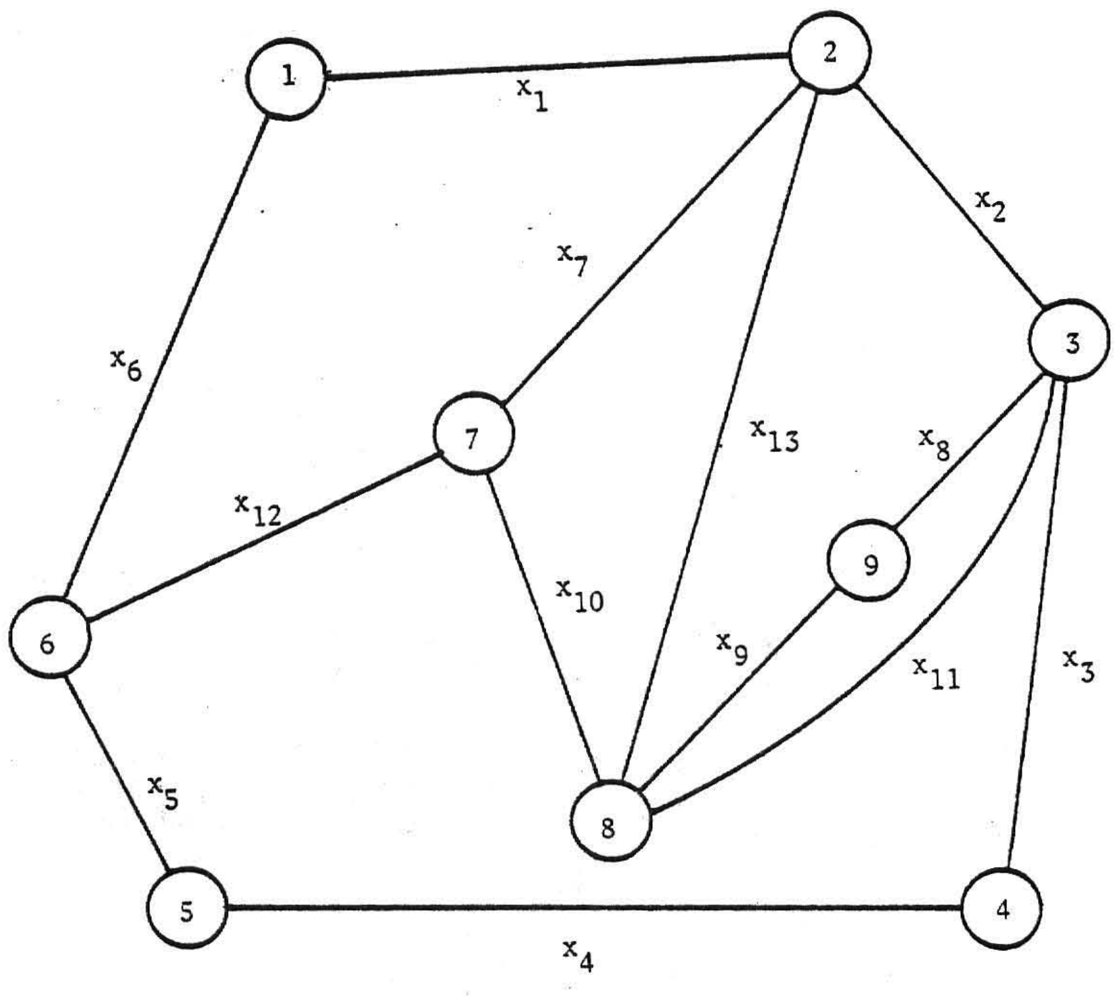


Fig. 2.11. A simple long distance telephone network [41]

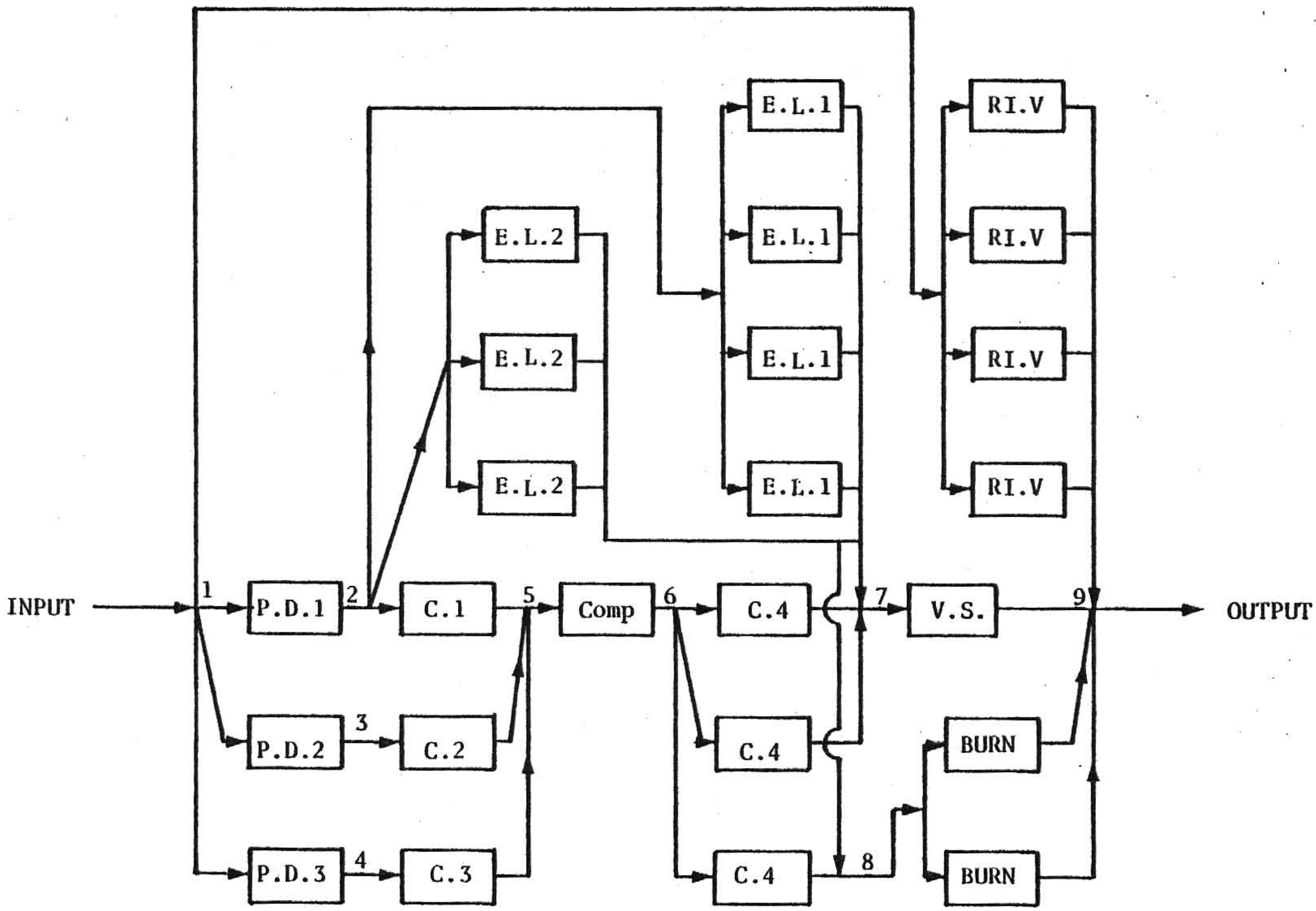


Fig. 2.12. A boiler safety system [33]

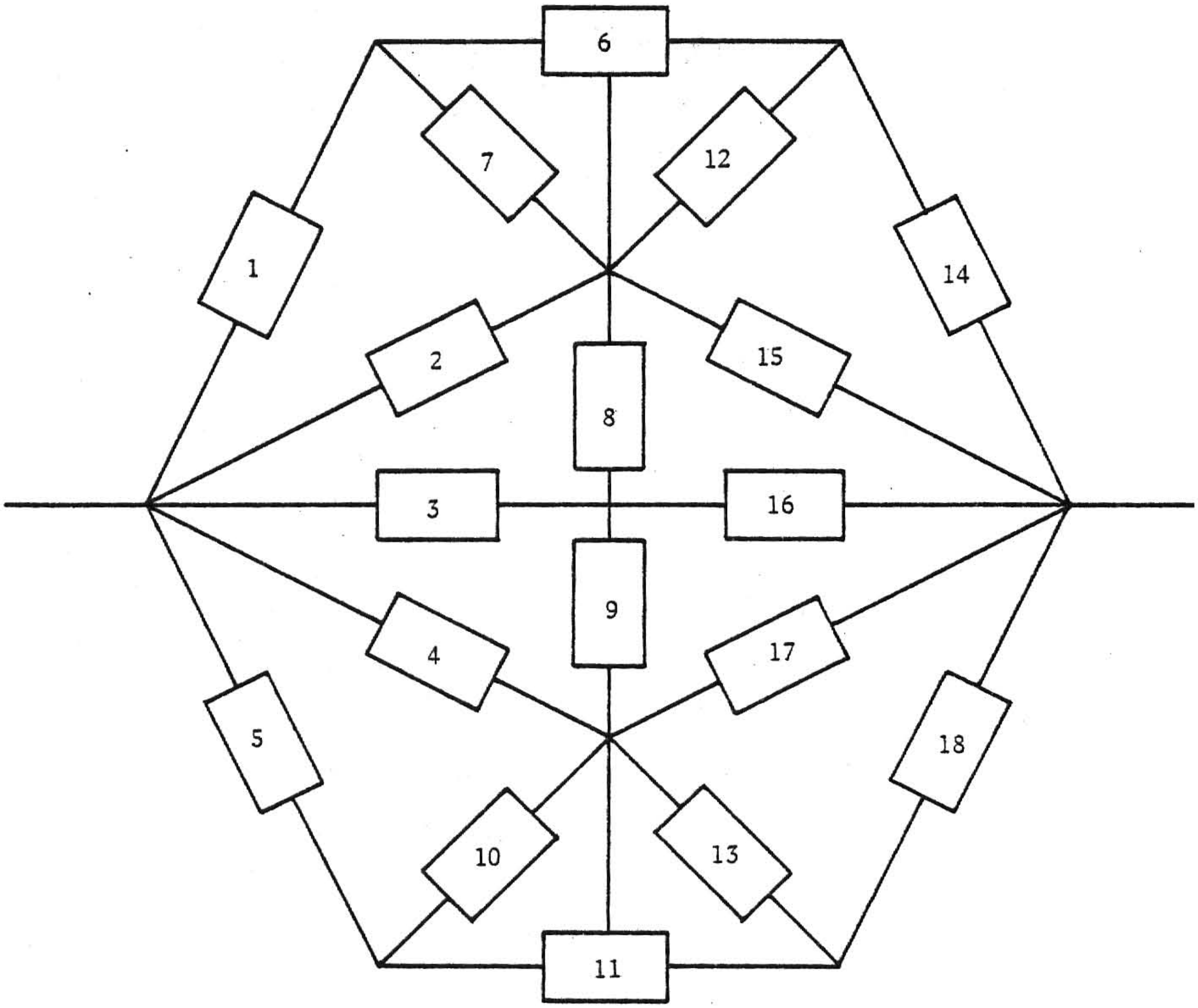


Fig. 2.13. A hypothetical 18-component s-coherent complex system [78].

22

Table 2.3. Reference classifications with regard to structure of the various system reliability evaluation problems.

Formulation of Problems	References
Problem 1: System reliability evaluation of Delta-Star, alternate or bridge type connections (Figs. 2.1 - 2.3).	5, 8, 14, 18, 34, 45, 54, 62.
Problem 2: System reliability evaluation with the simplfied reliability expression, the symbolic reliability expression or the terminal-pair reliability expression of a general network (Figs. 2.6, 2.11, 2.12).	2, 3, 6, 7, 13, 14, 16, 18, 21, 26, 27, 28, 32, 33, 36, 37, 38, 39, 41, 42, 45, 49, 55, 62, 65, 67
Problem 3: Computing the reliability of a system whose size of configuration is moderate (Figs. 2.4 - 2.6).	2, 3, 5, 6, 7, 8, 11, 12 13, 14, 16, 18, 26, 27, 28, 32, 33, 34, 36, 37, 38, 39, 41, 42, 45, 47, 49, 52, 53, 55, 62, 65, 67
Problem 4: Determining the reliability of a large/complex system which needs a computer program (Figs. 2.7, 2.8, 2.13).	8, 13, 25, 36, 46, 52, 62, 71, 74, 76, 77, 78, 79, 81
Problem 5: Obtaining the terminal reliability between a given pair of nodes in quite large communication networks (Figs. 2.9, 2.10).	18, 27, 28, 39, 47, 52 67

Problem 5

To compute the terminal reliability between a given pair of nodes, namely, the probability that there exists at least one path between these two nodes in quite large communication networks (Figs. 2.9, 2.10).

4. RELIABILITY EVALUATION TECHNIQUES USED TO DETERMINE THE VARIOUS SYSTEM MODELS

Many algorithms have been proposed but only a few have been effective when applied to large/complex system reliability evaluation problems. The literature on the reliability evaluation techniques which are relevant to this study is classified in Table 2.2. All the evaluation techniques employed have limited success in solving all of the problems.

Table 2.4 shows those system reliability evaluation techniques employed to find the overall reliability of corresponding system configurations from Fig. 2.1 through Fig. 2.13.

For a small/complex system, exhaustive search method, direct canonical expansion or probability map method can be used. As the size of the system configuration becomes moderate, such evaluation techniques as probability calculus, Bayes theorem, parametric method, algebraic extraction or fast algorithm may be employed. A symbolic reliability expression or simplified reliability expression may be obtained by using the concept of logical signal relations or the concept of exclusive operator.

For a large/complex system, computer programs provide the set of minimal cuts and calculates the minimal-cut approximation to system reliability. The literature on cut set generation techniques for a system which is relevant to this survey is [9, 15, 17, 24, 57, 83]. Based on minimal path (tie) sets, reliability approximations for a large/complex system can be obtained. And Monte Carlo method for system reliability evaluation has been found to be efficient when component reliabilities are sampled by Monte Carlo method.

In a large communication network, the terminal reliability between a given pair of nodes can be determined approximately with the aid of a computer.

24

Table 2.4. Reliability evaluation techniques used to determine the example system reliability of Fig.2.1 through Fig. 2.13.

Small Complex System

Figure	Evaluation Method Applicable	References	Recommended Method
Fig.2.1	1. Exhaustive Search of Successful States	5, 8, 18, 54, 62	
	2. Direct Canonical Expansion	5, 45	
	4. Probability Calculus	5, 26, 42, 53	
	5. Bayes Theorem	5, 14	
	7. Parametric Method	11, 12, 36, 52, 62	7
Fig.2.2	1. Exhaustive Search of Successful States	5, 8, 18, 54, 62	
	2. Direct Canonical Expansion	5, 45	
	3. Probability Map	5, 14, 34, 45	3
	4. Probability Calculus	5, 26, 42, 53	
	5. Bayes Theorem	5, 14, 84	5
	8. Algebraic Extraction	2, 5, 45	
Fig. 2.3	1. Exhaustive Search of Successful States	5, 8, 18, 54, 62	
	2. Direct Canonical Expansion	5, 45	
	3. Probability Map	5, 14, 34, 45	3
	4. Probability Calculus	5, 26, 42, 53	
	5. Bayes Theorem	5, 14	5
	6. Flow Graph Method	8, 47, 49	
	7. Parametric Method	11, 12, 36, 52, 62	7
	8. Algebraic Extraction	2, 5, 45	

Table 2.4(Continued)

25

Moderate Complex System

Figure	Evaluation Method Applicable	References	Recommended Method
Fig. 2.4	1. Exhaustive Search	5, 8, 18, 54, 62	
	2. Direct Canonical Expansion	5, 45	
	4. Probability Calculus	5, 26, 42, 53	
	5. Bayes Theorem	5, 14	5
	6. Flow Graph Method	8, 47, 49	6
	8. Algebraic Extraction	2, 5, 45	
	9. Fast Algorithm	2, 5, 6, 34, 67	9
	11. An Efficient Method for a General Network	2, 6, 14, 16, 28, 37, 38, 41, 55, 65	11
	12. Symbolic Reliability Evaluation Using Logical Signal Relations	3, 6, 7, 14, 18, 42, 45, 47, 53, 65, 67	12
Fig. 2.5	4. Probability Calculus	5, 26, 42, 53	
	5. Bayes Theorem	5, 14	5
	8. Algebraic Extraction	2, 5, 45	
	9. Fast Algorithm	2, 5, 6, 34, 67	
	11. An Efficient Method for a General Network	2, 6, 14, 16, 28, 37, 38, 41, 55, 65	
Fig. 2.6	10. Algorithm for SYMRAP	13, 18, 26, 27, 28, 32, 33, 36, 39, 41, 49, 62	
	11. An Efficient Method for a General Network	2, 6, 14, 16, 28, 37, 38, 41, 55, 65	11

Table 4 (continued)

Large System

Large Series-Parallel System

Figure	Evaluation Method Applicable	References	Recommended Method
Fig. 7	13. A computer Program for Approximating System Reliability	13, 36, 52, 62	13
	14. An algorithm to determine the reliability of a complex system	8, 9, 25, 36, 46, 62	
	15. A Boolean algebra method for computing the terminal reliability in a communication network	1, 16, 18, 27, 28, 39, 47, 52, 58, 67	
	16. A Monte Carlo method for system reliability calculations	71, 74, 76, 77, 78, 79, 81	

Large Complex System

Figure	Evaluation Method Applicable	References	Recommended Method
Fig. 8	13. A computer program for approximating system reliability	13, 36, 52, 62	
	14. An algorithm to determine the reliability of a complex system	8, 9, 25, 36, 46, 62	14
	15. A Boolean algebra method for computing the terminal reliability in a communication network	1, 16, 18, 27, 28, 39, 47, 52, 58, 67	
	16. A Monte Carlo method for system reliability calculations	71, 74, 76, 77, 78, 79, 81	
Fig. 9	13. A computer program for approximating system reliability	13, 36, 52, 62	
	14. An algorithm to determine the reliability of a complex system	8, 9, 25, 36, 46, 62	
	15. A Boolean Algebra Method for computing the terminal reliability in a communication network	1, 16, 18, 27, 28, 39, 47, 52, 58, 67	15
	16. A Monte Carlo method for system reliability calculations	71, 74, 76, 77, 78, 79, 81	

Table 4 (continued)

Large Complex System

Figure	Evaluation Method Applicable	References	Recommended Method
Fig. 10	13. A computer program for approximating system reliability	13, 36, 52, 62	
	14. An algorithm to determine the reliability of a complex system	8, 9, 25, 36, 46, 62	
	15. A Boolean algebra method for computing the terminal reliability in a communication network	1, 16, 18, 27, 28, 39, 47, 52, 58, 67	15
	16. A Monte Carlo method for system reliability calculations	71, 74, 76, 77, 78, 79, 81	
Fig. 11	10. Algorithm for SYMRAP	13, 18, 26, 27, 28, 32, 33, 36, 39, 41, 49, 62	10
	11. An efficient method for reliability evaluation of a general network	2, 6, 14, 16, 28, 37, 38, 41, 55, 65	
	13. A computer program for approximating system reliability	13, 36, 52, 62	
	15. A Boolean algebra method for computing the terminal reliability in a communication network	1, 16, 18, 27, 28, 39, 47, 52, 58, 67	
Fig. 12	10. Algorithm for SYMRAP	13, 18, 26, 27, 28, 32, 33, 36, 39, 41, 49, 62	10
	11. An efficient method for reliability evaluation of a general network	2, 6, 14, 16, 28, 37, 38, 41, 55, 65	
	13. A computer program for approximating system reliability	13, 36, 52, 62	
	15. A Boolean algebra method for computing the terminal reliability in a communication network	1, 16, 18, 27, 28, 39, 47, 52, 58, 67	

Table 4 (continued)

Large Complex System

Figure	Evaluation Method Applicable	References	Recommended Method
Fig. 13	13. A computer program for approximating system reliability	13, 36, 52, 62	
	14. An algorithm to determine the reliability of a complex system	8, 9, 25, 36, 46, 62	
	15. A Boolean algebra method for computing the terminal reliability in a communication network	1, 16, 18, 27, 28, 39, 47, 52, 58, 67	
	16. A Monte Carlo method for system reliability calculations	71, 74, 76, 77, 78, 79, 81	16

Miscellaneous methods for evaluating complex system reliability are a moment approach [35], a block diagram approach [56], a Bayesian decomposition method [50], a decomposition method by a Boolean expression [51] and such methods in [72, 75, 80, 82]. The evaluation techniques found in [1, 10, 60, 61, 64, 73] have been demonstrated to be effective when applied to network reliability problems.

5. CONLUDING REMARKS

All the evaluation techniques employed in the papers surveyed have limited success in solving some large/complex system reliability evaluation problems. Few techniques have been completely effective when applied to large system reliability problems.

We suggest the following new directions for additional system reliability evaluation work. First, a generally efficient graph partitioning technique for reliability evaluation of large, highly interconnected networks should be found. Second, extend the single objective problems to include multiple objective system reliability problems. One of the major reliability problems in the planning and design of a new system is the development of a system reliability goal. A reliability goal is the basis for comparing alternate system designs and deciding upon an optimal design policy.

A system designer must decide on what trade-offs he can make with the reliability engineer to achieve the reliability objective. These trade-offs may involve sacrificing weight, volume, cost or similar parameters in favor of achieving the prescribed reliability.

To cover all the techniques in a discussion of this sort is practically an impossibility, since the system reliability evaluation techniques are still in the process of evolution and we are continually learning more. We have tried our best to compile and explain all the current and significant works in this area in a systematic and effective manner.

This survey is a sequel to our previous literature surveys on optimization of system reliability [68], on availability of maintained systems [69], and on system effectiveness models [70].

CHAPTER 3 EVALUATION TECHNIQUES FOR THE SMALL COMPLEX AND MODERATE COMPLEX SYSTEMS RELIABILITY

Reliability evaluation techniques have their inherent characteristics and specific superiorities to solve complex systems reliability problems. In this chapter, various evaluation techniques are treated to:

1) evaluate the overall reliability of a system which cannot be reduced to a series-parallel model, such as a Delta-Star configuration, an alternate or bridge type connection, when the reliability of the elements are known,
2) find the simplified reliability expression, the symbolic reliability expression or the terminal-pair reliability expression of a general network;
3) compute the reliability of the system when its configuration is a moderate size.

In the previous chapter, references for system reliability evaluation techniques have been reviewed. The computational procedures of the evaluation techniques for the small complex and moderate complex system will be described in this chapter.

These evaluation techniques are:

1. Exhaustive search of successful states
2. Direct canonical expansion
3. Probability map method
4. Probability calculus
5. Bayes' Theorem
6. Flow graph method
7. Parametric method
8. Algebraic extraction

9. Fast algorithm
10. Algorithm for SYMRAP (Symbolic Reliability Analysis Program)
11. An efficient method for reliability evaluation of a general network
12. Symbolic reliability evaluation using logical signal relations

Among these evaluation techniques, Exhaustive search of successful states, Direct canonical expansion, and Probability map method are for the small complex system reliability. But, as the size of the system configuration becomes moderate, the problem one usually faces is computational difficulty and time consuming. So firstly those evaluation methods, having been successfully applied for the moderate complex system reliability problems, will be reviewed, classified, and modified. To cover a comprehensive discussion, various evaluation techniques will be used to solve various system reliability problems. Before dealing with each specific evaluation technique to system reliability problems, the following assumptions are made:

1) All elements are initially operating.
2) The states of all elements are statistically independent. This means that the failure of one element does not affect the probability of failure of other elements.
3) Each element may be represented as a two terminal device
4) The state of each element and of the system is either good (operating) or bad (failed).
5) The nodes of the systems are perfect.

3.1 Probability Calculus Method in Complex System Reliability

1. Introduciton

In the methods such as Exhaustive Search of Successful States and Direct Canonical Expansion, the resulting reliability expression is quite lengthy and requires much computational work for numerical evaluation of reliability. But this method can avoid the above difficulties.

2. Statement of the Problem and the Computational Procedure

The probability calculus method is based on a simple theorem [53]:

$$P(X \cup Y) = P(X) + P(Y) - P(X \cap Y) \tag{1}$$

In slight variation of this method, with the minimal cuts, an equation is derived which gives the system unreliability as a function of the unreliabilities of the components; the reliability is obtained by subtracting from one. A generalization of the above method is by Poincare [42]. Suppose that the system has m minimal paths. The system reliability is the probability that the state which the system is in, characterized as a binary vector, contains at least one of these minimal paths.

Let P_i, $i = 1,\ldots,m$, be the probability of the i^{th} minimal path; for example, if the components are statistically independent, P_i would be the product of the component reliabilities for those components that characterize the ith minimal path. Let R be the system reliability. Then, reliability expression can be written as:

$$R = S_1 - S_2 + S_3 \ldots + (-1)^{m-1} S_m \tag{2}$$

where,

$S_1 \equiv \sum_{i=1}^{m} P_i$, i.e., the sum of the probabilities of the m paths taken one at a time.

S_2 = the sum of the probabilities of the $\binom{m}{2}$ intersections formed by taking the minimal paths two at a time.

S_3 = the sum of the probabilities of the $\binom{m}{3}$ intersections formed by taking three minimal paths at a time, etc.

S_m = only one term because it is the probability of the intersection of all paths.

3. Example

In the bridge network case shown in Fig. 2.3,

$$R = P(S) = P(AB \cup CD \cup AED \cup BEC) \tag{3}$$

Now,

$$P(AB) = P_aP_b; \qquad P(AED) = P_aP_dP_e;$$

$$P(CD) = P_cP_d; \qquad P(BEC) = P_bP_cP_e$$

$$P(AB \cup CD) = P_aP_b + P_cP_d - P_aP_bP_cP_d$$

$$P(AB \cup CD \cup AED) = P_aP_b + P_cP_d - P_aP_bP_cP_d + P_aP_dP_e - P_aP_bP_dP_e - P_aP_cP_dP_e + P_aP_bP_cP_dP_e$$

Now,

$$\begin{aligned} R &= P(AB \cup CD \cup AED \cup BEC) \\ &= P_aP_b + P_cP_d + P_aP_dP_e + P_bP_cP_e - P_aP_bP_cP_d - P_aP_bP_dP_e \\ &\quad - P_aP_cP_dP_e - P_aP_bP_cP_e - P_bP_cP_dP_e + 2P_aP_bP_cP_dP_e \end{aligned} \tag{4}$$

In a slight variation of this method, reliability expression could also be evaluated as;

$$R = 1 - [\{1 - P(AB)\}\{1 - P(CD)\}\{1 - P(AED)\}\{1 - P(BEC)\}]. \tag{5}$$

A generalization of the above method can be done by using Poincare's method (inclusion-exclusion).

In this example,

$$S_1 = P_aP_b + P_cP_d + P_aP_dP_e + P_bP_cP_e$$

$$S_2 = P_aP_bP_cP_d + P_aP_bP_dP_e + P_aP_bP_cP_e + P_aP_cP_dP_e$$
$$+ P_bP_cP_dP_e + P_aP_bP_cP_dP_e$$

$$S_3 = P_aP_bP_cP_dP_e + P_aP_bP_cP_dP_e + P_aP_bP_cP_dP_e + P_aP_bP_cP_dP_e$$

$$S_4 = P_aP_bP_cP_dP_e$$

Substituting the values of S_1 through S_4 in (2), (4) immediately follows.

4. Conclusions

Reliability expression (4) is equivalent to that obtained by Exhaustive Search method but is much simpler and requires only 26 multiplications as compared to 64 in the Exhaustive search case.

3.2 Bayes' Theorem Applied to Complex System Reliability

1. Introduction

In a complex system where the system elements are not in a purely parallel or series configuration the reliability can be evaluated by using Bayes' theorem involving conditional probabilities [14].

2. Statement of the Problem and the Computational Procedure

In solving such problem, a simplified form of Bayes' probability theorem is used to decompose the complex system into simple substructures. The theorem states that if A is an event that depends on one of two mutually exclusive events B_i and B_j of which one must necessarily occur, then the probability of occurrence of A is given by

$$P(A) = P(A, \text{ given } B_i) \cdot P(B_i) + P(A, \text{ given } B_j) \cdot P(B_j) \qquad (1)$$

Let Q_s represent the probability of system failure, R_k the probability that component K is good, and Q_k the probability that component K is bad. Then we obtain the following expression for system unreliability,

$$Q_s = Q_s(\text{given K is good}) \cdot R_k + Q_s(\text{given K is bad}) \cdot Q_k \qquad (2)$$

The corresponding system reliability R_s is

$$R_s = 1 - Q_s \qquad (3)$$

The assumption is made that the reliability of the components are independent each other.

3. Example

For applying this theorem to the reliability evaluation of the bridge network; (see Fig. 2.3) let A be the event of system success; B_i be the event of branch E being good and B_j be the event of branch E being bad.

$$R = P\{S\} = P(S/E) \cdot P_e + P(S/\bar{E}) \cdot (1 - P_e) \tag{4}$$

To evaluate P(S/E) and $P(S/\bar{E})$, the bridge network has been simplified as directed networks shown in Fig. 3.2.1 and Fig. 3.2.2 respectively. Figures 3.2.1 and 3.2.2 are purely series parallel networks; therefore, by inspection;

$$P(S/E) = (P_a + P_b - P_a P_c)\ (P_b + p_d - P_b P_d)$$

$$P(S/\bar{E}) = P_a P_b + P_c P_d - P_a P_b P_c P_d$$

Substituting these values in (4); the reliability expression is obtained.

4. Conclusion

A simplified form of Bayes' theorem using conditional probabilities can be used to decompose the complex system into simple substructures.

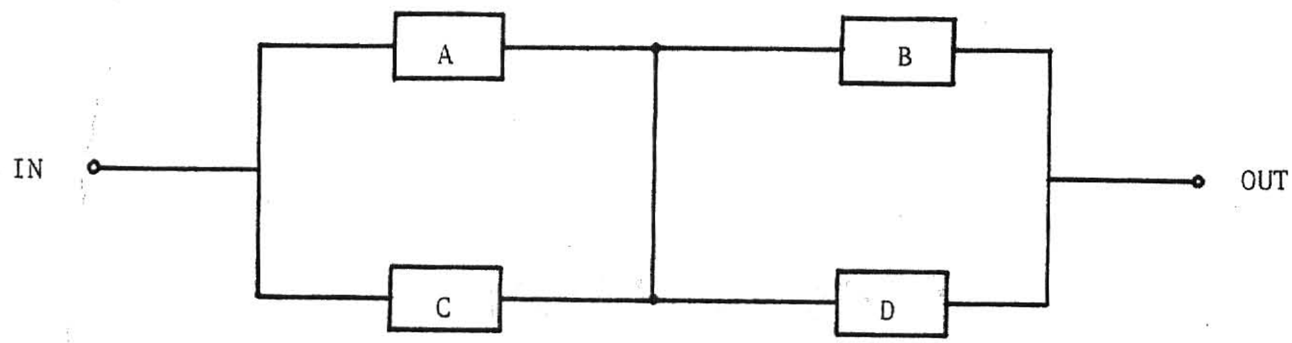


Fig. 3.2.1. System graph with branch E short.

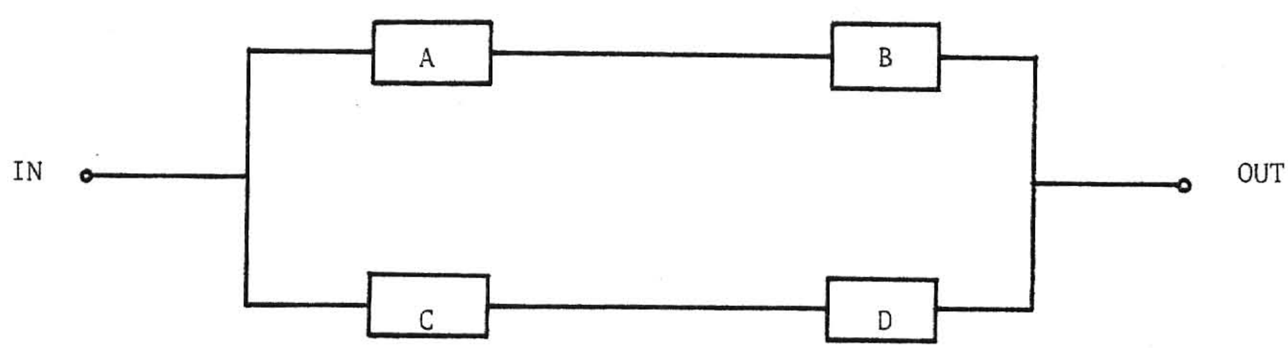


Fig. 3.2.2. System graph with branch E open.

3.3 Flow Graph Method Applied to Complex System Reliability

1. Introduction

A flow graph approach for reliability analysis is applied to the general case with elements in non-series-parallel combinations. The reliability of networks for elements with open or short failures is analyzed with flow graphs.

2. Flow Graph Method

In this method, all the branches in the network must be directed. Therefore, a given graph is decomposed into a number of graphs with all possible allowed directions of the interconnecting branches.

In a situation where an individual element can fail in either of the two ways, viz., open circuit or short circuit, the flow graph approach is very convenient.

Reliability in this method is given by:

$$R = F_0 - F_1 + F_2 - F_3 + \dots \tag{1}$$

where,

F_0 = Sum of the probabilities of all forward paths.

F_1 = Sum of the probabilities of all subgraphs with one loop.

F_2 = Sum of the probabilities of all subgraphs with two loops and so on.

3. Example

Take the example of the bridge network of Fig. 2.3. Element E is an interconnecting element and cannot be given fixed orientation.

Since element E may be oriented in either direction, two separate networks with all other elements having their orientation the same, except that of E being developed are shown in Fig. 3.3.1(a) and (b).

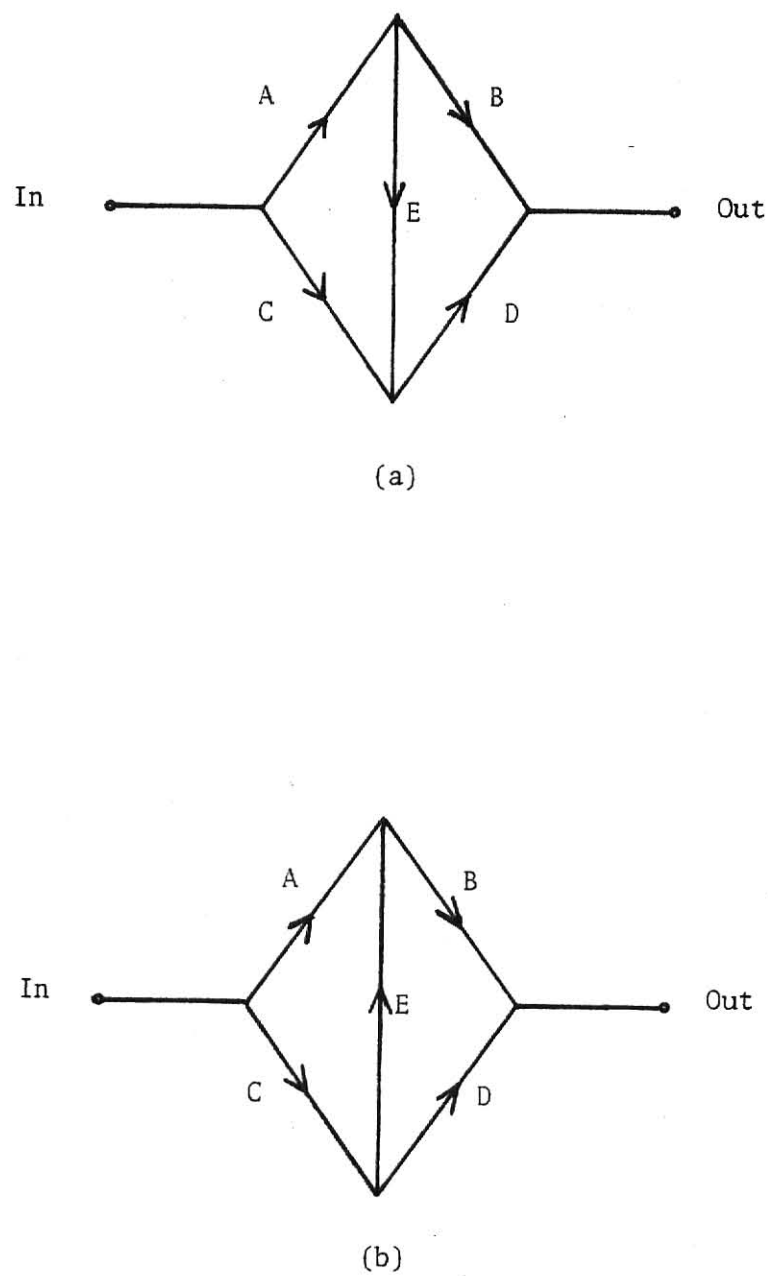


Fig. 3.3.1. Directed graphs of the system.

The solution of these two networks by graph theory are found separately.

Its two directed graphs are shown in Fig. 3.3.1. F_0, F_1, F_2 can be found from Figs. 3.3.2, 3.3.3 and 3.3.1 respectively.

$$F_0 = P_aP_b + P_cP_d + P_aP_dP_e + P_bP_cP_e$$

$$F_1 = P_aP_bP_dP_e + P_aP_bP_cP_d + P_aP_cP_dP_e$$
$$+ P_aP_bP_cP_e + P_bP_cP_dP_e$$

$$F_2 = P_aP_bP_cP_dP_e + P_aP_bP_cP_dP_e$$

Substituting these values in (1), the following expression is obtained.

$$R = Pr\{AB \cup CD \cup AED \cup BEC\}$$
$$= P_aP_b + P_cP_d + P_aP_dP_e + P_bP_cP_e - P_aP_bP_cP_d$$
$$- P_aP_bP_dP_e - P_aP_cP_dP_e - P_aP_bP_cP_e - P_bP_cP_dP_e$$
$$+ 2P_aP_bP_cP_dP_e$$

Consider another example of Fig. 3.3.4(a) consisting of three elements which can either open or short.

The flow diagram for paths of short failures is shown in Fig. 3.3.4(b). The event of failure of the network due to shorts is

$$S = 1_s2_s \cup 1_s3_s \qquad (2)$$

The flow diagram for cuts for the consideration of open circuit is given in Fig. 3.3.4(c). When applying the topological method it is often easier to write all possible paths; in that case, one can write down the cuts of a system by finding the paths of the dual network for the original network. Thus the event of failure due to opens of the elements is

$$0 = 1_0 \cup 2_03_0$$

The open and short events are mutually exclusive; so their probabilities add.

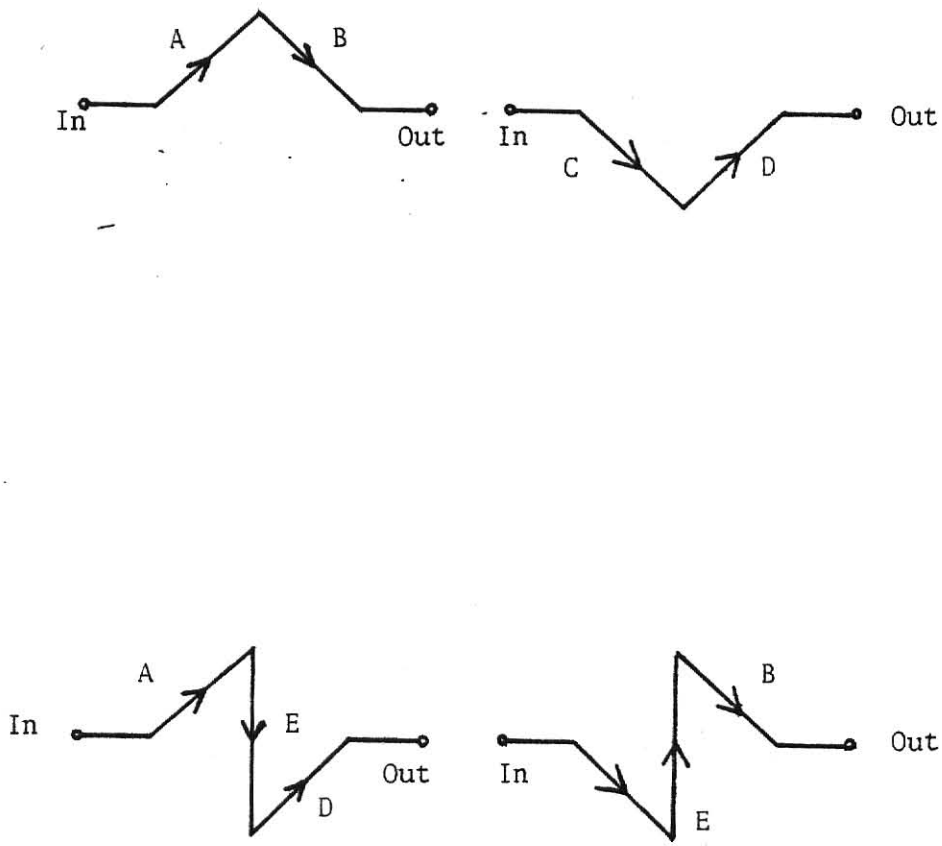


Fig. 3.3.2. Paths of the network.

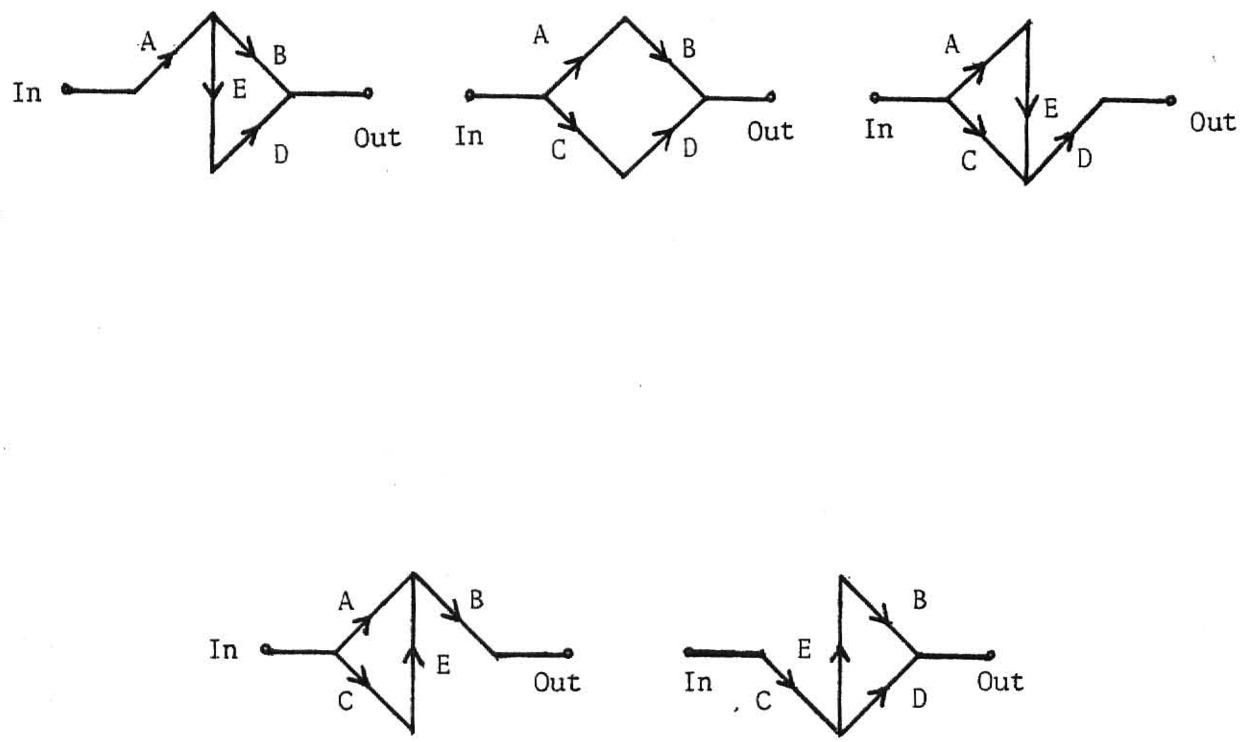


Fig. 3.3.3. Subgraphs with one loop.

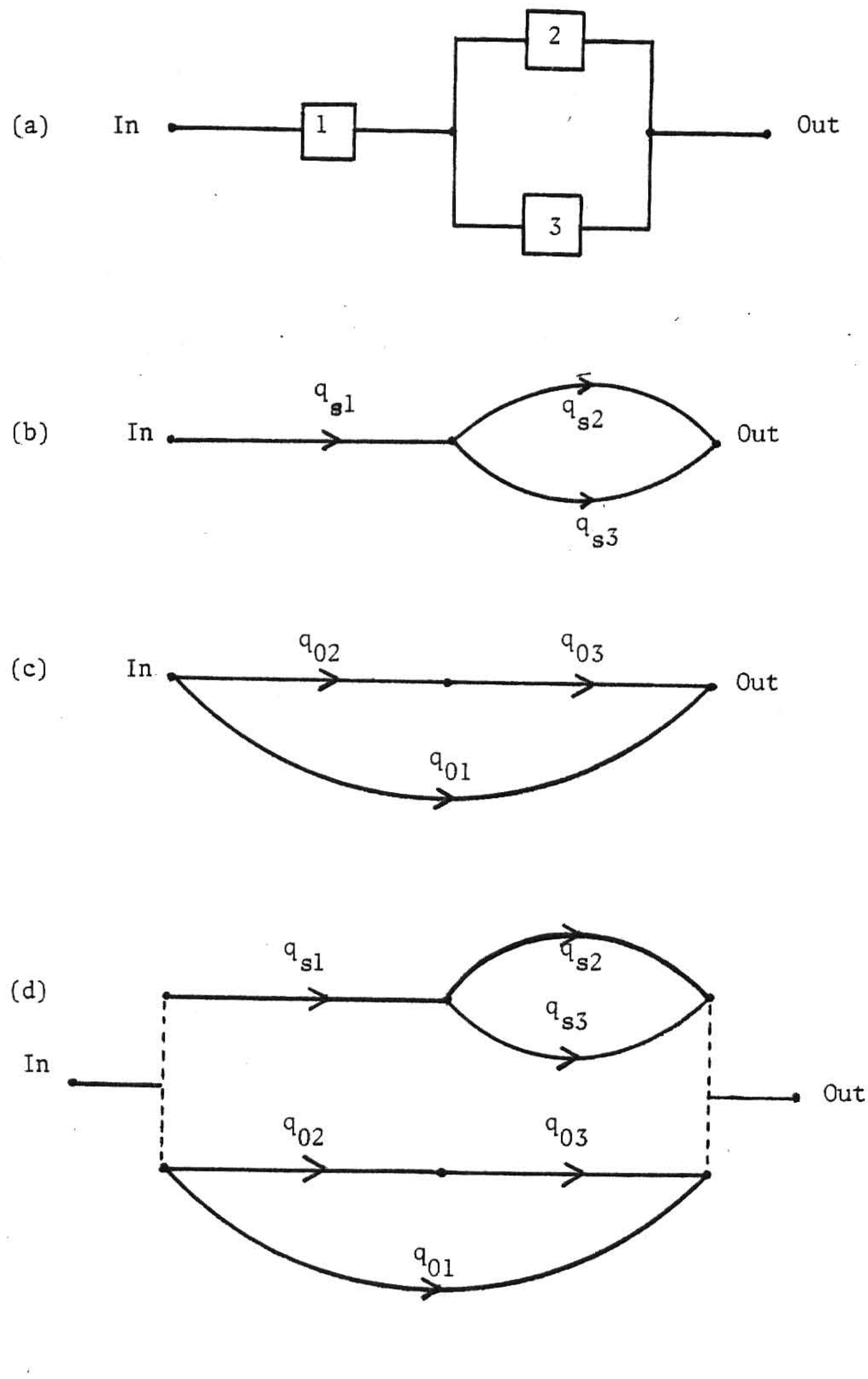


Fig. 3.3.4. Network of three elements.

$$q = q_s + q_0 = P\{S + 0\} = P(S) + P(O)$$

$$= (q_{1s}q_{2s} + q_{1s}q_{3s} - q_{1s}q_{2s}q_{3s}$$

$$+ (q_{10} + q_{20}q_{30} - q_{10}q_{20}q_{30}).$$

Graphically the situation is as shown in Fig. 3.3.4(d). Branch 1 of the diagraph considers the short failures and branch 2, the open ones.
Finally, the reliability of the system is $R = 1 - q$.

To distinguish between mutually exclusive events and otherwise in a diagraph we may use dotted lines for the former and firm lines for the latter. Such a situation is shown in the diagraph of Fig. 3.3.4(d). This approach will be found very convenient for complicated networks.

4. Conclusions

For large and complex systems (especially non-series-parallel networks) this approach is quite straight forward. The method given for the analysis of networks whose elements can short or open is easy to apply. This method can also be used directly for any network configuration.

NOTE:

Open and Short Circuit Failures

Consider a simple parallel unit composed of two elements, A and B, each of which can fail in either of two ways - open failure or short failure. Since a short in either of the two elements will result in unit failure, the assumption that individual path failure does not result in unit failure is not always true.

Reliability of Basic Parallel Configuration

Definition of Failure

For two elements in the active-parallel redundant configuration shown in Fig. 3.3.5 the unit will fail if 1) either A or B shorts, or 2) both A and B open.

Since events (1) and (2) are mutually exclusive, the event of failure due to shorts is

$$S = A_s \cup B_s$$

The event of failure due to opens of the elements is

$$0 = A_o B_o$$

the probability of unit failure is

$$q = q_s + q_o = P\{s + 0\} = P(S) + P(0)$$

$$= q_{as} + q_{bs} - q_{as}q_{bs} + q_{ao}q_{bo}$$

and the reliability is,

$$R = 1 - q = (1 - q_{as})(1 - q_{bs}) - q_{ao}q_{bo}$$

Reliability of Basic Series Configurations

The reliability of a series system in which both short-circuit and open failure are possible is estimated, with a two-element series unit used for illustration, shown in Fig. 3.3.6.

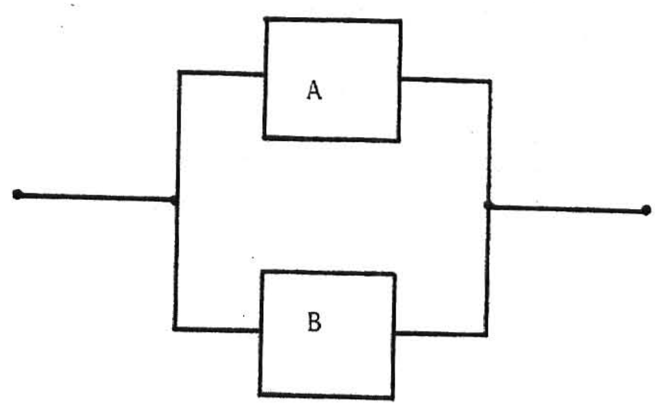


Fig. 3.3.5. Basic Parallel Configuration.

Fig. 3.3.6. Basic Series Configuration.

The unit will fail if 1) both A and B short, or 2) either A or B opens. Note that the definition of failure for a two-element parallel unit is exactly the opposite.

The respective events of these two failures are:

$$S = A_s B_s$$

$$0 = A_o \cup B_o$$

The probability of unit failure is

$$q = q_s + q_o = P\{S + 0\} = P(S) + P(0)$$

$$= q_{as} q_{bs} + q_{ao} + q_{bo} - q_{ao} q_{bo}$$

and the reliability is,

$$R = 1 - q = (1 - q_{ao})(i - q_{bo}) - q_{as} q_{bs}$$

3.4 Parametric Method Applied to Complex System Reliability

1. Introduction

In this approach, probability is treated as a Cartesian point and a parametric value is attached to it. Using the parametric values, the reliability of any complex structure can be easily evaluated. Evaluation of reliability in single-bridge and double-bridge networks can be done in a straight forward manner using the delta-star conversion technique.

2. Parametric Method

Let the probability of success of any event be x and probability of failure be $y \equiv 1 - x$. And the elements of the system here are assumed to be in two discrete states, either operation or failure. Introduce the parameters ϕ and θ defined by the relations

$$\phi \equiv \tan\theta \equiv y/x = (1-x)/x = y/(1-y).$$

Then

$$x = 1/(\phi+1) \text{ and } y = 1/(1+\cot\theta) \tag{1}$$

To evaluate the reliability of complex structures such as bridge networks, a delta-star conversion technique is introduced. The delta network and the equivalent star network are shown in Fig. 3.4.1.

One can go to C from A by traversing the series elements A and C in the star network. In the delta network, there are two parallel paths to go from A to C, through the element AC and through the elements AB and BC in series.

Therefore we have

(A and C) in series = [(AB and BC) in series] in parallel with AC

(A " B) " " = [(AC " BC) " "] " " " AB

(B " C) " " = [AB " AC) " "] " " " BC

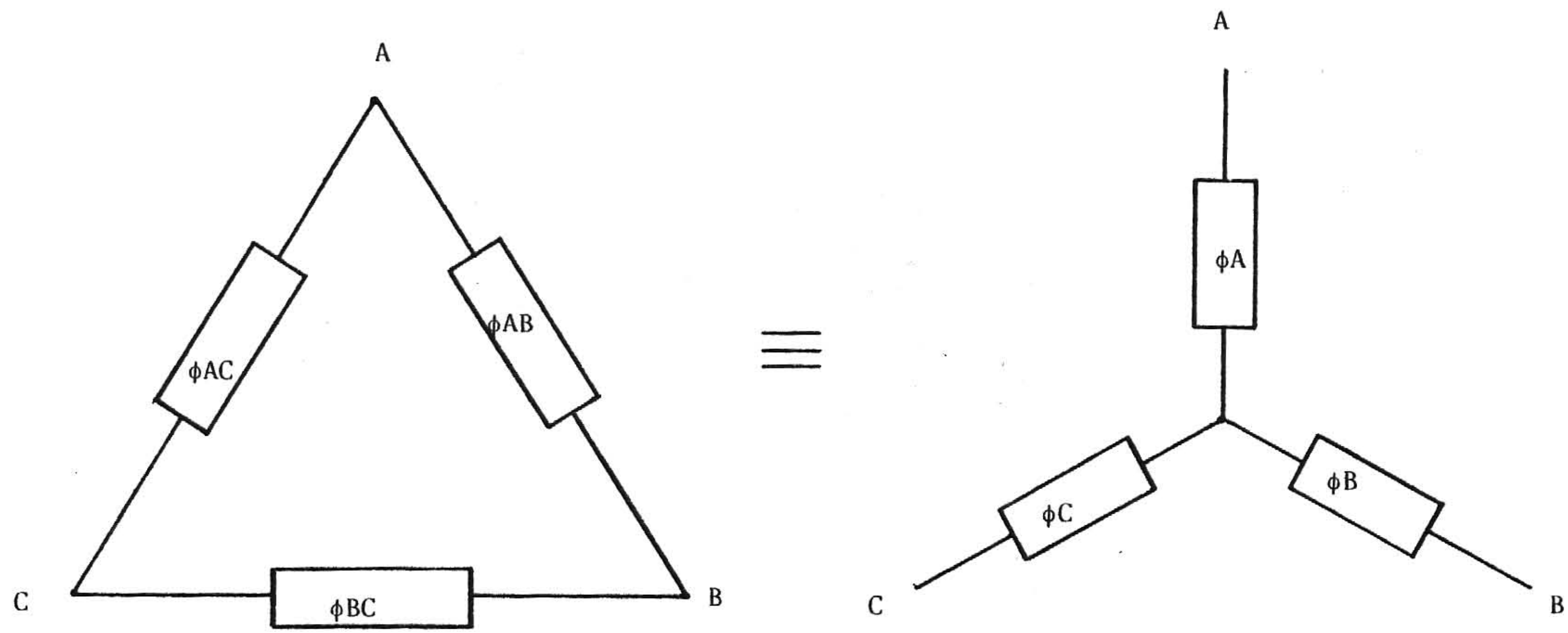


Fig. 3.4.1. Delta-star equivalent.

$$\phi_A + \phi_C = \frac{(\phi_{AB}+\phi_{BC})\ \phi_{AC}}{1+\phi}$$

$$\phi_A + \phi_B = \frac{(\phi_{AC}+\phi_{BC})\ \phi_{AB}}{1+\phi}$$

$$\phi_B + \phi_C = \frac{(\phi_{AB}+\phi_{AC})\ \phi_{BC}}{1+\phi} \qquad (2)$$

Solving (2), we get the following delta-star conversion equations:

$$\phi_A = \frac{\phi_{AC}\ \phi_{AB}}{1+\phi}\ , \quad \phi_B = \frac{\phi_{BC}\ \phi_{AB}}{1+\phi}\ , \quad \phi_C = \frac{\phi_{BC}\ \phi_{AC}}{1+\phi} \qquad (3)$$

If ϕ_{AB}, ϕ_{BC}, and ϕ_{AC} of the delta structure are known, the equivalent ϕ_A, ϕ_B, and ϕ_C of the star elements can be found using (3). Here the assumption is that we are dealing with components having a relatively high value of "x", i.e., $\phi << 1$.

3. Example

Example 1 Single-Bridge Network

The network is shown in Fig. 3.4.2(a). ϕ_1, ϕ_3, and ϕ_4 are in delta. The equivalent star values are ϕ_a, ϕ_b, and ϕ_c. The reliability is

$$R = 1/(1+\phi_g) \qquad (4)$$

where

$$\phi_g \equiv \phi_a + \phi_f + \phi_a \cdot \phi_f$$

$$\phi_a \approx \phi_1\phi_4/(1+\phi)$$

$$\phi_b \approx \phi_1\phi_3/(1+\phi)$$

$$\phi_c \approx \phi_3\phi_4/(1+\phi)$$

$$\phi \equiv \phi_1 + \phi_3 + \phi_4$$

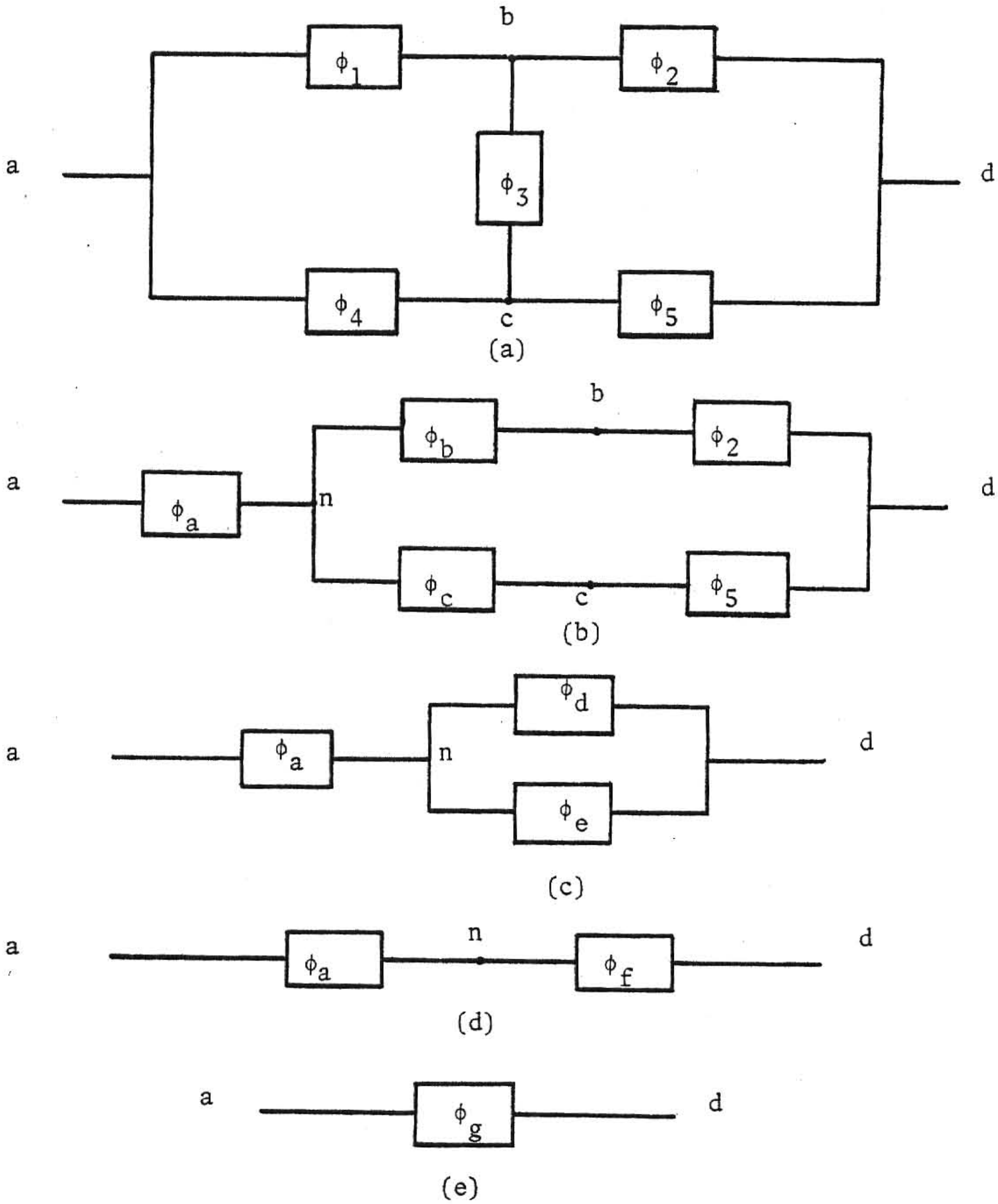


Fig. 3.4.2. Single-bridge network.

$$\phi_d \equiv \phi_b + \phi_2 + \phi_b \cdot \phi_2$$

$$\phi_e \equiv \phi_c + \phi_5 + \phi_c \cdot \phi_5$$

$$\phi_f \simeq \phi_d\phi_e/(1+\phi_d+\phi_e).$$

In the very high reliability region, ϕ_g is approximately

$$\phi_g \simeq \phi_1\phi_4 + \phi_f(1+\phi_1\phi_4) \qquad \phi_f = \phi_d \cdot \phi_e$$

$$= \phi_1\phi_4 + \phi_1\phi_3\phi_5 + \phi_2\phi_3\phi_4 + \phi_2\phi_5 \tag{5}$$

If element 3 has unit reliability ($\phi_3 = 0$), then

$$\phi_g = \phi_1\phi_4 + \phi_2\phi_5 \tag{6}$$

The values obtained by the classical method (event space method) and those by the parametric method are shown in Table 3.4.1 for several examples.

Example 2 Double-Bridge Network

The network is shown in Fig. 3.4.3(a). The network reduction using delta-star conversion is done as shown in Fig. 3.4.3(b) and 3.4.3(c). The reliability is

$$R = 1/(1+\phi_g).$$

where

$$\phi_g = \phi_a + \frac{(\phi_b+\phi_2+\phi_c)(\phi_d+\phi_5+\phi_e)}{1 + (\phi_b+\phi_2+\phi_c+\phi_d+\phi_5+\phi_e)} + \phi_f$$

Table 3.4.1

Single-Bridge Network

x_1, x_2, x_3, x_4, x_5					System Reliability	
					Classical Method	Parametric Method
0.999,	0.998,	0.997,	0.996,	0.995	0.9999859	0.9999859
0.99,	0.98,	0.91,	0.96,	0.97	0.9989056	0.9989211
0.98,	0.79,	0.96,	0.81,	0.97	0.9883914	0.9884202
0.86,	0.91,	0.90,	0.94,	0.88	0.9790449	0.9792424
0.94,	0.93,	0.92,	0.91,	0.90	0.9868051	0.9869310

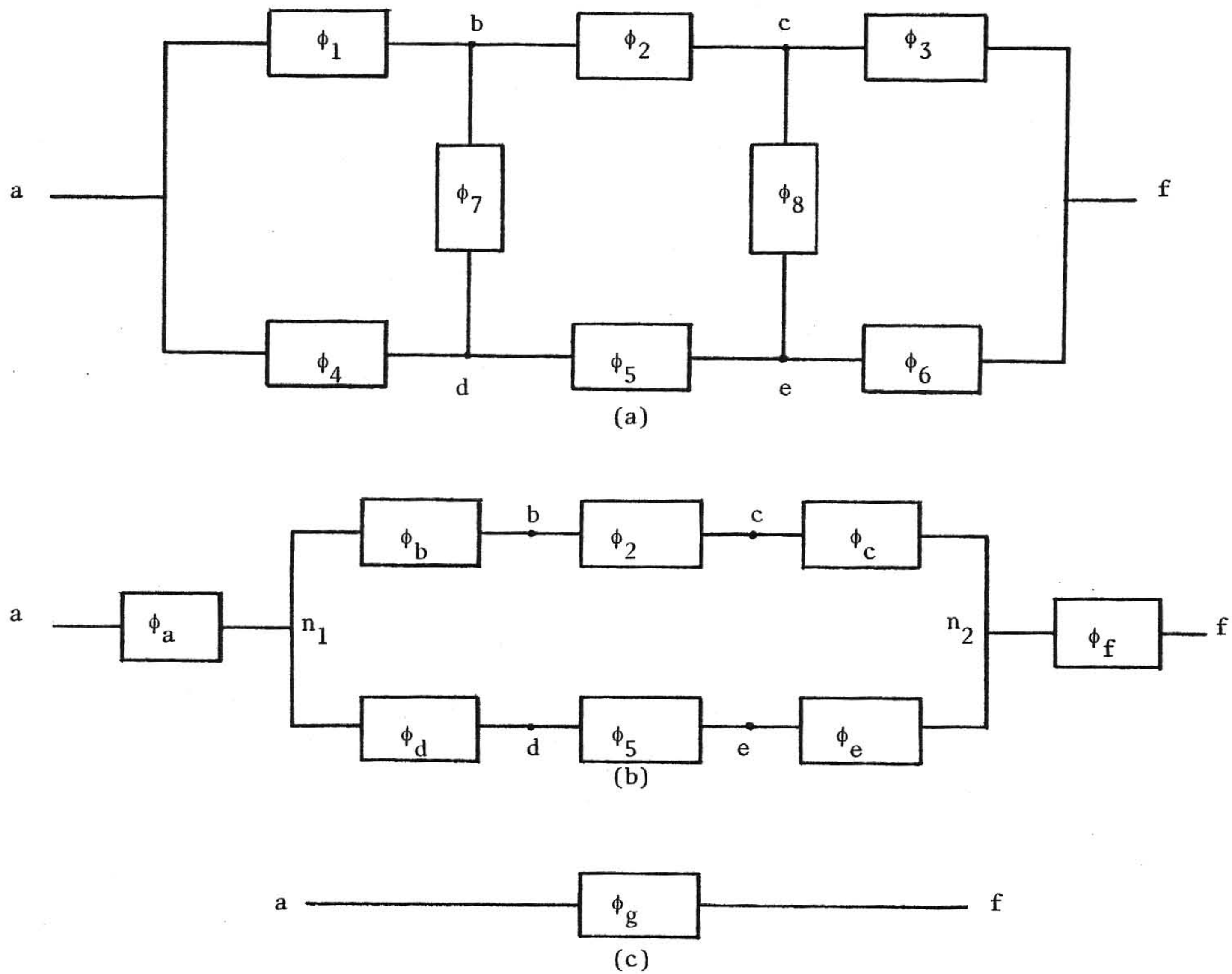


Fig. 3.4.3. Double-bridge network.

$$\left.\begin{aligned}\phi_a &= \frac{\phi_1\phi_4}{1+\phi}\\ \phi_b &= \phi_1\phi_7/(1+\phi)\\ \phi_d &= \phi_4\phi_7/(1+\phi)\end{aligned}\right\} \quad \text{where} \quad \phi = \phi_1 + \phi_2 + \phi_7$$

$$\left.\begin{aligned}\phi_c &= \phi_3\phi_8/(1+\phi)\\ \phi_e &= \phi_6\phi_8/(1+\phi)\\ \phi_f &= \phi_3\phi_6/(1+\phi)\end{aligned}\right\} \quad \text{where} \quad \phi = \phi_3 + \phi_6 + \phi_8$$

If the values of ϕ are very small, ϕ_g can be directly written as

$$\phi_g = \phi_1\phi_4 + \phi_3\phi_6 + (\phi_1\phi_7 + \phi_2 + \phi_3\phi_8)(\phi_4\phi_7 + \phi_5 + \phi_6\phi_8) \qquad (7)$$

Table 3.4.2 gives the values obtained by the classical method and this parametric method.

From Tables 3.4.1 and 3.4.2, it can be seen that the parametric method yields values that are a little higher than those given by the classical method.

4. Conclusions

The evaluation of system reliability using the parametric operator ϕ is simple and straight forward. The cumbersome process of identification of cut sets in the minimal cut set method or evaluation of probability corresponding to every conceivable state in the event space method is avoided.

The values obtained by the classical method and that obtained by this parametric method are in close agreement, provided that the approximations are reasonable.

Table 3.4.2

Double-Bridge Network

x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8				System Reliability	
				Classical Method	Parametric Method
0.999, 0.995,	0.998, 0.994,	0.997, 0.993,	0.996, 0.992	0.9999676	0.9999676
0.99, 0.95,	0.98, 0.94,	0.97, 0.93,	0.96, 0.92	0.9965101	0.9965768
0.96, 0.92,	0.95, 0.91,	0.94, 0.90,	0.93 0.89	0.9863162	0.9865903
0.94, 0.90,	0.93, 0.89,	0.92, 0.88,	0.91, 0.87	0.9758914	0.9763668
0.85, 0.85,	0.85, 0.85,	0.85, 0.85,	0.85, 0.85	0.9240378	0.9248903

NOTE:

Delta-star conversion

Technique: Physical meaning

Power System

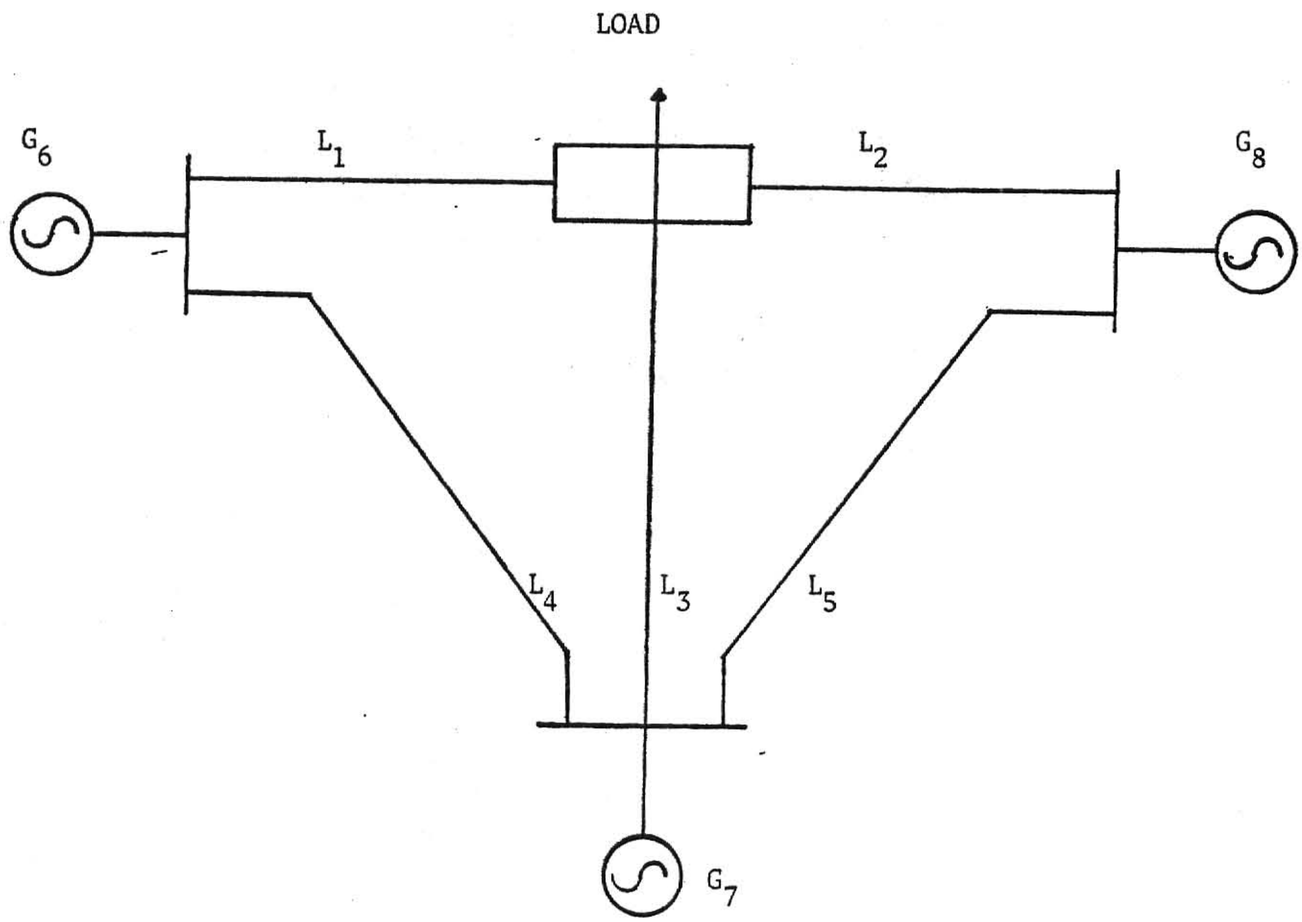


Fig. 3.4.4. Power system

G : Generator

L_1, L_2, L_3: lines which connects Generator and LOAD

L_4, L_5 : interconnector between Generators.

Apply the parametric method to a simple power system as shown in Fig. 3.4.4. The reliability diagram of the above problem is shown in Fig. 3.4.5(a). The network reduction proceeds, as shown in Fig. 3.4.5(b)-(e). The reliability is

$$R = 1/(1+\phi_p).$$

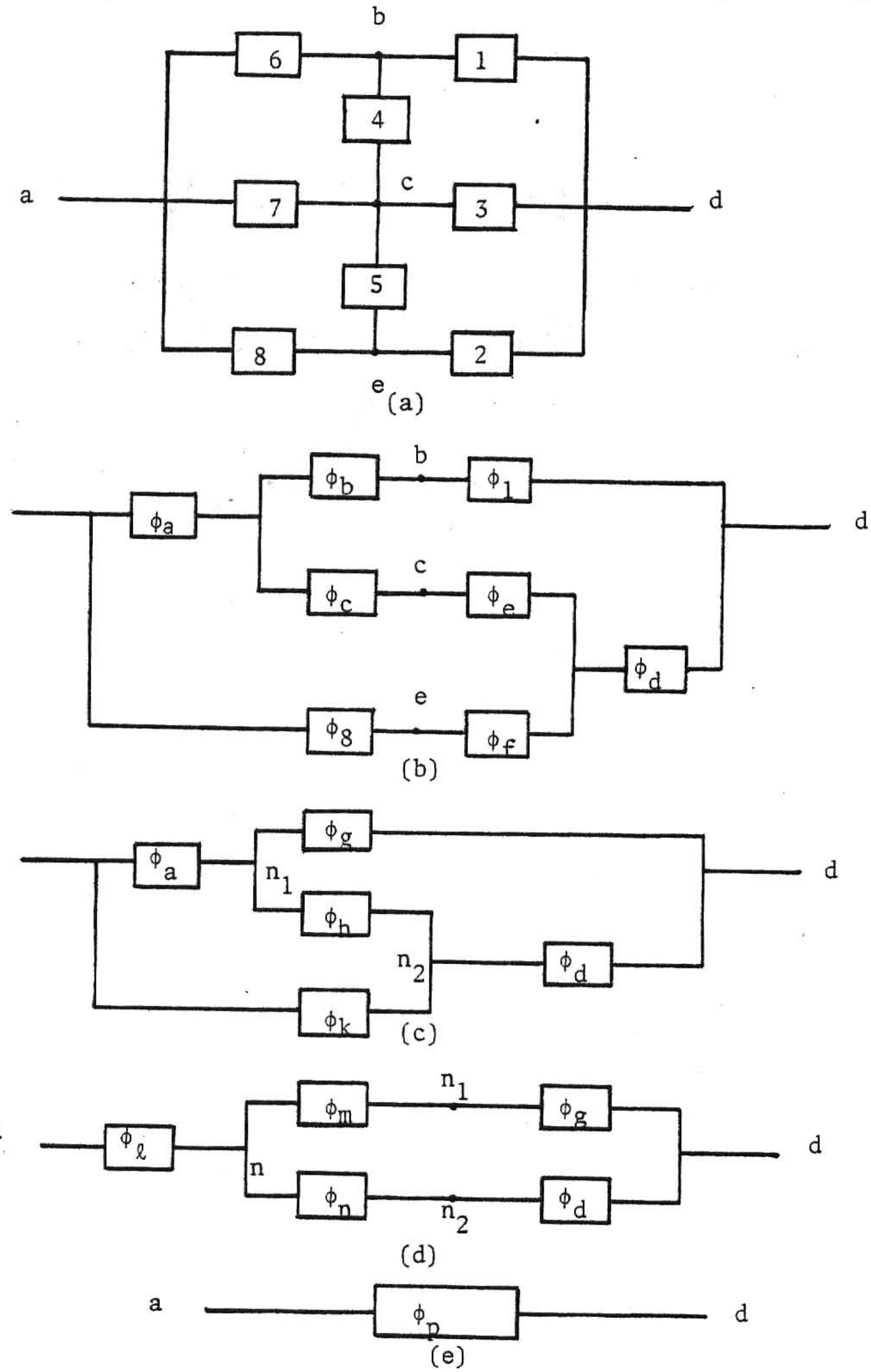


Fig. 3.4.5. Network reduction.

The reliability of lines L_1, L_2, and L_3 is 0.996.

The " of interconnectors L_4 and L_5 is 0.994.

The " of generators G_6, G_7, G_8 is 0.792 or 0.898

The system reliability

$$R_s = 0.9910 \quad \text{when} \quad x = 0.792$$

$$= 0.9989 \quad \text{when} \quad x = 0.898$$

3.5 Algebraic Extraction of Exclusive Terms in Complex System Reliability

1. Introduciton

In order to derive the reliability expression, probability calculus, Bayes' Theorem, Flow Graph method and parametric method are presented. But these method become very tedious in case the number of variables is large. In a general system, the number of elements and hence the number of variables becomes large in practically all cases. To avoid these limitations, an analytic method is presented [2, 5].

2. Formulation of the Problem

The method consists in rewriting S in this way that all its terms are mutually exclusive. To do this, paths are first arranged in a manner that the first path uses fewest literals, the next one more than that and so on. The first term is taken as it is. The second term is expanded about variables which have occurred in the first, but not in the second; and thus this term is rewritten such that it is disjoint with the first term. This is repeated for all the terms.

3. Example

In the bridge-network case, shown in Fig. 2.3,

$$S = AB \cup CD \cup BCE \cup ADE \tag{1}$$

Letting AB as it is, and expanding CD about A, we have:

$$CD = ACD \cup \bar{A}CD$$

Now, $\bar{A}CD$ is disjoint with AB and expanding ACD about B,

$$ACD = ABCD \cup A\bar{B}CD$$

Now, ABCD is contained in AB and $A\bar{B}CE$ is disjoint with AB. Therefore, disjoint portion of CD is:

$$CD(dis) = \bar{A}CD \cup A\bar{B}CD \quad (2)$$

Similarly

$$BCE(Dis) = \bar{A}BC\bar{D}E \quad (3)$$

$$ADE(Dis) = A\bar{B}\bar{C}DE \quad (4)$$

Substituting (2) through (4) in (1); we have

$$S(dis) = AB \cup \bar{A}CD \cup A\bar{B}CD \cup \bar{A}BC\bar{D}E \cup A\bar{B}\bar{C}DE \quad (5)$$

All the terms here are mutually disjoint. Therefore, reliability expression directly follows as:

$$R = p_a p_b + q_a p_c p_d + p_a q_b p_c p_d + q_a p_b p_c q_d p_e + p_a q_b q_c p_d p_e \quad (6)$$

4. Conclusions

The expression (6) is the same as the one derived by other method and has all the advantages but not the disadvantages of such as probability map method, etc.

3.6 Fast Algorithm Applied to Complex System Reliability

1. Introduction

An algorithm is developed to obtain a simplified reliability expression for a general network. In an attempt to simplify the reliability expression, Hurley [34] used a graphical method. An analytical method has been presented by Aggarwal, Gupta and Misra [2]. The algorithm described here gives the simplified reliability expression directly without the intermediate steps of [2].

2. Formulation of the Problem

If there are m success paths in a general network and their associated sets are $P_1, P_2, \ldots, p_m$; system success and reliability can be described by

$$S = P_1 \cup P_2 \cup \ldots \cup P_m \tag{1}$$

$$R \equiv P_r\{S\} = P_r\{P_1 \cup P_2 \cup \ldots \cup P_m\} \tag{2}$$

The sets p's are not disjoint (mutually exclusive); therefore,

$$R \not\equiv P_r\{P_1\} + \ldots + P_r\{P_m\} \tag{3}$$

If the P's are made disjoint --still retaining the property of system success --reliability is at once known.

The method for making P's disjoint is easy if paths are enumerated in such a way that <u>the path having fewest branches</u> is listed first and so on.

To select $P_{2,dis}$ from P_2, expand P_2 about a set K_1 (corresponding to a branch K_1) $\subset P_1$ not contained in P_2 as;

$$P_2 = (P_2 \cap K_1) \cup (P_2 \cap \bar{K}_1) \tag{4}$$

Now if $(P_2 \cap K_1) \subset P_1$; $(P_2 \cap K_1)$ is dropped from further consideration (because it is already included); otherwise it is further expanded about a set K_2 (corresponding to another branch K_2) and so on.

If

$$(P_2 \cap \bar{K}_1) \cap (P_1) = \Phi \tag{5}$$

subset $(P_2 \cap \bar{K}_1)$ is disjoint with P_1.

If however (5) is not true, this subset is further expanded about K_2 and so on.

Ultimately, we shall find all subsets of P_2 which are disjoint with P_1. Union of all these subsets if $P_{2,dis}$.

Similarly, we find $P_{j,dis}$ for all j such that $P_{j,dis} \cap P_i = \Phi$ for all $i < j$. This step is fastest if we first expand P_j about a set which contains maximum P_j's $(i < j)$, and so on. This corresponds to expanding about a branch which has occured in P_i's most often. Then,

$$S\ dis = \bigcup_{i=1}^{m} P_{i,dis} \tag{6}$$

The reliability is;

$$R = P_r\{S_{dis}\} = P_r\{\bigcup_{i=1}^{m} P_{i,dis}\}$$

$$= \bigcup_{i=1}^{m} P_r\{P_{i,dis}\} \tag{7}$$

The notations employed here are described as follows:

X	set indicating successful operation of branch x.
$\bar{X}$	set indicating unsuccessful operation of branch x.
Φ	null set.
p_x	reliability of branch x.
q_x	unreliability of branch x.
S	set indicating successful operation of system.
R	reliability of system: $\Pr\{S\}$.
b	number of branches in system.
m	number of paths in system.
P_i	set formed by the intersection of all sets which indicate successful operation of branches in path i.
E_i	vector corresponding to P_i.
T_j	vector $\sum_i E_i$ for all $i \le j$.
$E_i(K)$	modification of E_i where 0 in position k is replaced by 1 (indicates that an additional branch k is also successful).
$E_i(\bar{K})$	modification of E_i where 0 in position k is replaced by -1 (indicates that branch k is unsuccessful).
$P_{j,\,dis}$	subset of P_j such that $P_{j,\,dis} \cap P_i = \Phi$, for all $i<j$.
S_{dis}	modified S such that all terms are mutually disjoint.
adjacency	intersection (for sets indicating successful operation of a branch).

3. Computational procedure

The technique discussed in section 2 can be put in the following procedures.

1. Find all m path of the network. This can be done by a number of techniques.
2. Define a b dimensional vector $\underset{\sim}{E}_i$ ($i = 1,2,\ldots,m$) corresponding to P_i such that element k of this vector is 1 if the set $K \supset P_i$ and 0 otherwise.
3. Define $\underset{\sim}{T}_j \equiv \sum_{i \le j} \underset{\sim}{E}_i$; $i = 1,2,\ldots,m$ (8)
4. $P_{1,\ dis} = P_1$; $j = 1$ (Initialize) (9)
5. Let $j = j + 1$ (10)

   (A) If there are any nonzero entries in $\underset{\sim}{T}_j$ corresponding to zero entries in $\underset{\sim}{E}_j$, record their positions in order of their descending magnitude in $\underset{\sim}{T}_j$. Let these be $K_1, K_2, \ldots, K_r$

   This ordering helps in getting the minimal expression fast.

   (B) Decompose $\tilde{E}_j$ in two components $\underset{\sim}{E}_j(K_1)$ and $\underset{\sim}{E}_j(\bar{K}_1)$. This corresponds to expanding P_j about set K_1.

   $\underset{\sim}{E}_j(K_1)$ and $\underset{\sim}{E}_j(\bar{K}_1)$ are formed by replacing 0 in position K_1 in $\underset{\sim}{E}_j$ by 1 and -1 respectively.

   If $\underset{\sim}{E}_j(K_1)$ contains 1's in all the positions where there have been 1's in any $\underset{\sim}{E}_i$ $(i<j)$; then $\underset{\sim}{E}_j(K_1)$ is <u>DROPPED</u> from further analysis because it is already included in a previous path. If $\underset{\sim}{E}_j(\bar{K}_1)$ contains -1 in any position where there is 1 in $\underset{\sim}{E}_i$ for all $i<j$; then $\underset{\sim}{E}_j(\bar{K}_1)$ is <u>RETAINED</u> as a disjoint subset.

   If $\underset{\sim}{E}_j$ is not dropped and/or $\underset{\sim}{E}_j(\bar{K}_1)$ is not retained, then these are further decomposed about K_2 and so on, carrying out the dropping and retaining tests at each step. Union of the retained component of $\underset{\sim}{E}_j$ is $P_{j,\ dis}$.
6. If $j<m$, go to step 5.

7. Apply equation (7) to derive the reliability expression.

4. Example

Example 1

The bridge network case shown in Fig. 3.6.1 is taken as an example. The sets associated with the paths in this network are:

$P_1 = AB$; $P_2 = CD$; $P_3 = BCE$; $P_4 = ADE$.

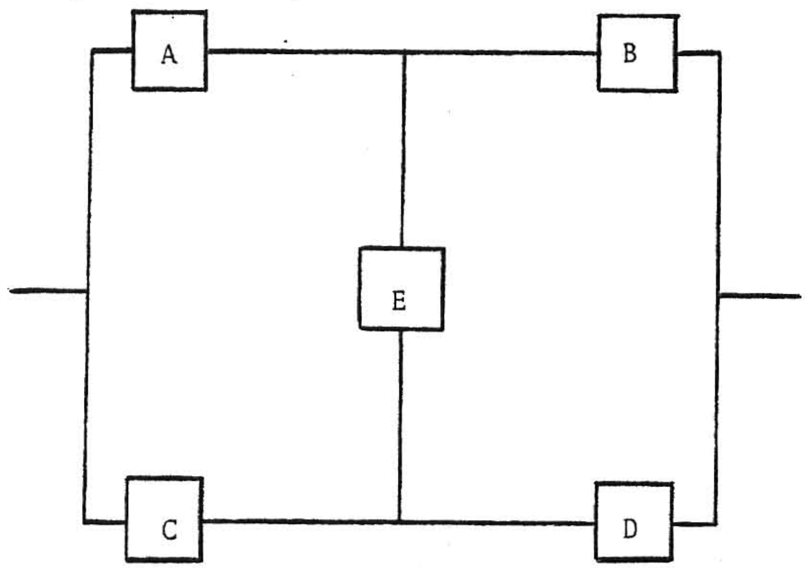


Fig. 3.6.1. A bridge network

Corresponding $\underset{\sim}{E}_i$'s and $\underset{\sim}{T}_j$'s are -

Branches ;	a	b	c	d	e
$\underset{\sim}{E}_1 = [$	1	1	0	0	0]
$\underset{\sim}{E}_2 = [$	0	0	1	1	0]
$\underset{\sim}{E}_3 = [$	0	1	1	0	1]
$\underset{\sim}{E}_4 = [$	1	0	0	1	1]
$\underset{\sim}{T}_1 = [$	1	1	0	0	0]
$\underset{\sim}{T}_2 = [$	1	1	1	1	0]
$\underset{\sim}{T}_3 = [$	1	2	2	1	1]
$\underset{\sim}{T}_4 = [$	2	2	2	2	2]

$$P_{1,\ dis} = P_1 = AB \tag{11}$$

Consider $\underset{\sim}{E}_2$ and $\underset{\sim}{T}_2$; $K_1 \equiv A$, which indicates expansion of $P_2 = CD$ about A; therefore,

$$\underset{\sim}{E}_2(A) = [\ 1 \quad 0 \quad 1 \quad 1 \quad 0\ ]$$

$$\underset{\sim}{E}_2(\bar{A}) = [-1 \quad 0 \quad 1 \quad 1 \quad 0\ ] \quad \text{Retain } \bar{A}CD \text{ (Disjoint subset)}$$

$\underset{\sim}{E}_2(A)$ indicates ACD while $\underset{\sim}{E}_2(\bar{A})$ indicates $\bar{A}CD$. Since $ACD \not\subset AB$, further decompose about $K_2^{(B)}$. But $(\bar{A}CD) \cap (AB) = \Phi$, therefore it is retained. Hence,

$$P_{2,\ dis} = \bar{A}CD \tag{12}$$

$$\underset{\sim}{E}_2(B|A) = [\ 1 \quad 1 \quad 1 \quad 1 \quad 0\ ] \quad \text{(ABCD) DROPPED}$$

$$\underset{\sim}{E}_2(B|A) = [\ 1 \quad -1 \quad 1 \quad 1 \quad 0\ ] \quad (A\bar{B}CD) \text{ Retain (Disjoint subset)}$$

$\underset{\sim}{E}_2(B|A) \rightarrow ABCD$ while $\underset{\sim}{E}_2(\bar{B}|A) \rightarrow A\bar{B}CD$
(since $ABCD \subset AB$,)
Drop ABCD

consider E_3 and T_3 ; $K_1 \equiv D$, $K_2 \equiv A$ $\left(BCE \xrightarrow{\text{expand}} \text{about D and A}\right.$ $\left.\because BCDE \subset CD\right)$

	a	b	c	d	e	
$E_3(D)$ = [	0	1	1	1	1]	BCDE DROP
$E_3(\bar{D})$ = [	0	1	1	-1	1]	$BC\bar{D}E$ CONTINUE
$E_3(\bar{D})(A)$ = [	1	1	1	-1	1]	$(ABC\bar{D}E)$ DROPPED
$E_3(\bar{D})(\bar{A})$ = [	-1	1	1	-1	1]	Retain

Hence,

$$P_{3,\ dis} = \bar{A}BC\bar{D}E \tag{13}$$

Consider E_4 and T_4 ; $K_1 \equiv C$, $K_2 \equiv B$

$$\begin{array}{lll} \underset{\sim}{E}_4(C) & = [1 \quad 0 \quad 1 \quad 1 \quad 1\,] & \text{DROP} \\ \underset{\sim}{E}_4(\bar{C}) & = [1 \quad 0 \quad -1 \quad 1 \quad 1\,] & \text{CONTINUE} \\ \underset{\sim}{E}_4(\bar{C})(B) & = [1 \quad 1 \quad -1 \quad 1 \quad 1\,] & \text{DROP} \\ \underset{\sim}{E}_4(\bar{C})(\bar{B}) & = [1 \quad -1 \quad -1 \quad 1 \quad 1\,] & \text{Retain} \end{array}$$

Hence,

$$P_{4,dis} = A\bar{B}\bar{C}DE \tag{14}$$

Using (7) and (11) to (14) ; the reliability is

$$R = P_r\{S_{dis}\} = P_r\{\bigcup_{i=1}^{m} P_{i,dis}\} = \sum_{i=1}^{m} P_r\{P_{i,dis}\}$$

$$= AB + A\bar{C}D + A\bar{B}CD + \bar{A}BC\bar{D}E + A\bar{B}\bar{C}DE.$$

Example 2

The algorithm is illustrated with an example from [2] given in Fig. 3.6.2.

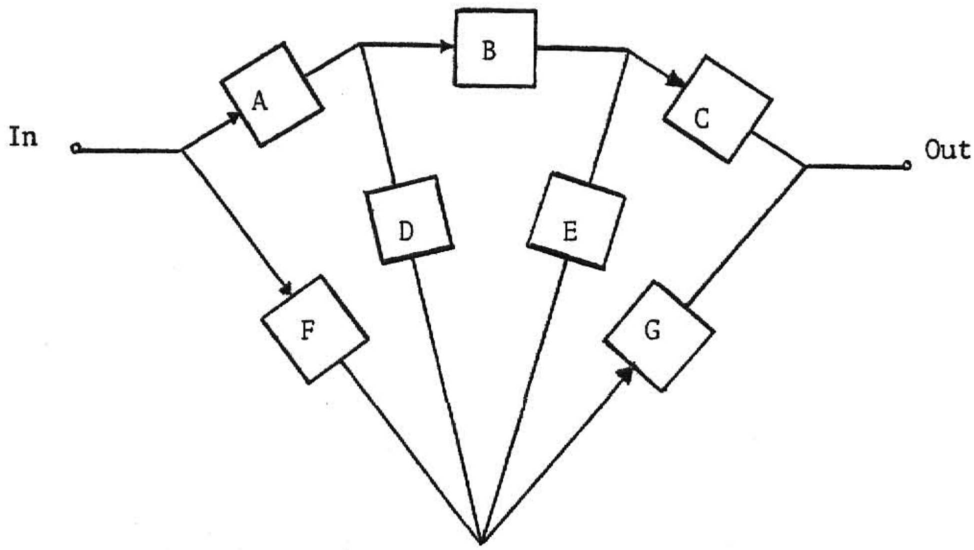


Fig. 3.6.2. A General Network

The sets associated with the paths in this network are: P_1 = FG; P_2 = ADG; P_3 = ABC; P_4 = CEF; P_5 = ABEG; P_6 = ACDE; P_7 = BCDF.

Corresponding E_i's and T_j's are -

Branches		a	b	c	d	e	f	g	
E_1	=	[0	0	0	0	0	1	1]	FG
E_2	=	[1	0	0	1	0	0	1]	ADG
E_3	=	[1	1	1	0	0	0	0]	
E_4	=	[0	0	1	0	1	1	0]	
E_5	=	[1	1	0	0	1	0	1]	
E_6	=	[1	0	1	1	1	0	0]	
E_7	=	[0	1	1	1	0	1	0]	
T_1	=	[0	0	0	0	0	1	1]	
T_2	=	[1	0	0	1	0	1	2]	
T_3	=	[2	1	1	1	0	1	2]	
T_4	=	[2	1	2	1	1	2	2]	
T_5	=	[3	2	2	1	2	2	3]	
T_6	=	[4	2	3	2	3	2	3]	
T_7	=	[4	3	4	3	3	3	3]	

$$p_{1,\ dis} = p_1 = FG \tag{15}$$

Consider E_2 and T_2; $K_1 \equiv F$, which indicates expansion of p_2 = ADG about F; therefore,

$E_2(F) = [\ 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad 1 \quad 1\ ]$ DROP ADFG

$E_2(\bar{F}) = [\ 1 \quad 0 \quad 0 \quad 1 \quad 0 \quad -1 \quad 1\ ]$ RETAIN $AD\bar{F}G$ (disjoint subset)

$E_2(F)$ indicates ADFG while $E_2(\bar{F})$ indicates $AD\bar{F}G$. Since $(ADFG) \subset (FG)$, $E_2(F)$ is dropped.

Also $(AD\bar{F}G) \cap (FG) = \Phi$, therefore it is retained. Hence,

$$p_{2,\ dis} = AD\bar{F}G \tag{16}$$

Consider E_3 and T_3; $K_1 \equiv G$, $K_2 \equiv F$, $K_3 \equiv D$.

	a	b	c	d	e	f	g	
$E_3(G)$ = [	1	1	1	0	0	0	1]	CONTINUE
$E_3(\bar{G})$ = [	1	1	1	0	0	0	-1]	RETAIN
$E_3(G)(F)$ = [	1	1	1	0	0	1	1]	DROP
$E_3(G)(\bar{F})$ = [	1	1	1	0	0	-1	1]	CONTINUE
$E_3(G)(\bar{F})(D)$ = [	1	1	1	1	0	-1	1]	DROP
$E_3(G)(\bar{F})(\bar{D})$ = [	1	1	1	-1	0	-1	1]	RETAIN

Therefore, $P_{3,\ dis} = (ABC\bar{G}) \cup (ABC\bar{D}\bar{F}G)$

$$= (ABC) \cap (\bar{G} \cup \bar{D}\bar{F}G) \tag{17}$$

Consider E_4 and T_4 ; $K_1 \equiv G$, $K_2 \equiv A$, $K_3 \equiv D$ and $K_4 \equiv B$

Branches		a	b	c	d	e	f	g	
$E_4(G)$	=	[0	0	1	0	1	1	1]	DROP
$E_4(\bar{G})$	=	[0	0	1	0	1	1	-1]	Continue
$E_4(\bar{G})(A)$	=	[1	0	1	0	1	1	-1]	Continue
$E_4(\bar{G})(\bar{A})$	=	[-1	0	1	0	1	1	-1]	Retain
$E_4(\bar{G})(A)(D)$	=	[1	0	1	1	1	1	-1]	Continue
$E_4(\bar{G})(A)(\bar{D})$	=	[1	0	1	-1	1	1	-1]	Continue
$E_4(\bar{G})(A)(D)(B)$	=	[1	1	1	1	1	1	-1]	DROP
$E_4(\bar{G})(A)(D)(\bar{B})$	=	[1	-1	1	1	1	1	-1]	Retain
$E_4(\bar{G})(A)(\bar{D})(B)$	=	[1	1	1	-1	1	1	-1]	DROP
$E_4(\bar{G})(A)(\bar{D})(\bar{B})$	=	[1	-1	1	-1	1	1	-1]	Retain

$$P_{4,dis} = \bar{A}CEF\bar{G} + A\bar{B}CDEF\bar{G} + A\bar{B}C\bar{D}EF\bar{G}$$

$$= CEF\bar{G}(\bar{A} + A\bar{B}) = CEF\bar{G} \cap (\bar{A} \cup A\bar{B}) \qquad (18)$$

Consider E_5 and T_5 ; $K_1 \equiv F$, $K_2 \equiv D$, $K_3 \equiv C$

Branches	a	b	c	d	e	f	g	
$E_5(F)$ =	1	1	0	0	1	1	1	DROP
$E_5(\bar{F})$ =	1	1	0	0	1	-1	1	CONTINUE
$E_5(\bar{F})(D)$ =	1	1	0	1	1	-1	1	DROP
$E_5(\bar{F})(\bar{D})$ =	1	1	0	-1	1	-1	1	CONTINUE
$E_5(\bar{F})(\bar{D})(C)$ =	1	1	1	-1	1	-1	1	DROP
$E_5(\bar{F})(\bar{D})(\bar{C})$ =	1	1	-1	-1	1	-1	1	Retain

$$P_{5,dis} = AB\bar{C}\bar{D}E\bar{F}G \qquad (19)$$

Consider E_6 and T_6 ; $K_1 \equiv G$, $K_2 \equiv F$, $K_3 \equiv B$

Branches	a	b	c	d	e	f	g	
$E_6(G)$ =	1	0	1	1	1	0	1	DROP
$E_6(\bar{G})$ =	1	0	1	1	1	0	-1	CONTINUE
$E_6(\bar{G})(F)$ =	1	0	1	1	1	1	-1	DROP
$E_6(\bar{G})(\bar{F})$ =	1	0	1	1	1	-1	-1	CONTINUE
$E_6(\bar{G})(\bar{F})(B)$ =	1	1	1	1	1	-1	-1	DROP
$E_6(\bar{G})(\bar{F})(\bar{B})$ =	1	-1	1	1	1	-1	-1	Retain

$$P_{6,dis} = A\bar{B}CED\bar{F}\bar{G} \qquad (20)$$

Consider E_7 and T_7 ; $K_1 \equiv A$, $K_2 \equiv E$, $K_3 \equiv G$

Branches		a	b	c	d	e	f	g	
$E_7(A)$	=	[1	1	1	1	0	1	0]	DROP
$E_7(\bar{A})$	=	[-1	1	1	1	0	1	0]	CONTINUE
$E_7(\bar{A})(E)$	=	[-1	1	1	1	1	1	0]	DROP
$E_7(\bar{A})(\bar{E})$	=	[-1	1	1	1	-1	1	0]	CONTINUE
$E_7(\bar{A})(\bar{E})(G)$	=	[-1	1	1	1	-1	1	1]	DROP
$E_7(\bar{A})(\bar{E})(\bar{G})$	=	[-1	1	1	1	-1	1	-1]	Retain

$$P_{7,dis} = \bar{A}BCD\bar{E}F\bar{G} \tag{21}$$

Using (7) and (13) to (19) ; the Reliability is :

$$\begin{aligned} R = P_r\{S_{dis}\} &= P_r\{\bigcup_{i=1}^{7} P_{i,dis}\} \\ &= P_fP_g + P_aP_dq_fP_g + P_aP_bP_c(q_g + q_dq_fP_g) \\ &\quad + P_cP_eP_fq_g(q_a + P_aq_b) + P_aP_bq_cq_dP_eq_fP_g \\ &\quad + P_aq_bP_cP_dP_eq_fq_g + q_aP_bP_cP_dq_eP_fq_g \end{aligned} \tag{22}$$

The above steps for selecting $P_{3,\ dis}$ from P_3 can be explained in terms of sets and Boolean operations as follows.

P_3 = ABC is expanded about G to give ABCG and ABC$\bar{G}$. (ABC$\bar{G}$ $\cap$ (P_i) = Φ (for i = 1,2); therefore it is retained. ABCG cannot be dropped and is therefore expanded about F as ABCFG and ABC$\bar{F}$G.

Now (ABCFG) $\subset$ (FG), therefore it is dropped. But ABC$\bar{F}$G cannot be retained. So, it is expanded about D as ABCD$\bar{F}$Gand ABC$\bar{D}\bar{F}$G; the former can be dropped and the latter can be retained.

4. Conclusions

The algorithm is easy and computationally economical. This is particularly so if the number of paths in the network is large. The resulting expression (20) in the example 2 is very simple and requires only 32 multiplications for numerical evaluation.

The same example when tried with other known techniques provides a reliability expression which requires a minimum of 118 multiplications for numerical evaluation. This algorithm is very effective when applied to Fig. 2.4 and Fig. 2.5.

3.7 Symbolic System Reliability Analysis Program (SYMRAP) Applied to Complex System Reliability.

1. Introduction

In system reliability analysis it is customary to represent the system by a probabilistic graph G in which each node and each branch (directed or undirected) has a given probability of being good [41].

This method makes no attempt to generate mutually exclusive events from the set of paths or cutsets but uses a technique to reduce greatly the number of terms in the reliability expression.

2. Formulation of the problem

This method depends very much on the systematic arrangement of variables and their subscripts. Therefore, it is vitally important to define the notation clearly.

Each branch (directed or undirected) and each node is an element of the probabilistic Graph G. Let there be a total of γ unreliable elements denoted by $X_1, X_2, \ldots, X_\gamma$.

Perfectly reliable elements will be denoted by symbols other than X. In addition the special pair of nodes under consideration are denoted by S and T.

Each element X_j is in one of two possible states; good (existence) and bad (nonexistence). Let there be m paths from S to T, called (S, T) paths, denoted by $P_1, P_2, \ldots, P_m$.

NOTATION

x_j	$\Pr\{X_j \text{ is good}\}$	(1)
y_j	$1 - x_j$	(2)
$(x_j, x_k)'$	$1 - x_j x_k$	(3)
$'$	prime, denotes complement	

p_j $\quad$ $Pr\{P_j\}$ = Reliability of P_j (4)

q_j $\quad$ $1 - P_j$ = unreliability of P_j (5)

$p_{i,j,k}$ $\quad$ $Pr\{P_iP_jP_k\}$ = joint probability that paths P_i and P_j and P_k are good (6)

$q_{i,j,k}$ $\quad$ $1 - P_{i,j,k}$ (7)

p_{st} $\quad$ (S, T) terminal-pair reliability

= Pr{at least one path from S to T is good}

$= Pr\{P_1 \cup P_2 \ldots \cup P_m\}$ (8)

To evaluate the r.h.s. of (8), we use the following theorem on the probability of the union of m events $A_1, A_2, \ldots, A_m$:

$$Pr\{A_1 \cup A_2 \cup A_3 \ldots \cup A_m\}$$

$$= [Pr\{A_1\} + Pr\{A_2\} + \ldots + Pr\{A_m\}] \leftarrow [{m \atop 1}] = m \text{ terms}$$

$$- [Pr\{A_1A_2\} + \ldots + \underset{i \neq j}{Pr\{A_iA_j\}}] \leftarrow [{m \atop 2}] \text{ terms}$$

$$+ [Pr\{A_1A_2A_3\} + \ldots + \underset{i \neq j \neq k}{Pr\{A_iA_jA_k\}}] \leftarrow [{m \atop 3}] \text{ terms}$$

$$\ldots$$

$$+ (-1)^{m-1} Pr\{A_1,A_2 \ldots A_m\} \leftarrow [{m \atop m}] = 1 \text{ term} \qquad (9)$$

For the special case m = 2, we have

$$Pr\{A_1 \cup A_2\} = Pr\{A_1\} + Pr\{A_2\} - Pr\{A_1A_2\} \qquad (10)$$

Almost every textbook of probability theory [26] proves (10), whereas the proof of the general case (9) is usually considered an easy extension. The number of terms in the explicit expression (9) is $2^m - 1$.

Let the events $(A_1, A_2, \ldots, A_m)$ in (9) be $(P_1, P_2, \ldots, P_m)$, and use the notation defined in (1) - (8). We obtain the following explicit expression for the terminal-pair reliability P_{ST} [27, 33].

$$P_{ST} = (S, T) \text{ terminal-pair reliab.}$$

$$= \Pr\{\text{at least one path from S to T is good}\}$$

$$= \Pr\{P_1 \cup P_2 \ldots \cup P_m\}$$

$$= \sum_{i=1}^{m} \Pr\{P_i\} - \sum_{i=1}^{m} \sum_{j>i}^{m} \Pr\{P_i \cap P_j\} + \sum_{i=1}^{m} \sum_{j>i}^{m} \sum_{k>j}^{m} \Pr\{P_i \cap P_j \cap P_k\}$$

$$+ --- + (-1)^{m-1} \Pr\{\bigcap_{i=1}^{m} P_i\}$$

$$= (P_1 + P_2 + \ldots) - (P_{1,2} + P_{2,3} + \ldots + \underset{i \neq j}{P_{ij}}) + (P_{1,2,3} + \ldots +$$

$$+ (P_{1,2,3} + \ldots + \underset{i \neq j \neq k}{P_{i,j,k}}) --- + (-1)^{m-1} P_{1,2,3,\ldots,m} \tag{11}$$

Under the assumption of S-independent element failures, each term in (11) can be expressed in terms of element probability as follows.

$$P_j \;\; = \underset{\text{all Xa in path } P_j}{\Pi\, xa} \qquad a \in P_i \tag{12}$$

a; the members of the i^{th} path

$$P_{i,j,k} = \underset{\substack{\text{all Xa in path } P_i \text{ or} \\ \text{path } P_j \text{ or path } P_k}}{\Pi\, xa} \qquad a \in P_i \cup P_j \cup P_k \tag{13}$$

Some simplified expressions for terminal-pair reliability are as follows:

$$P_{i(j,k)}\ Pr\{P_i | P_j P_k\} = \text{probability that } P_i \text{ is good under the condition that } P_j \text{ and } P_k \text{ are good} \quad (14)$$

$$q_{i(j,k)} = 1 - P_{i(j,k)} \quad (15)$$

For the special case of a probabilistic graph G having only 2 paths from S to T, (11) becomes

$$P_{ST} = P_1 + P_2 - P_{1,2} = P_1 + (1 - P_{1,2}/P_2)P_2$$

$$= P_1 + (1 - P_{1(2)})P_2 = P_1 + q_{1(2)}\ P_2. \quad (16)$$

Observe that (16) achieves a reduction of P_{ST} expression from 3 to 2 terms. By repeated application of such term-reduction technique, simplified expressions for graphs with 1 to 4 (S,T) paths are obtained and given in Table 3.7.1.

The algorithm for generating the reliability expression for the general case of m paths will be described in the next section in conjunction with some considerations in digital computer implementation of the method. Observe the following features in Table 3.7.1.

1. The number of terms in P_{ST} is doubled each time the number of paths is increased by 1.
2. The expression for (k+1) - paths contains all the terms for k-paths (plus the same number of additional terms).

For the general case of m paths, the explicit expression for P_{ST} by the present method has 2^{m-1} terms, which is about half of that contained in (9), the direct expansion theorem.

TABLE 3.7.1

Explicit Reliability Expressions

Number of Paths	Reliability Expression	Number of Terms
1	$p_{ST} = p_1$	1
2	$p_{ST} = p_1 + q_{1(2)}p_2$	2
3	$p_{ST} = p_1 + q_{1(2)}p_2 + q_{1(3)}p_3 - q_{1(2,3)}p_{2(3)}p_3$	4
4	$p_{ST} = p_1 + q_{1(2)}p_2 + q_{1(3)}p_3 - q_{1(2,3)}p_{2(3)}p_3$	
	$+ q_{1(4)}p_4 - q_{1(2,4)}p_{2(4)}p_4 - q_{1(3,4)}p_{3(4)}p_4$	
	$+ q_{1(2,3,4)}p_{2(3,4)}p_{3(4)}p_4$	8

This fact alone is not important when m is large. What makes the present methods useful is that with a suitable technique for generating the subscripts, the great majority of terms in P_{ST} need not be calculated at all (see Section 4).

From the definition (14), a typical factor $P_{i(j,k)}$ in Table 3.7.1 can be expressed in terms of element probabilities as follows.

$$P_{i(j,k)} \equiv Pr\{P_i | P_j P_k\} = \Pi xa \quad \text{all } Xa \text{ in path } P_i \text{ but not in path } P_j \text{ and not in path } P_k \tag{17}$$

and if no such Xa exists, then $p_{i(j,k)} = 1$ and $q_{i(j,k)} = 0$.

Some simple examples are illustrated with the use of the above reliability expression.

Example 1. The probabilistic graph G shown in Fig. 3.1.7; the nodes are perfectly reliable.

Since the graph is series-parallel with respect to terminal-pair (S,T), we can obtain the following answer by inspection [62]:

$$P_{ST} = x_3x_5 + (x_1 + x_2 - x_1x_2)\, x_4 - x_3x_5(x_1 + x_2 - x_1x_2)\, x_4 \tag{18}$$

In order to compare the results from various algorithm intended for non-series parallel networks, we shall disregard the series-parallel structure, and solve the probability by the general procedure.

There are 3(S,T) paths with the following unreliable elements;

P_1 $-X_1X_4$, P_2 - X_2X_4, P_3 - X_3X_5

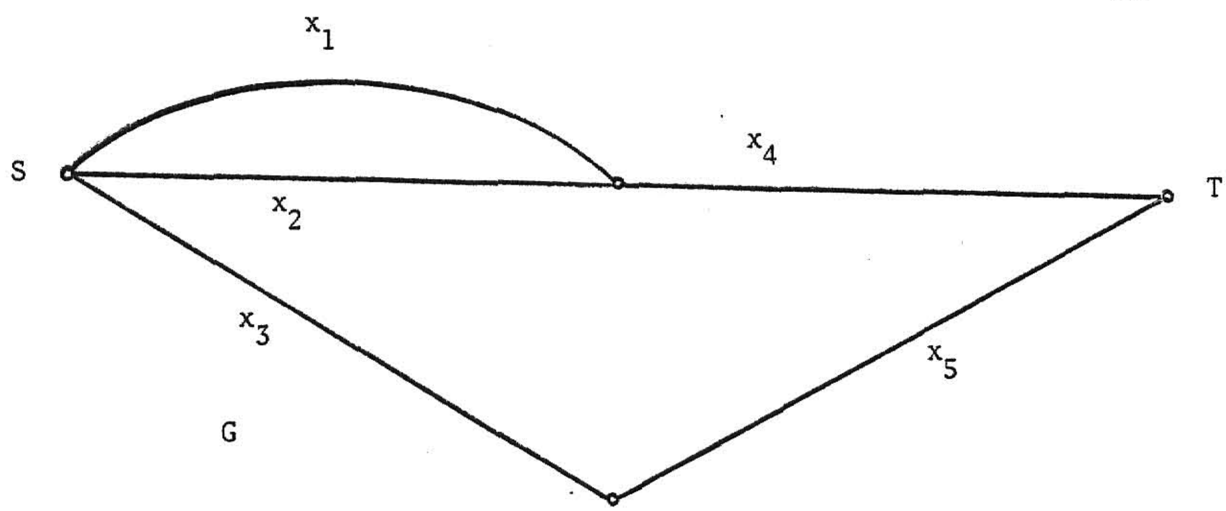


Fig. 3.7.1. A graph with 3 (S, T) paths.

Use the explicit expression for 3 paths in Table 3.7.1. We have, from (12) and (17)

$$p_1 = x_1x_4 \;,\; p_2 = x_2x_4,\; p_3 = x_3x_5,$$

$$p_{1(2)} = x_1,\; p_{1(3)} = x_1x_4,\; p_{2(3)} = x_2x_4,\; p_{1(2,3)} = x_1,$$

$$p_{ST} = x_1x_4 + x_1'x_2x_4 + (x_1x_4)'\; x_3x_5 - x_1'x_2x_4x_3x_5. \tag{19}$$

The direct application of the new method yields an expression consisting of 4 terms as shown in (19). This is to be compared with the following result obtained by the conventional path enumeration method given by (9) which yields 7 terms [33].

$$p_{ST} = x_1x_4 + x_2x_4 + x_3x_5 - x_1x_2x_4 - x_1x_3x_4x_5 - x_2x_3x_4x_5$$
$$+ x_1x_2x_3x_4x_5 \tag{20}$$

If the exhaustive search of successful states is used, then the answer will have 17 terms as shown in [33].

Example 2. The bridge network shown in Fig. 3.7.2; the nodes are perfectly reliable.

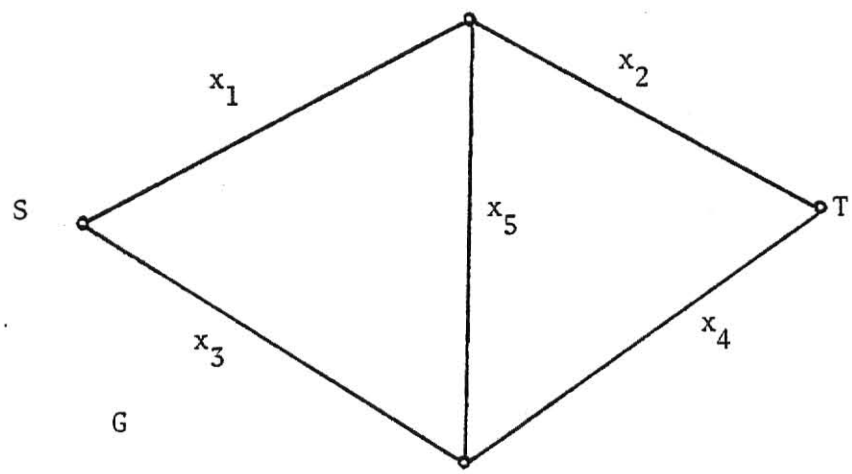


Fig. 3.7.2. A bridge network

Recall that explicit reliability expression for 4 paths in Table 3.7.1.

$$
\begin{aligned}
p_{ST} &= p_1 + q_{1(2)}p_2 + q_{1(3)}p_3 - q_{1(2,3)}p_{s(3)}p_3 + q_{1(4)}p_4 \\
&\quad - q_{1(2,4)}p_{2(4)}p_4 - q_{1(3,4)}p_{3(4)}p_4 \\
&\quad + q_{1(2,3,4)}p_{2(3,4)}p_{3(4)}p_4
\end{aligned}
$$

$$p_1 = x_1x_2,\ p_2 = x_3x_4,\ p_3 = x_1x_5x_4,\ p_4 = x_3x_5x_2,$$

$$p_{1(2)} = p_{1,2}/p_2 = x_1x_2,\ p_{1(3)} = p_{1,3}/p_3 = x_2,\ p_{1(2,3)} = p_{1,2,3}/p_{2,3}$$

$$= \frac{x_1x_2x_3x_4x_5}{x_1x_3x_4x_5} = x_2,$$

$$p_{2(3)} = p_{2,3}/p_3 = \frac{x_1x_3x_4x_5}{x_1x_5x_4} = x_3, \; p_{1(4)} = p_{1,4}/p_4 = \frac{x_1x_2x_3x_5}{x_2x_3x_5} = x_1,$$

$$p_{1(2,4)} = p_{1,2,4}/p_{2,4} = \frac{x_1x_2x_3x_4x_5}{x_2x_3x_4x_5} = x_1, \; p_{2(4)} = p_{2,4}/p_4 = \frac{x_2x_3x_4x_5}{x_2x_3x_5} = x_4,$$

$$p_{1(3,4)} = p_{1,3,4}/p_{3,4} = \frac{x_1x_2x_3x_4x_5}{x_1x_2x_3x_4x_5} = 1, \; p_{1(2,3,4)} = p_{1,2,3,4}/p_{2,3,4} = 1,$$

Since p_1 is good when p_3 and p_4 are good. Then $q_{1(3,4)} = 0$ and $q_{1(2,3,4)} = 0$ and the last two terms vanish. The remaining 6 terms lead to the answer

$$p_{ST} = x_1x_2 + (x_1x_2)'x_3x_4 + x_2'x_1x_5x_4 - x_2'x_3'x_1x_5x_4$$
$$+ x_1'x_3x_5x_2 - x_1'x_4'x_3x_5x_2 \qquad (21)$$

For this bridge network, the method of [32] also requires 6 terms. The answer obtained by the Boolean algebra method of [28] has 5 terms as follows:

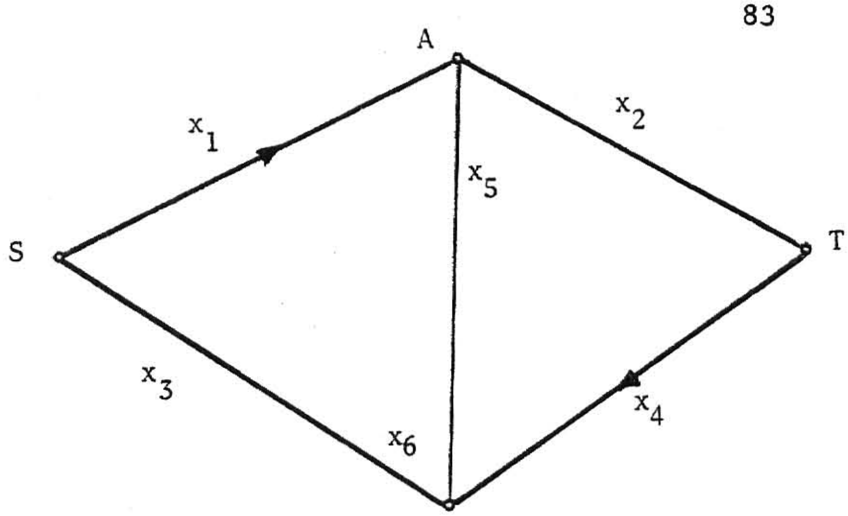
$$p_{ST} = x_1x_2 + x_1'x_3x_4 + x_1x_3'x_2'x_4 + x_1'x_3x_5x_2x_4' + x_1x_3'x_5x_2'x_4 \qquad (22)$$

However, one is cautioned not to draw conclusions on the merits of different methods based on just a few simple examples. If we group terms further, then (21) can be simplified by grouping terms 3 and 4, and grouping terms 5 and 6. The result is

$$p_{ST} = x_1x_2 + (x_1x_2)'x_3x_4 + x_2'x_3'x_1x_5x_4 + x_1'x_4'x_3x_5x_2 \qquad (23)$$

Ex. 3. The probabilistic graph G in Fig 3.7.3:

Fig. 3.7.3. A graph with unreliable nodes and branches.



G has both unreliable and perfectly reliable nodes, and has both directed and undirected branches. The 6 unreliable elements are labeled X_1 through X_6. There are two (S,T) paths with the following elements:

$$P_1 - SX_1AX_2T, \qquad P_2 - SX_3X_6X_5AX_2T.$$

Therefore,

$$p_1 = x_1x_2,\ p_2 = x_3x_6x_5x_2,\ p_{1(2)} = \frac{p_{1,2}}{p_2} = x_1,$$

From Table 3.7.1,

$$p_{ST} = p_1 + q_{1(2)}p_2 = x_1x_2 + x_1'x_3x_6x_5x_2 \tag{24}$$

3. Extension of the Cutset Method to Unreliable Nodes

In the preceding section, we have developed simplified reliability expressions based on paths. By a dual process, we can also obtain reliability expressions based on cutsets.

However, in order to handle the case of unreliable branches and nodes, we need the concept of 'feasible branch-node cutset' (a further generalization of the mixed cutset in [27]).

Definition: A feasible (S,T) branch-node cutset of a probabilistic graph G that can have both directed and undirected branches is a minimal set of unreliable elements, denoted by C, such that every (S,T) path has at least one element in common with C. For the sake of brevity, the single word cutset in here means a feasible (S,T) branch-node cutset.

To illustrate this concept, consider again the graph G shown in Fig. 3.7.3 in which the unreliable elements have been labeled X_1 through X_6. There are two (S,T) paths with the following elements:

$$P_1 - SX_1AX_2T, \qquad P_2 - SX_3X_6X_5AX_2T.$$

There are four cutsets with elements as follows:

$$C_1 - X_2, \quad C_2 - X_2X_3, \quad C_3 - X_1X_5, \quad C_4 - X_1X_6.$$

The set $\{X_2X_3X_5\}$ is disqualified as a cutset because it is not minimal. The set {A} is disqualified because it contains a perfectly reliable element. Had node A been unreliable (and labeled X_7), then the list would have one more cutset

$$C_5 - X_7.$$

Let there be m feasible (S,T) branch-node cutsets $C_1, C_2, \ldots, C_m$. A cutset C_j is said to be bad if all elements constituting C_j are bad. The same symbol C_j is also used to denote the event that the cutset C_j is bad.

NOTATION

c_j	$\Pr\{C_j\} = \Pr\{C_j\}$ is bad	(25)
d_j	$1 - c_j$	(26)

$c_{i,j,k}$ $\quad$ $\Pr\{C_i C_j C_k\}$ = joint probability that Cutsets C_i and C_j and C_k are bad (27)

$d_{i,j,k}$ $\quad$ $1 - c_{i,j,k}$ (28)

c_{ST} $\quad$ (S,T) terminal unreliability (29)

$\equiv$ Pr {at least one (S,T) Cutset is bad}

$= \Pr\{C_1 \cup C_2 \ldots \cup C_m\}$

From (29), it is obvious that

$$p_{ST} = 1 - c_{ST} \tag{30}$$

We can let the events $(A_1, A_2, \ldots, A_m)$ in (9) be $(C_1, C_2, \ldots, C_m)$, and use the notation defined in (25) - (29) to obtain an expression for C_{ST}, much the same as we obtain (11) for p_{ST}.

The same simplification technique can be used to reduce the number of terms in C_{ST}. A new set of relationships similar to those given in (11) - (17) and Table 1 will be obtained, the only difference being the following symbol and word changes.

P - C

p - c

q - d

x - y

path - Cutset

good - bad

reliability - unreliability

Ex. 4. $x_1 = 0.9$ $\quad$ $x_4 = 0.95$ $\quad$ $x_2 = 0.8$ $\quad$ $x_3 = 0.85$

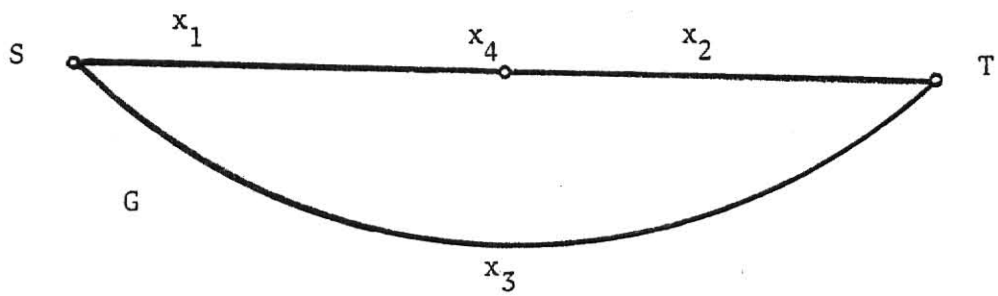


Fig. 3.7.4. An example for the Cutset method.

Consider the simple graph G with three unreliable branches and one unreliable node, shown in Fig. 3.7.4.

There are 3 cutsets;

$$C_1 = X_1X_3, \quad C_2 = X_4X_3, \quad C_3 = X_2X_3$$

C; feasible (S,T) branch node cutset of a graph G

From the duals of (12) and (17), we have

$$c_1 = y_1y_3, \quad c_2 = y_4y_3, \quad c_3 = y_2y_3,$$

$$c_{1(2)} = \frac{c_{1,2}}{c_2} = \frac{y_1y_3y_4}{y_4y_3} = y_1, \quad c_{1(3)} = \frac{y_1y_3y_2}{y_2y_3} = y_1$$

$$c_{2(3)} = \frac{c_{2,3}}{c_3} = \frac{y_4y_3y_2}{y_2y_3} = y_4, \quad c_{1(2,3)} = \frac{c_{1,2,3}}{C_{2,3}} = \frac{y_1y_2y_3y_4}{y_2y_3y_4} = y_1$$

Recall that (when number of paths are three in Table 3.7.1).

$$p_{ST} = p_1 + q_{1(2)}p_2 + q_{1(3)}p_3 - q_{1(2,3)}p_{2(3)}p_3$$

by a dual process

$$c_{ST} = c_1 + c_2 + c_3 - (c_{1,2} + c_{2,3} + c_{1,3}) + c_{1,2,3}$$

$$= c_1 + (1 - \frac{c_{1,2}}{c_2})c_2 + (1 - \frac{c_{1,3}}{c_3})c_3 + (1 - \frac{c_{1,2,3}}{c_{2,3}})c_{2,3}$$

$$= c_1 + d_{1(2)}c_2 + d_{1(3)}c_3 - d_{1(2,3)}c_{2(3)}c_3$$

$$= y_1y_3 + y_1'y_4y_3 + y_1'y_2y_3 - y_1'y_4y_2y_3 \qquad (31)$$

$$= (0.1)(0.15) + (0.9)(0.05)(0.15) + (0.9)(0.2)(0.15)$$

$$- (0.9)(0.05)(0.2)(0.15)$$

$$= 0.0474$$

(The sum of (32) and (33) is 1).

By path

$$P_1 = X_1X_4X_2, \quad P_2 = X_3$$

$$p_1 = x_1x_4x_2, \quad p_2 = x_3$$

$$p_{1(2)} = p_{1,2}/p_2 = \frac{x_1x_4x_2x_3}{x_3} = x_1x_4x_2$$

$$p_{ST} = p_1 + q_{1(2)}p_2 = x_1x_4x_2 + (x_1x_4x_2)'x_3$$

$$(= x_1x_4x_2 + (1 - x_1x_4x_2)x_3) \qquad (32)$$

$$= (0.9)(0.95)(0.8) + (1 - 0.9 \cdot 0.95 \cdot 0.8) \cdot 0.85$$

$= 0.684 + (1 - 0.684)\ 0.85$

$= 0.9526.$

It is easy to verify that the sum of the r.h.s.'s of (31) and (32) is 1.

This method of handling unreliable nodes on a cutset basis generally increases the number of cutsets. With the path method, the number of (S,T) paths is the same whether nodes are unreliable or perfectly reliable.

4. Algorithm for Generating the Reliability Expressions.

A careful examination of the terms in each expression of Table 3.7.1 reveals that the variables and subscripts possess a pattern from which we devise the following algorithm for obtaining the terminal-pair reliability expression.

The algorithm is given in step by step together with an example.

Consider any probabilistic graph G which has 4 paths from S to T, e.g., the graph of Fig. 3.7.2.

Step 0. There are m paths, $p_1, p_2, \ldots, p_m$ from S to T in the probabilistic graph G.

Step 1. Construct a directed gragh $\hat{G}$ with (m + 1) nodes; such that between any pair of nodes (i,j) in G, i < j, there is exactly one branch directed from i to j.

Then, in the example, $\hat{G}$ has 5 nodes; it is shown in Fig. 3.7.5. For such a simple graph, it is not difficult to find all paths from node 1 to node 5 by inspection.

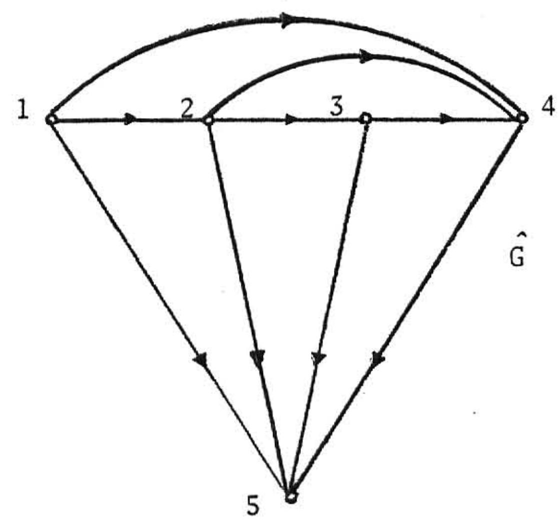


Fig. 3.7.5. The structure of $\hat{G}$ with 5 nodes.

Step 2. Find all paths in $\hat{G}$ from node 1 to node m + 1; there are M, $M \equiv 2^{m-1}$, such paths. Designate these paths by $\hat{P}_1$, $\hat{P}_2$, ---, $\hat{P}_M$

There are 8 paths with node sequences as follows:

$\hat{P}_1(1 - 5)$,

$\hat{P}_2(1 - 2 - 5)$, $\hat{P}_3(1 - 3 - 5)$, $\hat{P}_4(1 - 4 - 5)$,

$\hat{P}_5(1 - 2 - 3 - 5)$, $\hat{P}_6(1 - 2 - 4 - 5)$, $\hat{P}_7(1 - 3 - 4 - 5)$,

$\hat{P}_8(1 - 2 - 3 - 4 - 5)$. (35)

Step 3. For each path $\hat{P}\alpha$ with a sequence of nodes (always in ascending order owing to the structure of $\hat{G}$)

1, i, j, ..., k, ℓ, m + 1

we create a term $\hat{t}_\alpha$ as follows:

$$\hat{t}\alpha \equiv \begin{cases} p_1, \text{ if } \hat{P}\alpha \text{ has only two nodes (1 and m + 1)}, \\ (-1)^\beta q_1(i,j,\ldots,k,\ell)\ p_i(j,\ldots,k,\ell)\ \ldots\ p_{k(\ell)}p_\ell, \text{ otherwise} \end{cases} \tag{33}$$

where β is the total number of p's and q's in Eq. (33).

Using (33), we obtain 8 terms as follows:

$\hat{t}_1 = p_1, \ \hat{t}_2 = q_{1(2)}p_2,$

$\hat{t}_3 = q_{1(3)}p_3, \ \hat{t}_4 = q_{1(4)}p_4,$

$\hat{t}_5 = -q_{1(2,3)}p_{2(3)}p_3, \ \hat{t}_6 = -q_{1(2,4)}p_{2(4)}p_4,$

$$\hat{t}_7 = -q_{1(3,4)}p_{3(4)}p_4, \ \hat{t}_8 = q_{1(2,3,4)}p_{2(3,4)}p_{3(4)}p_4 \quad (36)$$

Step 4. The terminal-pair reliability expression is

$$p_{ST} = \Sigma_{\hat{G}} \hat{t}\alpha \quad (34)$$

($\Sigma_{\hat{G}}$; implies the sum over all α for which $\hat{p}_\alpha$ is in $\hat{G}$)

Substituting these terms ($\hat{t}_1$, $\hat{t}_2$, ..., $\hat{t}_8$) into (34), we obtain the expression for p_{ST} as given in Table 3.7.1 for 4 paths.

5. Examples

Table 3.7.2. SYMRAP Results for Example 5

S, source node	4
T, terminal node	9
m, number of paths from S to T	18
U, upper bound of number of terms in p_{ST}	131 072
Actual number of terms in p_{ST}	160
Computer time (CDC6500) for obtaining p_{ST} expression	79 s
Computer time for numerically evaluating p_{ST} from expression	0.65 s
Computer time for numerically evaluating p_{ST} from network	71 s

Ex 5. The graph shown in Fig. 2.11 represents a simple long distance telephone network. Let's find the reliability expression for the terminal pair (4,9). Facts about the results; Table 3.7.2 shows the results from SYMRAP.

1. There are 18 (S,T) paths.

   A directed expansion (11);

   $$2^{18} - 1 = 262143$$

   With new method, in the worst case p_{ST} (34)

   $$M = 2^{m-1} = 2^{17} = 131072 \text{ terms.}$$

   But because of the zero-valued q-factors, the actual number of terms is only 160.

2. The time required for obtaining the expression is 70 sec on a CDC 6500 computer. This is only slightly longer than 71 sec, the time required for a numerical case. Once the expression has been obtained, to substitute numerical values into the expression and find the result takes only 0.65 sec.

3. In actual practice, the network configuration will remain the same for a relatively long period of time (say one year). Suppose that we wish to calculate p_{ST} 100 times during that period with different element probabilities. The advantage of having an expression is obvious,

   $$79 + 100 \times 0.65 = 144 \text{ sec.}$$

   With numeric input and output only, the same results will require 100×71 sec $\cong$ 2 hrs.

4. Generally speaking, for $m \geq 10$ the time required to enumerate the m paths or m cutsets is only a very small fraction of the total computing time. It is advisable to find all paths and all cutsets and determine which approach to pursue. Unreliable nodes usually make the cutset method more complicated.

Example 6. Fig. 2.12 shows the reliability graph of a fairly complicated boiler safety system [33]. In using any reliability analysis program it is advisable for the user to do some preliminary simplification that will greatly reduce the computing time. These include combining series branches and parallel branches by the usual rules.

If the path method is used, series branches need not be combined. If the cutset method is used, parallel branches need not be combined. After such simplifications, the problem is reduced to that of finding p_{ST} for the probabilistic graph shown in Fig. 3.7.6.

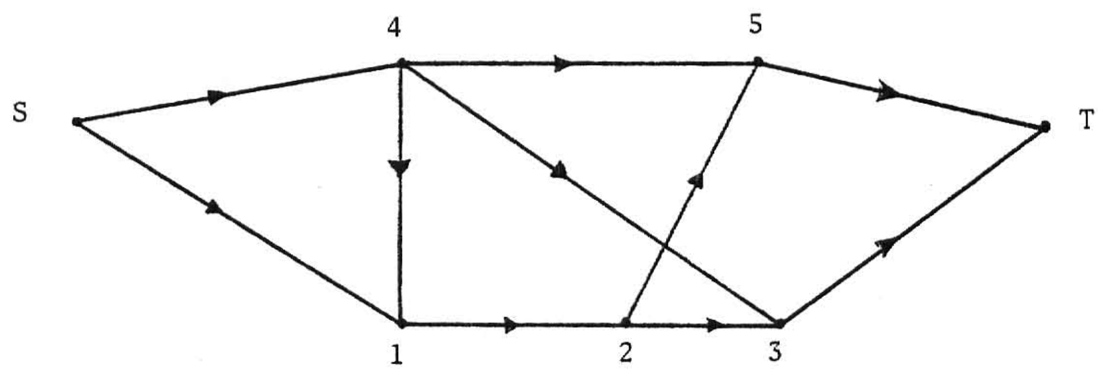


Fig. 3.7.6. Simplified representation of Fig. 2.12.

m, no. of paths from S to T, 6

U, upper bound of no. of terms in p_{ST}, 32

Actual no. of terms in p_{ST}, 20

Computer time (CDC 6500) for obtaining p_{ST} expression, 0.7 sec

6. Conclusions

With this new method, which also starts with path or cutset enumeration, can efficiently handle systems of moderate size (viz. system graph having fewer than 20 paths or cutsets between the (S,T) pair)

The number of terms in the reliability expression can be far below $2^m - 1$. (See Section 4) Even though symbolic reliability analysis is not applicable to large systems, there are many important systems in the real world that are small enough for the method to be fruitful [32,33].

Many systems have fixed configurations for a long period of time during which the element probabilities frequently vary. Symbolic reliability analysis performs the difficult task once and for all. For any given set of element probabilities, one need only substitute numerical values into the reliability expression and compute the result.

3.8 An Efficient Method Applied to Reliability Evaluation of a General Network.

1. Introduction

The techniques for reliability evaluation depend on the logic diagram of the system. Many methods have been discussed in the reliability literature to deal with non series-parallel networks. Some methods [16, 28] uses switching theory to compute the reliability expression. In these methods AND-OR expressed system-success function is the starting point. Then, suitably modified minimization techniques are applied to obtain an expression having all terms mutually exclusive. These techniques of generating disjoint terms require step by step testing for disjointness.

This efficient method deals with a technique for avoiding this test, although the number of terms in the reliability expression is the same as that in [6, 64]. The proposed method also applies to cutsets; another example is solved to determine unreliability expression for the network.

2. Formulation of the problem

To represent nodes (branches) in the reliability logic diagram, we shall use general Terminal Numbering Convention (TNC) [65]. In this convention the numbering of nodes (branches) begins at the source and continues in such a way that the output terminal of each branch (node) is assigned a number greater than the number used for its input, taking further care that each node (branch) is assigned a different number.

Using TNC, the first vertex n_1 represents the source and the last vertex n_k represents the sink. The notation and assumptions employed here are as follows:

k	total number of nodes
n_j	node j (j=1 is source)
b	total number of branches
m	total number of paths
'	Boolean negation
	Boolean intersection is denoted by juxtaposition
X_i, X_i'	successful and unsuccessful operation of branch x_i
p_i	probability of success (Reliability) of branch x_i
q_i	$1 - p_i$
S	system success function
[C]	connection matrix
E	Exclusive operator
R	terminal-pair reliability; probability that at least one path from n_1 to n_k is successful
Q	1 - R = terminal-pair unreliability

Assumptions

1. All nodes are perfectly reliable.
2. Each branch and the whole network each have two states: good or bad. Branch failures are s-independent.
3. The network is free from self-loops and directed cycles.

(1) Path Enumeration

The first step in most reliability evaluation techniques is to enumerate all minimum paths from n_1 to n_k in the reliability block diagram. Refs. [37, 38] show a method to find paths using the connection matrix. The connection matrix is defined as an analytic correspondence of the system graph and has a size k x k. An important property of this matrix is: Entry in $n_i n_j$ position of matrix $[C]^r$ gives all paths from n_i to n_j of size r.

In a connected graph of k nodes, the largest path will be of size (k-1); therefore, we can determine all paths in the network if we find $[c]^r$ where $r = 1,2,\ldots,k-1$. The main disadvantage of this method lies in the difficulty of repeated matrix multiplication. Therefore, a different method is proposed based on writing system success determinant $|s|$ from the knowledge of the connection matrix [c], and then expanding $|s|$.

Example 1. The procedure can be illustrated by the ARPA network of Fig. 3.8.1.

$$[c] = \begin{array}{c} \\ n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ n_6 \end{array} \begin{array}{c} \begin{array}{cccccc} n_1 & n_2 & n_3 & n_4 & n_5 & n_6 \end{array} \\ \begin{bmatrix} 0 & x_1 & x_2 & 0 & 0 & 0 \\ 0 & 0 & x_3 & 0 & x_4 & 0 \\ 0 & x_3 & 0 & x_6 & x_5 & 0 \\ 0 & 0 & 0 & 0 & x_7 & x_9 \\ 0 & 0 & x_5 & x_7 & 0 & x_8 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \qquad (1)$$

**Entry in $n_i n_j$ position of matrix $[c]^r$ (when $r = 6$, $n_3 n_4 = x_6$) gives all paths from n_i to n_j of size r.

Algorithm;

1. Write the connection matrix [c] for the logic graph of the network.

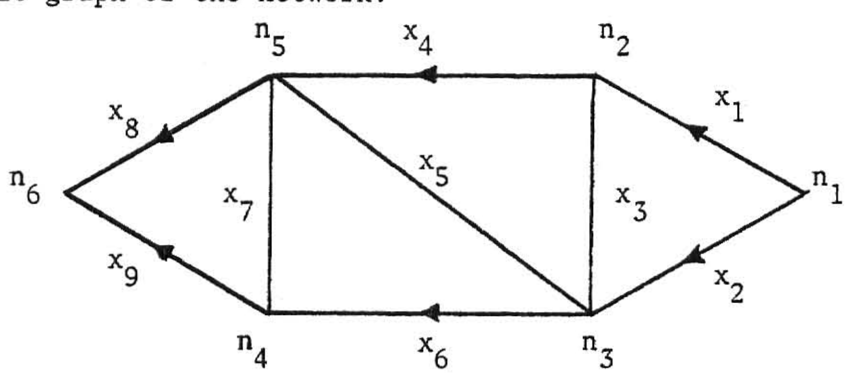


Fig. 3.8.1. Modified graph of ARPA network [55].

2. Add a diagonal unity matrix [U] of size p x p to the connection matrix [c].

$$[c] + \underset{pxp}{[U]} = \begin{array}{c c} & \begin{array}{cccccc} n_1 & n_2 & n_3 & n_4 & n_5 & n_6 \end{array} \\ \begin{array}{c} n_1 \\ n_2 \\ n_3 \\ n_4 \\ n_5 \\ n_6 \end{array} & \begin{bmatrix} 1 & x_1 & x_2 & 0 & 0 & 0 \\ 0 & 1 & x_3 & 0 & x_4 & 0 \\ 0 & x_3 & 1 & x_6 & x_5 & 0 \\ 0 & 0 & 0 & 1 & x_7 & x_9 \\ 0 & 0 & x_5 & x_7 & 1 & x_8 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

3. a. Remove the column corresponding to n_1 and the row corresponding to n_k in the matrix generated at step 2.

   b. Take remaining rows and columns and define the system success determinant $|s|$ of size (k-1). In this substep, all the algebraic variables are changed to the corresponding Boolean variables.

$$|S| = \begin{vmatrix} X_1 & X_2 & 0 & 0 & 0 \\ 1 & X_3 & 0 & X_4 & 0 \\ X_3 & 1 & X_6 & X_5 & 0 \\ 0 & 0 & 1 & X_7 & X_9 \\ 0 & X_5 & X_7 & 1 & X_8 \end{vmatrix} \qquad (2)$$

4. Expand the determinant $|S|$ using Boolean sum and product operations.

Expand (2) in accordance with step 4:

$$S = X_1X_3X_6X_7X_8 \cup X_1X_3X_6X_9 \cup X_1X_3X_5X_8 \cup X_1X_3X_5X_7X_9$$

$$\cup\ X_1X_4X_8 \cup X_1X_4X_7X_9 \cup X_1X_4X_5X_6X_9 \cup X_2X_6X_7X_8$$

$$\cup\ X_2X_6X_9 \cup X_2X_5X_8 \cup X_2X_5X_7X_9 \cup X_2X_3X_4X_8$$

$$\cup\ X_2X_3X_4X_7X_9. \tag{3}$$

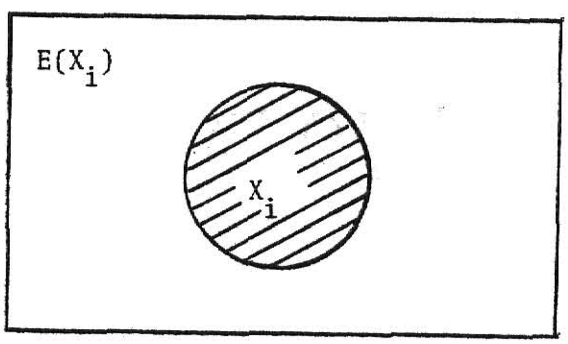
Hence 13 different minimal paths are obtained corresponding to the 13 terms in (3).

This method has an advantage of not requiring repeated matrix multiplications, but requires only that one determinant of size (k-1) be expanded. The method can easily be computerized [55].

(2) Exclusive Operator

Exclusive Operator E is one that operates on Boolean expressions as follows:

$$E(X_i) \equiv X_i' \tag{4a}$$



X_i; shaded area

$E(X_i)$; unshaded portion

Fig. 3.8.2. Venn diagram for $E(X_i)$.

$$E(F_1F_2 \ldots F_m) \equiv E(F_1) \cup F_1E(F_2) \cup \ldots \cup F_1F_2 \ldots F_{m-1}\, E(F_m) \tag{4b}$$

$$E(F_1 \cup F_2 \cup \dots \cup F_m) \equiv E(F_1)\ E(F_2)\ \dots\ E(F_m) \tag{4c}$$

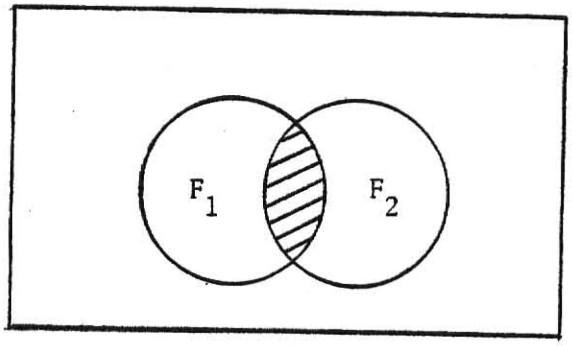


Fig. 3.8.3. Venn diagram for $E(F_1F_2) = E(F_1) \cup F_1E(F_2)$

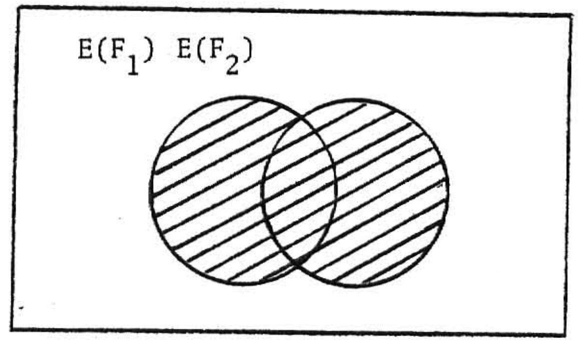


Fig. 3.8.4. Venn diagram for $E(F_1 \cup F_2) = E(F_1)E(F_2)$

properties (4a) through (4c) can be illustrated graphically by Venn diagrams. Fig. 3.8.2 represents (4a) on Venn diagram; X_i is shown by the shaded area while $E(X_i)$ is the unshaded portion.

Fig. 3.8.3 is the Venn diagram to illustrate (4b) for $m = 2$. $E(F_1)$ is the area outside F_1 in the universal set while $F_1E(F_2)$ is the portion of F_1 which is not included in F_2. This concept can easily be extended for $F_1, F_2, \dots, F_m$.

In a similar manner, (4c) is represented for $m = 2$ by Venn diagram in Fig. 3.8.4.

For a particular case when $F_i = X_i$, for all i, the above relations simplify to

$$E(X_1X_2 \dots X_m) = E(X_1) \cup X_1E(X_2) \cup \dots \cup X_1X_2 \dots X_{m-1}E(X_m)$$

$$= X_1' \cup X_1X_2' \cup \dots \cup X_1X_2 \dots X_{m-1}X_m' \tag{4d}$$

$$E(X_1 \cup X_2 \cup \dots \cup x_m) = E(X_1)\ E(X_2)\ \dots\ E(X_m)$$

$$= X_1'X_2' \dots X_m' \tag{4e}$$

It is obvious from (4d) that all conjunctive terms are mutually disjoint.

Example 2.

For $F_1 = X_1 \cup X_2$ and $F_2 = X_1X_2$

$$E(F_1F_2) = E(F_1) \cup F_1E(F_2) \quad ; \text{ from (4b)}$$

$$= E(X_1 \cup X_2) \cup (X_1 \cup X_2)\ E(X_1X_2)$$

$$= E(X_1)\ E(X_2) \cup (X_1 \cup X_2)\ E(X_1) \cup X_1E(X_2)$$

$$= X_1'X_2' \cup X_1X_2' \cup X_1'X_2 \quad ; \text{ from (4d) and (4e)}$$

3. Computational Procedure

The starting point can be either the system-success function or the system-failure function. The choice between the two depends on the number of paths or cutsets. The method consists in applying Exclusive operator on S, which results in all its terms being mutually disjoint. The paths are first arranged in ascending order of number of literals.

Recently, Bennetts [16] has used iterative minimization to obtain S (disjoint) from S. Nondisjoint pairs T_1, T_2 can be made mutually exclusive by using non-assigned literals and relative complement of T_2 with T_1. This technique of generating disjoint terms require a step by step testing algorithm for disjointness. This could limit this method in some cases, but often does not. To avoid this, the following method is presented.

1. Write down system success function as

$$S = T_1 \cup T_2 \cup \dots \cup T_m \tag{5}$$

where the T_i's represent minimal paths of the network.

Eq. (5) is directly obtained in the process of determining paths by the method proposed in section 2.

2. For each term T_i, $1 \le i \le m$,

F_i is defined to be the union of all predecessor terms $T_1, T_2, \ldots, T_{i-1}$ in which any literal that is present in both T_i and any of the predecessor terms is deleted from those predecessor terms, i.e.

$$F_i = T_1 \cup T_2 \cup \ldots \cup T_{i-1} \,\Big|\, \text{Each literal of } T_i \rightarrow 1 \tag{6}$$

In effect, the literals of T_i are assigned the Boolean value of 1 and this value is substituted in any predecessor term in which they occur. The resulting function F_i can be simplified using standard Boolean reduction identities.

3. Use Exclusive operator E (section 2), to get

$$S(\text{disjoint}) = T_1 \bigcup_{i=2}^{m} T_i E(F_i) \tag{7}$$

4. Change all logical variables into their analogous probability variables to get the reliability expression (all terms are mutually exclusive)

$$R = S(\text{disjoint}) \,\Big|\, X_i \rightarrow p_i,\ X_i' \rightarrow q_i \tag{8}$$

If we use source-terminal cutsets instead of paths in a particular system, then system failure function is obtained and processed similarly to derive system unreliability expression. The algorithm to generate S(disjoint) from S is

1. Set $i = 1$, S(disjoint) $= T_1$(Initialize)
2. Set $i = 2$
3. Obtain F_i; see (4)
4. Generate $E(F_i)$
5. S(disjoint) = S(disjoint) $\cup\ T_iE(F_i)$
6. If $i < m$; $i \leftarrow i + 1$; go to step 3.
7. Stop

4. Examples

Ex. 3. The ARPA network shown in Fig. 3.8.1 is used to illustrate the procedure for deriving the reliability expression. Rewriting (3) after arranging the terms properly results in

$$S = X_1X_4X_8 \cup X_2X_5X_8 \cup X_2X_6X_9 \cup X_1X_4X_7X_9 \cup X_1X_3X_6X_9$$
$$\cup X_1X_3X_5X_8 \cup X_2X_5X_7X_9 \cup X_2X_3X_4X_8 \cup X_2X_6X_7X_8 \cup X_1X_3X_6X_7X_8$$
$$\cup X_1X_3X_5X_7X_9 \cup X_1X_4X_5X_6X_9 \cup X_2X_3X_4X_7X_9 \tag{9}$$

On applying steps 2 and 3, the F_i's and $E(F_i)$'s for $i = 2, \ldots, 13$, are obtained as shown in Table 3.8.1.

From (8) the reliability expression is

$$R = p_1p_4p_8 + p_2p_5p_8(q_1 + p_1q_4)$$
$$+ p_2p_6p_9(q_8 + p_8q_5q_1 + p_8p_1q_4q_5)$$
$$+ p_1p_4p_7p_9(q_2q_8 + p_2q_6q_8) + p_1p_3p_6p_9(q_2q_4 + q_2p_4q_7q_8)$$
$$+ p_1p_3p_5p_8(q_2q_4q_6 + p_6q_2q_4q_9)$$

Table 3.8.1. Terms for Example 3.

F_i	$E(F_i)$	$T_iE(F_i)$
$F_2 = X_1X_4$	$X_1' \cup X_1X_4'$	$X_2X_5X_8(X_1' \cup X_1X_4')$
$F_3 = X_8(X_5 \cup X_1X_4)$	$X_8' \cup X_8X_5'(X_1' \cup X_1X_4')$	$X_2X_6X_9(X_8' \cup X_8X_5'X_1' \cup X_8X_1X_4'X_5')$
$F_4 = X_8 \cup X_2X_6$	$X_8'(X_2' \cup X_2X_6')$	$X_1X_4X_7X_9(X_8'X_2' \cup X_2X_6'X_8')$
$F_5 = X_2 \cup X_4(X_7 \cup X_8)$	$X_2'(X_4' \cup X_4X_7'X_8')$	$X_1X_3X_6X_9(X_2'X_4' \cup X_2'X_4X_7'X_8')$
$F_6 = X_2 \cup X_4 \cup X_6X_9$	$X_2'X_4'(X_6' \cup X_6X_9')$	$X_1X_3X_5X_8(X_2'X_4'X_6' \cup X_6X_2'X_4'X_9')$
$F_7 = X_6 \cup X_8 \cup X_1X_4$	$X_6'X_8'(X_1' \cup X_1X_4')$	$X_2X_5X_7X_9(X_1'X_6'X_8' \cup X_1X_4'X_6'X_8')$
$F_8 = X_1 \cup X_5 \cup X_6X_9$	$X_1'X_5'(X_6' \cup X_6X_9')$	$X_2X_3X_4X_8(X_1'X_5'X_6' \cup X_6X_1'X_5'X_9')$
$F_9 = X_5 \cup X_9 \cup X_4(X_1 \cup X_3)$	$X_5'X_9'(X_4' \cup X_4X_1'X_3')$	$X_2X_6X_7X_8(X_4'X_5'X_9' \cup X_4X_1'X_3'X_5'X_9')$
$F_{10} = X_2 \cup X_4 \cup X_5 \cup X_9$	$X_2'X_4'X_5'X_9'$	$X_1X_3X_6X_7X_8X_2'X_4'X_5'X_9'$
$F_{11} = X_2 \cup X_4 \cup X_6 \cup X_8$	$X_2'X_4'X_6'X_8'$	$X_1X_3X_5X_7X_9(X_2'X_4'X_6'X_8')$
$F_{12} = X_2 \cup X_3 \cup X_7 \cup X_8$	$X_2'X_3'X_7'X_8'$	$X_1X_4X_5X_6X_9(X_2'X_3'X_7'X_8')$
$F_{13} = X_1 \cup X_5 \cup X_6 \cup X_8$	$X_1'X_5'X_6'X_8'$	$X_2X_3X_4X_7X_9X_1'X_5'X_6'X_8'$

$$+ p_2p_5p_7p_9(q_1q_6q_8 + p_1q_4q_6q_8)$$

$$+ p_2p_3p_4p_8(q_1q_5q_6 + q_1q_5p_6q_9)$$

$$+ p_2p_6p_7p_8(q_5q_4q_9 + q_1q_3p_4q_5q_9) + p_1p_3p_6p_7p_8q_2q_4q_5q_9$$

$$+ p_1p_3p_5p_7p_9q_2q_4q_6q_8 + p_1p_4p_5p_6p_9q_2q_3q_7q_8$$

$$+ p_2p_3p_4p_7p_9q_1q_5q_6q_8 . \qquad (10)$$

Example 4.

For the reliability block diagram in Fig. 3.8.5, the cutsets are (x_1x_2, x_6x_7, $x_4x_5x_6$, $x_2x_3x_4$, $x_1x_3x_5x_6$, $x_2x_3x_5x_7$)

The system unreliability function is

$$S' = X_1'X_2' \cup X_6'X_7' \cup X_4'X_5'X_6' \cup X_2'X_3'X_4' \cup X_1'X_3'X_5'X_6' \cup X_2'X_3'X_5'X_7'$$

$$F_2 = X_1'X_2', \quad F_3 = X_1'X_2' + X_7', \quad F_4 = X_1' + X_6'X_7' + X_5'X_6'$$

$$= X_1' + X_6'(X_5' + X_7')$$

$$F_5 = X_2' + X_7' + X_4' + X_2'X_4'$$

$$= X_2' + X_4' + X_7'$$

$$F_6 = X_1' + X_6' + X_4'X_6' + X_4' + X_1'X_6'$$

$$= X_1' + X_4' + X_6'$$

Apply the definition of Exclusive operator to obtain

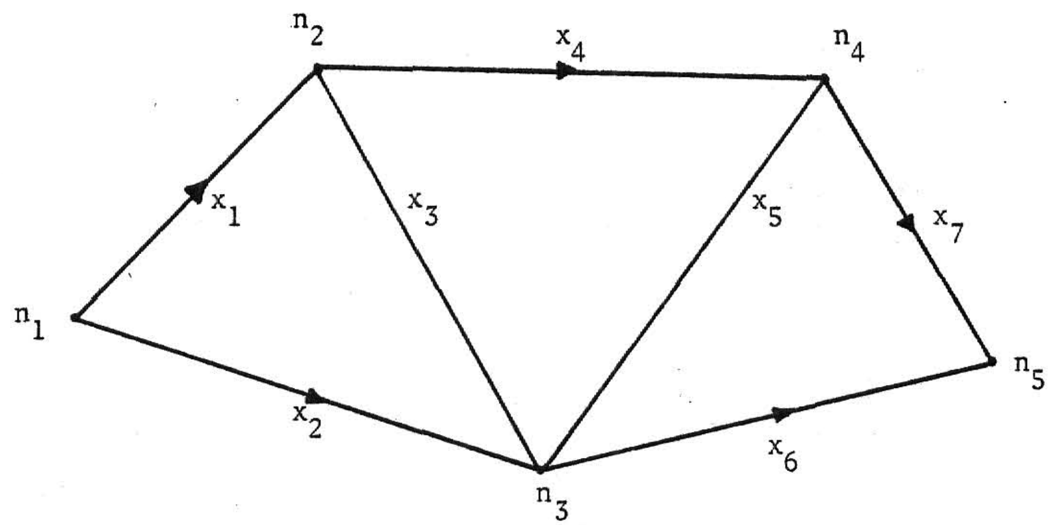


Fig. 3.8.5. A general nonseries parallel network.

$$E(F_2) = X_1 \cup X_1'X_2, \quad E(F_3) = X_7(X_1 \cup X_1'X_2),$$

$$E(F_4) = X_1X_6 \cup X_1X_6'X_5X_7, \quad E(F_5) = X_2X_4X_7$$

$$E(F_6) = X_1X_4X_6 \tag{11}$$

Use (7), (8), (11); the unreliability expression is

$$\begin{aligned} S(\text{disjoint}) &= T_1 \bigcup_{i=2}^{m} T_iE(F_i) \\ &= T_1 + T_2E(F_2) + T_3E(F_3) + T_4E(F_4) + T_5E(F_5) + T_6E(F_6) \\ &= q_1q_2 + q_6q_7(p_1 + q_1p_2) + q_4q_5q_6(p_1p_7 + q_1p_2p_7) \\ &+ q_2q_3q_4(p_1p_6 + p_1p_5q_6p_7) + q_1q_3q_5q_6p_2p_4p_7 \\ &+ q_2q_3q_5q_7p_1p_4p_6 \end{aligned} \tag{12}$$

5. Conclusions

For the example 3, (10) contains only 22 terms. The same example has been solved by Lin et al. [41] and the resulting expression contains 61 terms.

The only difficulty with the proposed methods seems to be in finding F_i's corresponding to T_i if i is large. As inspection of the terms occurring in $E(F_i)$ in Table 1 reveals that for the largest path of size $(k - 1)$, $E(F_i)$ is simply the intersection of complements of remaining $(b - (k - 1))$ branches.

3.9 Symbolic Reliability Evaluation Using Logical Signal Relations

1. Introduction

A method for finding the symbolic reliability expression of a general network is presented. It is based on the concept of logical signal relations. The algorithm applies to networks having unreliable nodes and/or branches.

An important advantage of the method is that it does not require a prior knowledge of any path or any cutset of the network. Such a knowledge is a prerequisite in most other methods of reliability analysis [55, 67].

2. Formulation of problem

Notation and definitions employed here are as follows:

k	total number of nodes
n_j	node j ($j = 1$ is source)
$'$	Boolean negation
	Boolean multiplication is shown by juxtaposition
$X_i (X_i')$	logical success (failure) of branch x_i
N_j	logical success of node n_j
$\overline{n_i n_j}$	fusion of nodes n_i and n_j
p_i	probability of success (reliability) of branch x_i
q_i	$1 - p_i$
$S(n_j)$	logical presence of the signal at node n_j
S	system success function; $S(n_k)$ for $S(n_1) = 1$
M	number of multiplications
A	number of additions
T_A, T_M	computation time for one addition or one multiplication

T_c total computation time for the numerical value of system reliability

R terminal-pair reliability

Δe absolute error

ΔR absolute error in system reliability R

Source (sink): The branches go from (to) the source (sink) node in the reliability block diagram, and no branch goes to (from) it.

Terminal Numbering Convention (TNC): To represent nodes (branches) in the reliability logic diagram, we will use general TNC [65].

Fusion of Nodes: A pair of nodes n_i and n_j are fused (merged) if the two nodes are replaced by a single new node such that all branches that were incident on either n_i or n_j or on both are incident on the new node.

The logical signal relations for common subnetworks are given in Table 3.9.1. Each relation is expressed so that its terms are always mutually disjoint.

In Table 3.9.1, logical relations have been expressed assuming nodes to be perfect. If the nodes are not perfect, these relations can be used in their modified form [7].

3. Computational procedure

The algorithm steps are as follows:

1. a. write the logical signal relation for the sink node.

   b. successively proceed towards for the sink node using relations in Table 1. Repeat b until the source node is reached.

   Substitute

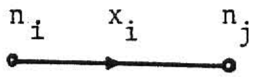
$$S(n_1) = S(\overline{n_1 \ \ldots \ }) = 1 \tag{1}$$

   where, $(\overline{n_1 \ldots})$; the fusion of any number of nodes, one of which is source node n_1.

TABLE 3.9.1.

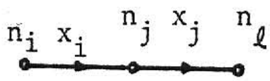
Subnetwork	Logical signal relations

1.



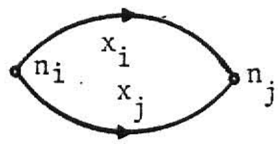
$$S(n_j) = X_i S(n_i)$$

2.



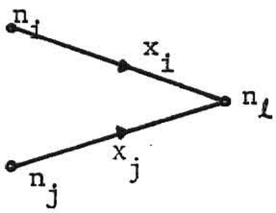
$$S(n_\ell) = X_j S(n_j) = X_i X_j \; S(n_i)$$

3.



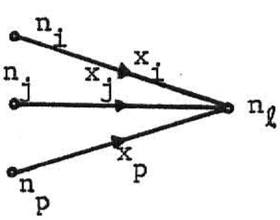
$$S(n_j) = X_i S(n_i) \;\mathrm{U}\; X_i' X_j S(n_i)$$
$$= (X_i \;\mathrm{U}\; X_i' X_j) \; S(n_i)$$

4.



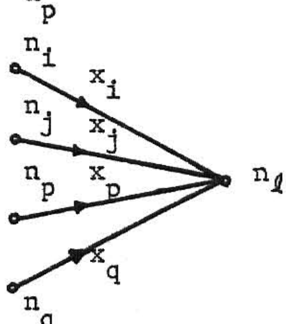
$$S(n_\ell) = X_i X_j' S(n_i) \;\mathrm{U}\; X_j X_i' S(n_j) \;\mathrm{U}$$
$$X_i X_j S(\overline{n_i n_j})$$

5.



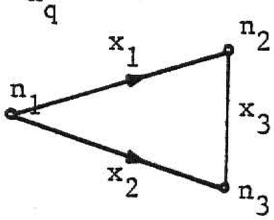
$$S(n_\ell) = X_i X_j' S(n_i) \mathrm{U} X_i X_j' S(\overline{n_i n_j})$$
$$\mathrm{U} X_i' X_j' X_p S(n_p)$$

6.



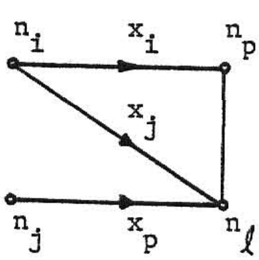
$$S(n_\ell) = X_i X_j' S(n_i) \;\mathrm{U}\; X_j X_i' S(n_j) \;\mathrm{U}$$
$$X_i X_j S(\overline{n_i n_j}) \;\mathrm{U}\; X_i' X_j' X_p S(n_p)$$
$$\mathrm{U}\; X_i' X_j' X_p' X_q S(n_q)$$

7.



$$S(n_2) = X_1 S(n_1) \;\mathrm{U}\; X_1' X_2 X_3 S(n_1)$$
$$S(n_3) = X_2 S(n_1) \;\mathrm{U}\; X_2' X_1 X_3 S(n_1)$$
$$S(\overline{n_2 n_3}) = X_1 S(n_1) \;\mathrm{U}\; X_1' X_2 S(n_1)$$

8.



$$S(\overline{n_p n_\ell}) = (X_i X_j' \mathrm{U}\; X_j X_p') S(n_i) \;\mathrm{U}$$
$$X_p X_i' S(n_j) \;\mathrm{U}\; X_i X_j X_p S(\overline{n_i n_j})$$

2. In the expression thus obtained for $S(n_k)$ in step 1, replace the logical variables by the corresponding probability variables to obtain the required terminal pair reliability expression. This is possible because the terms in $S(n_k)$ are mutually disjoint. If the actual sink node is not perfect, a perfect logical sink node must be added to the diagram.

4. EXAMPLE The network is shown in Fig. 3.9.1.

All nodes in this network are perfect.

The basic equation is

$$S(n_5) = X_7X_6' \; S(n_4) \cup X_6X_7' \; S(n_3) \cup X_6X_7 \; S(\overline{n_3n_4}) \tag{2}$$

Using Table 3.9.1 (#4, #5, #7) gives

$$\begin{aligned} X_7X_6' \; S(n_4) = {} & X_7X_6'[X_1X_4X_5' \cup X_1'X_2X_3X_4X_5' \cup X_2X_4'X_5 \\ & \cup X_1X_2'X_3X_4'X_5 \cup X_1X_4X_5 \cup X_1'X_2X_4X_5] \end{aligned} \tag{3}$$

$$X_6X_7' \; S(n_3) = X_6X_7'[X_2 \cup X_1X_2'X_3 \cup X_1X_2'X_3'X_4X_5] \tag{4}$$

Using relations #7, #8 of Table 3.9.1 gives

$$X_6X_7 \; S(\overline{n_3n_4}) = X_6X_7[X_1X_3'X_4 \cup X_1X_2'X_3 \cup X_1'X_2X_3'X_4 \cup X_2X_4' \cup X_2X_3X_4] \tag{5}$$

Substitute (3) - (5) in (2) and apply step 2 of the algorithm; the symbolic reliability expression is

$$\begin{aligned} R = {} & p_1p_4q_5q_6p_7 + q_1p_2p_3p_4q_5q_6p_7 + p_2q_4p_5q_6p_7 \\ & + p_1q_2p_3q_4p_5q_6p_7 + p_1p_4p_5q_6p_7 + q_1p_2p_4p_5q_6p_7 \\ & + p_2p_6q_7 + p_1q_2p_3p_6q_7 + p_1q_2q_3p_4p_5p_6q_7 \end{aligned}$$

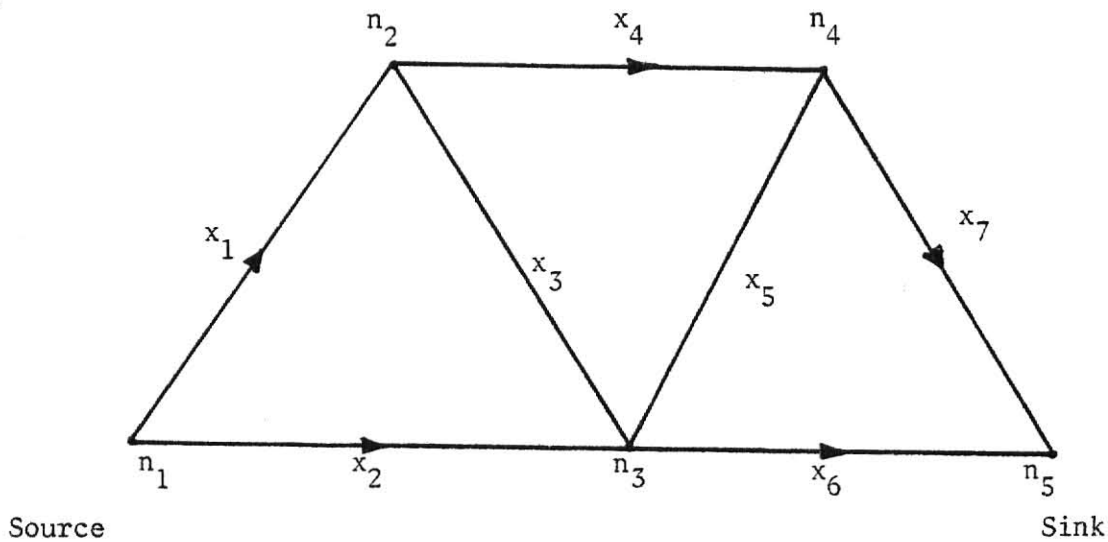


Fig. 3.9.1. Example of a signal network.

$$+ p_1q_3p_4p_6p_7 + p_1q_2p_3p_6p_7 + q_1p_2q_3p_4p_6p_7$$

$$+ p_2q_4p_6p_7 + p_2p_3p_4p_6p_7. \tag{6}$$

The minimal paths in the network are obtained by writing a Boolean expression (T) consisting only of uncomplemented variables from each of the terms in $S(n_k)$ and simplifying it by using the Boolean relation $Y \cup YZ \equiv Y$, where Y and Z are Boolean functions. In this example,

$$\begin{aligned} T = {} & X_1X_4X_7 \cup X_2X_3X_4X_7 \cup X_2X_5X_7 \cup X_1X_3X_5X_7 \\ & \cup X_1X_4X_5X_7 \cup X_2X_4X_5X_7 \cup X_2X_6 \\ & \cup X_2X_3X_6 \cup X_1X_3X_6 \cup X_1X_4X_5X_6 \\ & \cup X_1X_4X_6X_7 \cup X_1X_3X_6X_7 \\ & \cup X_2X_4X_6X_7 \cup X_2X_3X_4X_6X_7 \\ & \cup X_1X_6X_7 \\ = {} & X_1X_4X_7 \cup X_2X_3X_4X_7 \cup X_1X_3X_5X_7 \\ & \cup X_2X_5X_7 \cup X_2X_6 \cup X_1X_3X_6 \\ & \cup X_1X_4X_5X_6. \end{aligned} \tag{7}$$

Hence, from (7) the minimal paths are: $x_1x_4x_7$, $x_2x_3x_4x_7$, $x_1x_3x_5x_7$, $x_2x_5x_7$, x_2x_6, $x_1x_3x_6$, $x_1x_4x_5x_6$.

5. Conclusions

To compare the computer time required [3] for the evaluation of numerical value of reliability from (6), we observe that $M = 61$ and $A = 13$. Therefore, $T_C = 623\ T_A$.

For the purpose of error analysis it is assumed that all component reliability values have an absolute error of Δe. For simplicity, all reliability values are assumed equal to p, and Δe to be very small.

It is known [3]:

i) The absolute error in the sum of certain terms is equal to the sum of the absolute errors in these terms.

ii) The relative error in the multiplication of certain terms is equal to the sum of the relative errors in these terms. Absolute error, if desired, can be found by multiplying the relative error with the product.

Using points i & ii, error in system reliability ΔR, derived from (6) is

$$\begin{aligned}\Delta R = \Delta e[(5/p)p^3 &+ 3(4/p + 1/q)p^4 q + 2(4/p + 2/q)p^4 q^2 \\ &+ 3(4/p + 3/q)p^4 q^3 + (3/p + 1/q)p^3 q \\ &+ 3(3/p + 2/q)p^3 q^2 + (2/p + 2/q)p^2 q^2] \\ &\approx 9\Delta e;\ \text{if } p \approx 1. \end{aligned} \tag{8}$$

These results along with the corresponding results for existing methods are compared in Table 3.9.2. The proposed method is better than most existing methods. Although, it is less economical than method [6], it has the advantage of not requiring all paths of the network.

Table 3.9.2. Computational time and absolute error comparison.

Method	M	A	T_C/T_A	$\Delta R/\Delta e$	Remarks
1. Exhaustive search method [18]	354	58	3598	14	Applicable only for very small systems.
2. Direct canonical expansion method [45]					
3. Probability calculus method [42]	116	28	1188	178	Method useful only if number of interconnecting branches is small.
4. Bayes'theorem method [14]					
5. Flow graph method [47]					
6. Algebraic methods [6, 55]	46 39	9 8	469 398	5 4	Requires prior knowledge of paths, method general.
7. Proposed method	61	13	623	9	No prior knowledge of paths (cutsets) required.

3.10 Exhaustive Search Method Applied to Small Complex System Reliability

1. Introduction

This is one of the most primitive but straight-forward technique of reliability evalution. It is sometimes called as either State Enumeration algorithm or Event Space method.

2. Statement of the problem and the Computational Procedure

The method consists of listing all possible states; then sorting out those states in which system is a success; and hence writing the reliability expression.

The algorithm for the state enumeration method can be briefly stated as follows:

1) Find the all possible combinations of the states of the units (operating or failed).

2) For each combination which connects input and output, calculate the product of the unreliabilities of the failed units and the reliabilities of the up units.

3) Sum up the products obtained in step 2. This gives the system reliability expression.

3. Example

If there are b elements in a network and each element has two states (operating or failed); then, in all, there are 2^b states of the whole system. For the bridge network of Fig. 3.10.1; there are 5 elements and hence 32 states.

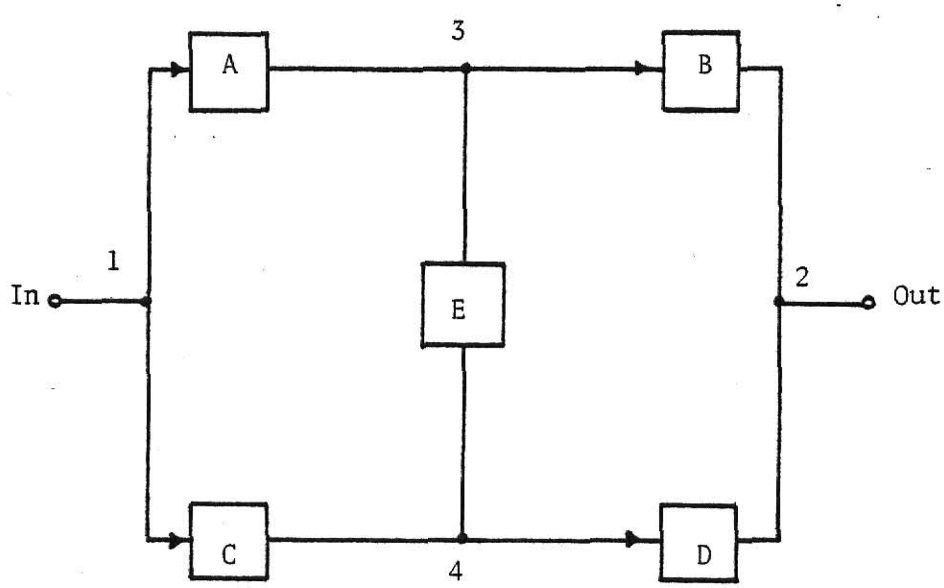


Fig. 3.10.1. A nonseries parallel system.

For this network, all combination of the states is shown in Table 3.10.1. From the table, we observe that there are 16 success states for the system and hence reliability expression is easily written as:

$$\begin{aligned} R = {} & q_a q_b p_c p_d q_e + q_a q_b p_c p_d p_e + q_a p_b p_c q_d p_e \\ & + q_a p_b p_c p_d p_e + q_a p_b p_c p_d q_e + p_a p_b q_c q_d q_e \\ & + p_a p_b q_c q_d p_e + p_a p_b q_c p_d p_e + p_a p_b q_c p_d q_e \\ & + p_a p_b p_c p_d q_e + p_a p_b p_c p_d p_e + p_a p_b p_c q_d p_e \\ & + p_a p_b p_c q_d q_e + p_a q_b p_c p_d p_e + p_a q_b p_c p_d q_e \\ & + p_a q_b q_c p_d p_e . \end{aligned} \tag{1}$$

4. Conclusions

The method has a serious drawback in the fact that the number of possible states of a system rises enormously with an increase in the number of branches. For a network with just 20 elements, number of all possible states is over one million.

An almost similar method has been proposed by Brown [18] which also gives an expression similar to (1).

Table 3.10.1

The Complete Listing of States

State number	Binary Number A	B	C	D	E	S	"Giving a path?
1	0	0	0	0	0	0	No
2	0	0	0	0	1	0	No
3	0	0	0	1	1	0	No
4	0	0	0	1	0	0	No
5	0	0	1	1	0	1	Yes
6	0	0	1	1	1	1	Yes
7	0	0	1	0	1	0	No
8	0	0	1	0	0	0	No
9	0	1	1	0	0	0	No
10	0	1	1	0	1	1	Yes
11	0	1	1	1	1	1	Yes
12	0	1	1	1	0	1	Yes
13	0	1	0	1	0	0	No
14	0	1	0	1	1	0	No
15	0	1	0	0	1	0	No
16	0	1	0	0	0	0	No
17	1	1	0	0	0	1	Yes
18	1	1	0	0	1	1	Yes
19	1	1	0	1	1	1	Yes
20	1	1	0	1	0	1	Yes
21	1	1	1	1	0	1	Yes
22	1	1	1	1	1	1	Yes
23	1	1	1	0	1	1	Yes
24	1	1	1	0	0	1	Yes
25	1	0	1	0	0	0	No
26	1	0	1	0	1	0	No
27	1	0	1	1	1	1	Yes
28	1	0	1	1	0	1	Yes
29	1	0	0	1	0	0	No
30	1	0	0	1	1	1	Yes
31	1	0	0	0	1	0	No
32	1	0	0	0	0	0	No

3.11 Direct Canonical Expansion Applied To Small Complex System Reliability

1. Introduction

Exhaustive search method consists of listing all possible states and then sorting out those states in which system is a success. To avoid enumerating the all possible combinations of the states, we write an expression for system success in terms of paths and then expand it into its canonical form [45].

2. Statement of the problem and the computational procedure

To evaluate system reliability, determination of all m paths is necessary. In a simple network, this may be possible by inspection; but in a general network some systematic method has to be used. Such methods available are Powers of connection matrix [37, 38, 48], State removal algorithm [2], and Graph theory method [59].

With all m paths obtained by above mentioned methods, we expand them into canonical form.

Special forms of Boolean expressions such as Expanded sum of products and Expanded product of sums are of particular interest here.

In the expanded sum of products, each term contains every variable, either uncomplemented or complemented. To obtain the expanded sum of products from a sum of products, the missing variables are supplied in all possible combinations to each product. Actually, in so doing, following theorem is used [45].

$$X = XY + X\overline{Y} \qquad (1)$$

As an example, the sum of products

$$\bar{A}C\bar{D} + A\bar{B}D + A\bar{C} \qquad (2)$$

The first term, $\bar{A}C\bar{D}$, has one missing variable B which is supplied in both its uncomplemented and complemented form; $\bar{A}C\bar{D}$ thus expans into two terms:

$$\bar{A}C\bar{D} = \bar{A}BC\bar{D} + \bar{A}\bar{B}C\bar{D} \qquad (2a)$$

The term, $A\bar{B}D$ also expands into two terms:

$$A\bar{B}D = A\bar{B}CD + A\bar{B}\bar{C}D \qquad (2b)$$

The $A\bar{C}$ term has two missing variables B and D.

Two variables can occur in four possible combinations. Therefore, the term $A\bar{C}$ expands into four terms:

$$A\bar{C} = AB\bar{C}D + AB\bar{C}\bar{D} + A\bar{B}\bar{C}D + A\bar{B}\bar{C}\bar{D} \qquad (2c)$$

The $A\bar{B}\bar{C}D$ term has already been obtained by the expansion of the ABD term and is not repeated.

The expanded sum of products is therefore

$$\bar{A}\bar{B}C\bar{D} + \bar{A}BC\bar{D} + A\bar{B}\bar{C}D + A\bar{B}CD + A\bar{B}\bar{C}\bar{D} + AB\bar{C}\bar{D} + AB\bar{C}D \qquad (2d)$$

Note, in this case, that the expanded sum of products contains seven of the sixteen possible combinations of the four variables. Although the "+" stands for the "inclusive or", the nature of an expanded sum of products is such that all terms are mutually exclusive.

3. Example

For the bridge network problem, the event of system success S can be written as:

$$S = AB \cup CD \cup AED \cup BEC. \tag{3}$$

Or,

$$\begin{aligned} S = {} & ABCDE + ABCD\bar{E} + ABC\bar{D}E + ABC\bar{D}\bar{E} \\ & + AB\bar{C}DE + AB\bar{C}D\bar{E} + AB\bar{C}\bar{D}E + AB\bar{C}\bar{D}\bar{E} \\ & + A\bar{B}CDE + A\bar{B}CD\bar{E} + \bar{A}BCDE + \bar{A}BCD\bar{E} \\ & + \bar{A}\bar{B}CDE + \bar{A}\bar{B}CDE + A\bar{B}\bar{C}DE + \bar{A}BC\bar{D}E \end{aligned} \tag{4}$$

In the canonical expansion, each term corresponds to a particular success state of the system and reliability expression (5) can be directly written from (4) by replacing X by p_x and $\bar{X}$ by q_x respectively.

$$\begin{aligned} R = {} & q_a q_b p_c p_d q_e + q_a q_b p_c p_d p_e + q_a p_b p_c q_d p_e \\ & + q_a p_b p_c p_d p_e + q_a p_b p_c p_d q_e + p_a p_b q_c q_d q_e \\ & + p_a p_b q_c q_d p_e + p_a p_b q_c p_d p_e + p_a p_b q_c p_d q_e \\ & + p_a p_b p_c p_d q_e + p_a p_b p_c p_d p_e + p_a p_b p_c q_d p_e \\ & + p_a p_b p_c q_d q_e + p_a q_b p_c p_d p_e + p_a q_b p_c p_d q_e \\ & + p_a q_b q_c p_d p_e . \end{aligned} \tag{5}$$

4. Conclusions

In this method, the resulting reliability expression is quite lengthy and requires much computational work for numerical evaluation of reliability. Even in such a simple example, the expression (5) requires 64 multiplications.

3.12 Probability Map Method Applied To Small Complex System Reliability.

1. Introduction

A map method is easy to use because the expression to be simplified is automatically expanded as it is entered on the map, and the prime implicants can be identified by the visual recognition of certain basic patterns. However, some practice is required before the user can feel confident in the use of maps, particularly when the number of variables becomes large.

2. Statement of the problem

A map method is presented here for combining component - part reliabilities, or, in general, for combining probabilities. The type of map used herein is similar in form to the truth maps (Veitch diagrams, Karnaugh maps, etc.) used so extensively in combinatorial studies of Boolean algebra.
The particular map chosen permits a consistent two-dimensional representation for any number of variables.
A map for n variables contains 2^n squares, there being a square on the map for every possible input combination. A 1 is placed in each square representing a combination for which an output is desires; a 0 is placed in each square representing a combination for which no output is desired. Often, to reduce the writing, the 0's are omitted, and a blank square is understood to represent a no-output combination.
It is assumed that the reader is familiar with truth maps and with probability (reliability) combinations.

3. Examples

Example 1 The block diagram for a three-part series case is shown in Fig. 3.12.1. In general, A, B, and C represent three independent events (such as the proper operations of three independent components). In particular,

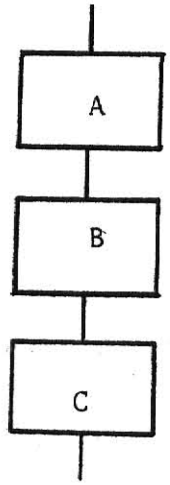


Fig. 3.12.1. A three-part series case.

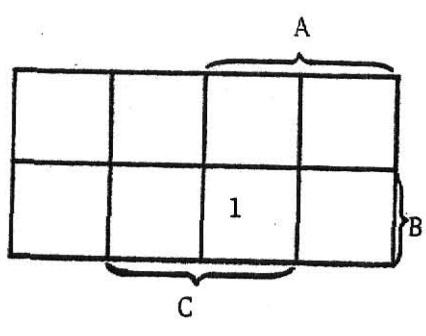


Fig. 3.12.2. Truth map for a three-part series case.

the series connection indicates that all three events must occur for the event of system success, S (such as the proper operation of a three-part system), to occur.

The Boolean algebra expression for the occurrence of the over-all event is therefore

$$S = ABC. \tag{1}$$

This function is plotted in a truth map in Fig. 3.12.2. Here it is clear that the over-all event requires the intersection of A, and B, and C for its occurrence.

In general, each different path through a probability diagram (such as Fig. 3.12.1) is represented by an intersection on the truth map.

Now if the peripheral column and row upper-case letters of the Boolean algebra truth map are changed to lower-case letters, we have an ordinary algebra probability map, as illustrated in Fig. 3.12.3. Here the letters on the edges represent the probabilities of occurrences of independent events.

Each cell (term) in the map represents a product of these independent probabilities, and each cell is mutually exclusive of all other cells. It is then evident that the probability of the event of system success S is simply

$$R_s = P_r \{S\} = abc \tag{2}$$

That is, we compose the map by filling in (with ones) all cells that represent complete paths through the reliability block diagram.

The over-all reliability is then the sum of all the (mutually exclusive) cells containing ones.

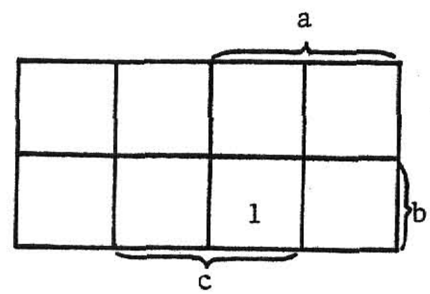


Fig. 3.12.3. Probability map for a three-part series case.

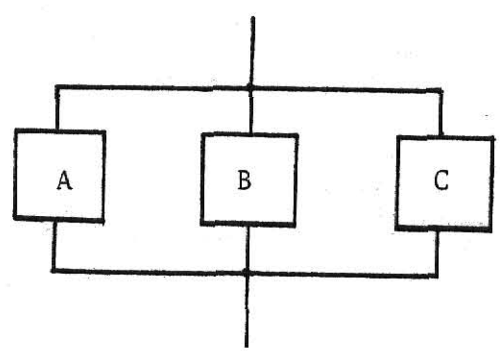


Fig. 3.12.4. A three-part parallel case.

Example 2. A three-part parallel case is pictured in Fig. 3.12.4. Here, the event of system success S will occur if any one (or more) of the independent events A, B, or C occur. Thus, there are three different complete paths through the system.

The filling in of path A is illustrated in Fig. 3.12.5 (a) - everything "under" A. In Fig. 3.12.5 (b), everything "under" B is filled with ones to represent the path through B in Fig. 3.12.4. Finally, in Fig. 3.12.5 (c) everything is filled in "under" C.

The complete map, converted to probability form (lower-case letters) is given in Fig. 3.12.6.

Taking every filled cell from left to right along the top, then along the bottom, the over-all probability function is

$$R_s = \bar{a}\bar{b}c + a\bar{b}c + a\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}bc + abc + ab\bar{c}. \tag{3}$$

However, in this case it would be simpler to "cover" the single empty cell instead of the seven filled cells. Thus,

$$\bar{R} = \bar{a}\bar{b}\bar{c} \tag{3a}$$

from which

$$R = 1 - \bar{a}\bar{b}\bar{c} \tag{3b}$$

or

$$R = 1 - (1 - a)\ (1 - b)\ (1 - c) \tag{3c}$$

Example 3. A simple series-parallel block diagram is shown in Fig. 3.12.7. The different paths through this diagram are AC, AD, BC, and BD. The filling in of these paths in the truth map is pictured, step-by-step, in Fig. 3.12.8, and the resultant probability map is given in Fig. 3.12.9.

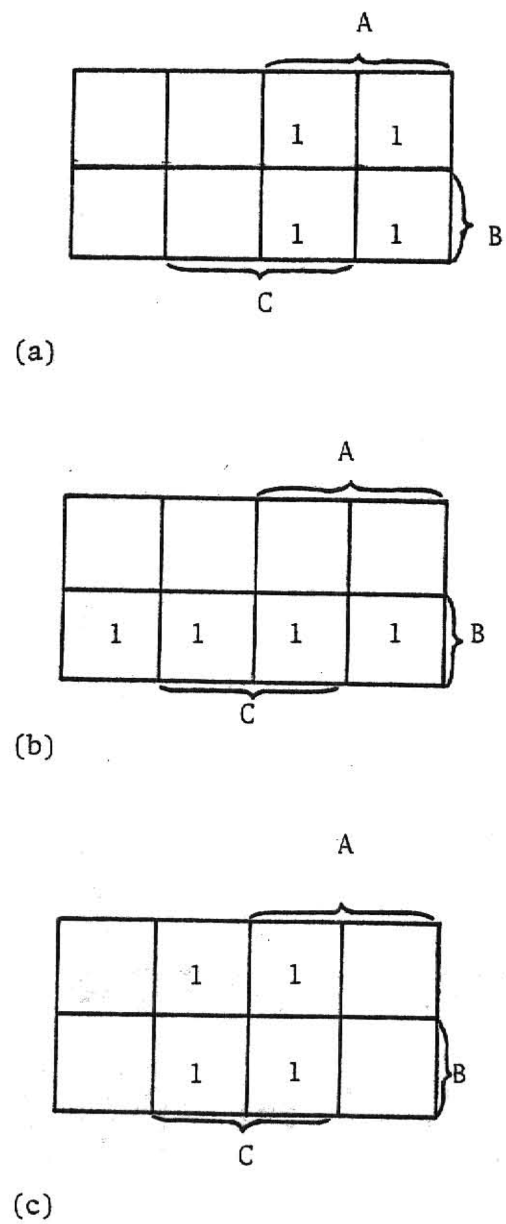


Fig. 3.12.5. Filling in paths in the truth map for a three-part parallel case. (a) path A, (b) path B, (c) path C.

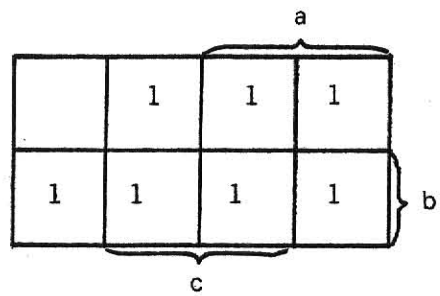


Fig. 3.12.6. Probability map for a three-part parallel system.

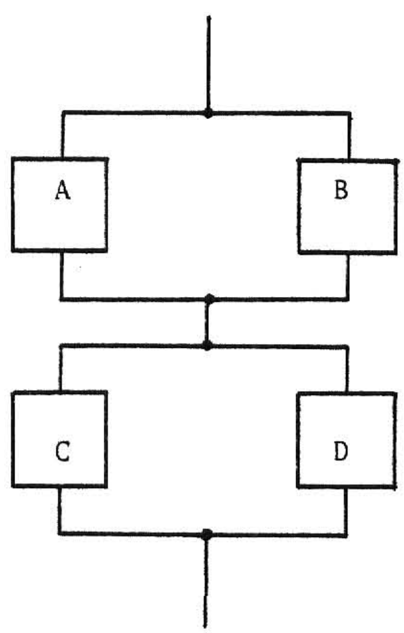


Fig. 3.12.7. A four-part series-parallel case.

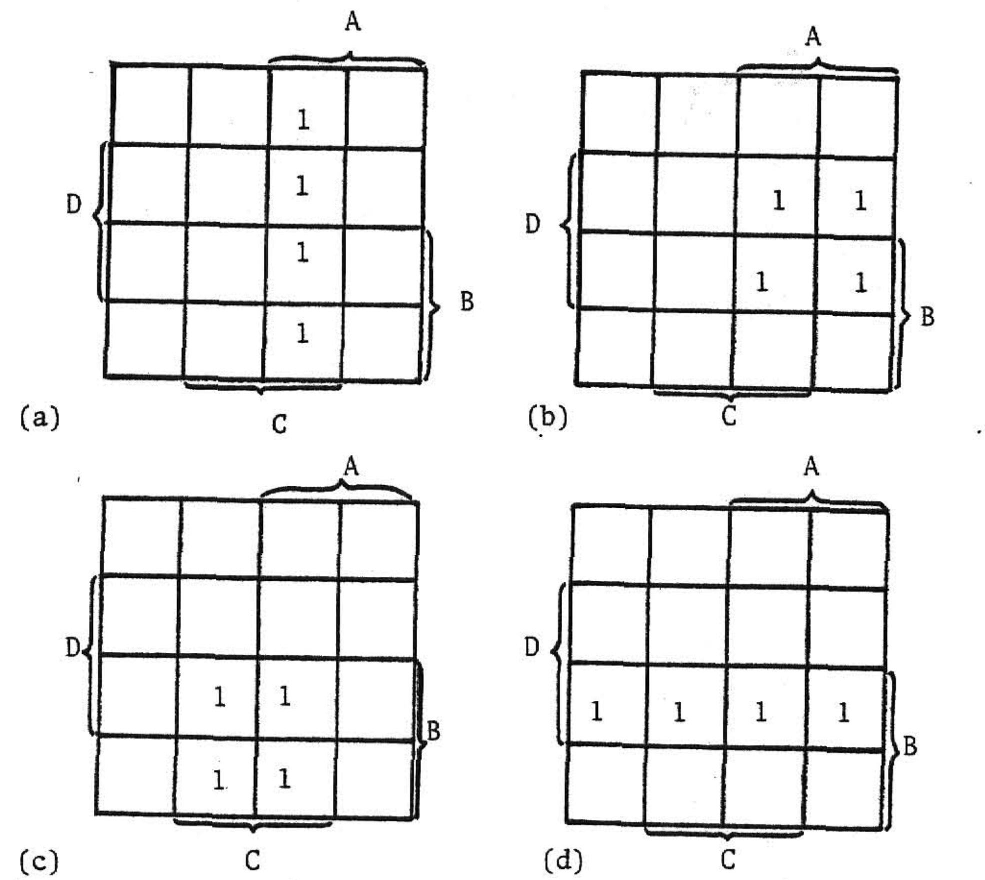


Fig. 3.12.8. Filling in the paths in the truth map for a particular four-part series-parallel case. (a) path AC, (b) path AD, (c) BC, (d) BD.

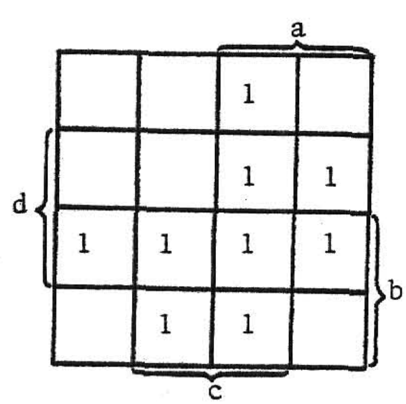


Fig. 3.12.9. Probability map for a particular four part series-parallel case.

The over-all reliability Rs can be written from either the nine filled cells or from the seven empty cells. However, properly composed maps permit the choice of a variety of useful groupings that lead to simplified or reduced functional forms.

Taking the given series-parallel case, some compatible groupings are illustrated in Fig. 3.12.10.

From Fig. 3.12.10 (a), left-to-right by columns,

$$Rs = \bar{a}b\bar{c}d + \bar{a}bc + ac + a\bar{c}d \qquad (4)$$

and from Fig. 3.12.10 (b), top-to-bottom by rows.

$$Rs = a\bar{b}c\bar{d} + a\bar{b}d + bd + bc\bar{d}. \qquad (4a)$$

These expressions, of course, are subject to all the rules of ordinary algebra such as factoring and substitution. Thus, (4a) can be factored to give

$$\begin{aligned} Rs &= a\bar{b}(c\bar{d} + d) + b(d + c\bar{d}) \qquad (4b) \\ &= (a\bar{b} + b)(c\bar{d} + d) \end{aligned}$$

and substitutions can be made to give

$$Rs = a(1 - b)[c(1 - d) + d] + b[d + c(1 - d)] \qquad (4c)$$

and so forth.

In Fig. 3.12.10 the top single cell gives $\underline{a\bar{b}c\bar{d}}$, the right-hand block of four gives $\underline{ad}$, the left single cell gives $\underline{\bar{a}b\bar{c}d}$, and the lower block of four gives $\underline{bc}$. However, the cell $\underline{abcd}$ was used twice in forming the two blocks of four. Therefore, its second use must be subtracted back out. From Fig. 3.12.10 (c),

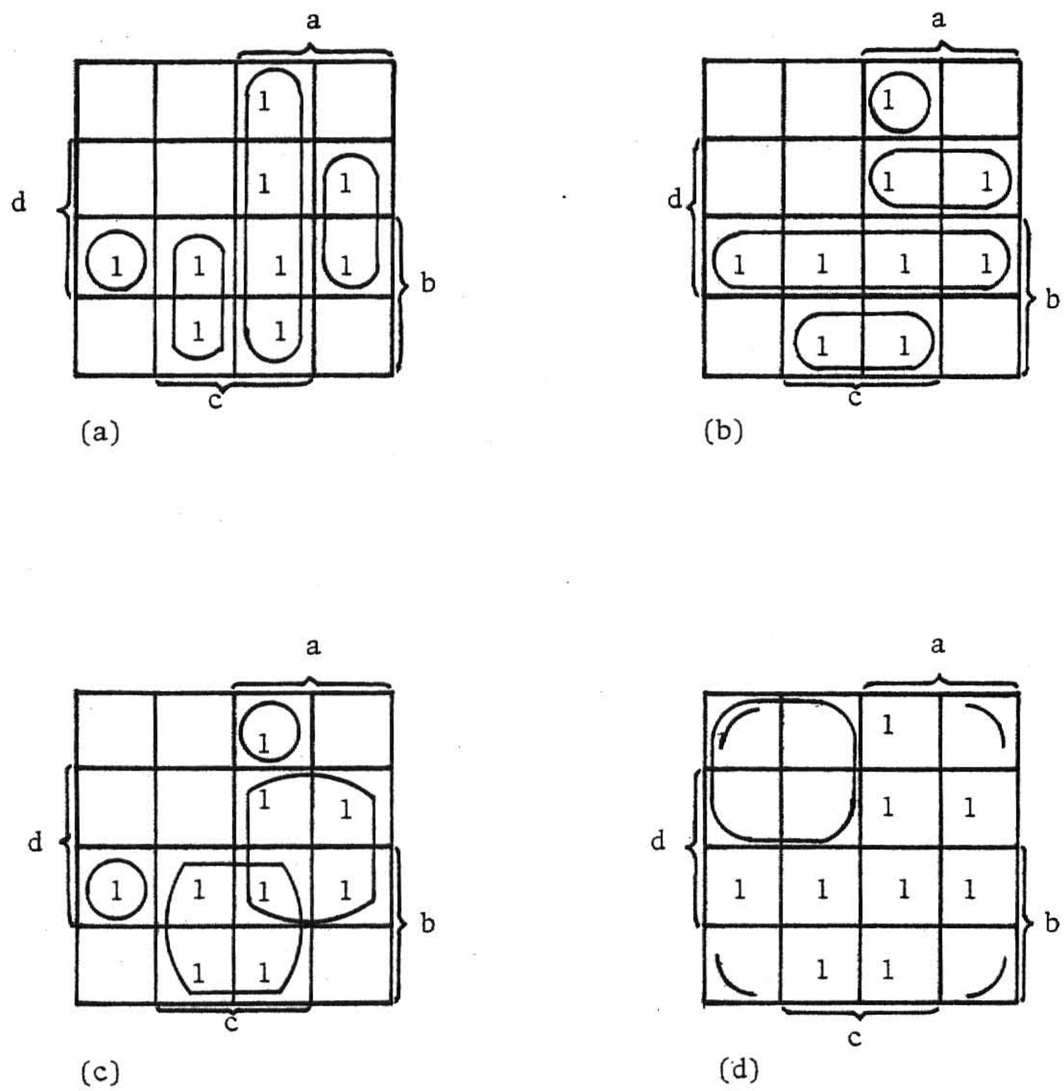


Fig. 3.12.10. Some examples of compatible groupings in the four-part series-parallel case.

then,

$$Rs = a\bar{b}c\bar{d} + ad + \bar{a}b\bar{c}d + bc - abcd. \qquad (4d)$$

In Fig. 3.12.10 (d) we have chosen to cover the empty cells. The upper-left block of four gives $\underline{\overline{ab}}$ and the four corners give $\underline{\overline{cd}}$. However, we used the upper-left corner cell $\underline{\overline{abcd}}$ twice, so we must subtract it once. Here,

$$\bar{R} = \overline{ab} + \overline{cd} - \overline{abcd} \qquad (4e)$$

OR

$$R = 1 - \overline{ab} - \overline{cd} + \overline{abcd}. \qquad (4f)$$

Example 4.

A nonseries-parallel diagram is shown in Fig. 3.12.11. Here, component A feeds B, only, and, in parallel, C feeds D, only. However, E can feed either B or D (can alternate). Any continuous through path allows the system to operate. Thus, the paths are AB, CD, EB, and ED, and the corresponding intersections are filled with ones in Fig. 3.12.12.

For the arbitrary selection of cell groupings in Fig. 3.12.12, the eight cell group gives $\underline{ab}$, the four-cell group gives $\underline{\bar{a}cd}$, the three two-cell groups give a $\underline{\overline{ac}de}$, $\underline{\bar{a}b\bar{d}e}$, and $\underline{a\bar{b}cd}$, and the single cell gives $\underline{a\overline{bc}de}$. Hence,

$$Rs = ab + \bar{a}cd + \overline{ac}de + \bar{a}b\bar{d}e + a\bar{b}cd + a\overline{bc}de. \qquad (5)$$

Example 5.

Another nonseries-parallel case, that of the bridge, is pictured in Fig. 3.12.13. Here, E represents a two-way bridging element, and the through paths are AB, AED, CD, and CEB. These intersections are filled with ones in Fig. 3.12.14.

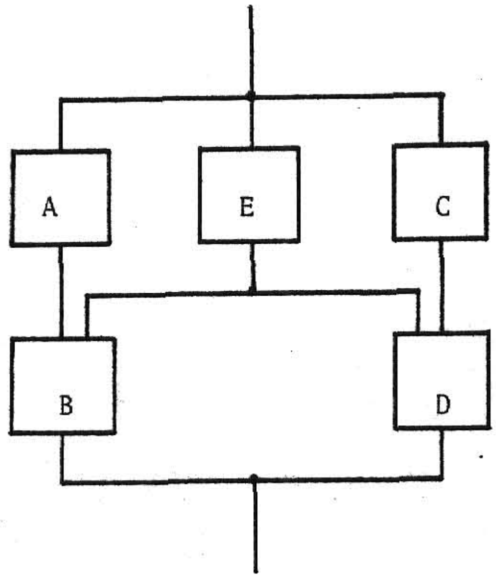


Fig. 3.12.11. An alternate case.

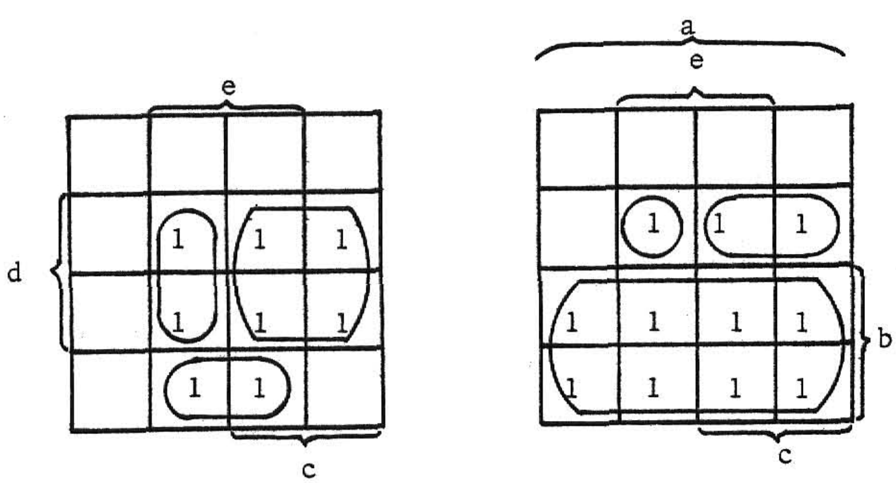


Fig. 3.12.12. Probability map for an alternate case, with an arbitrary selection of groupings.

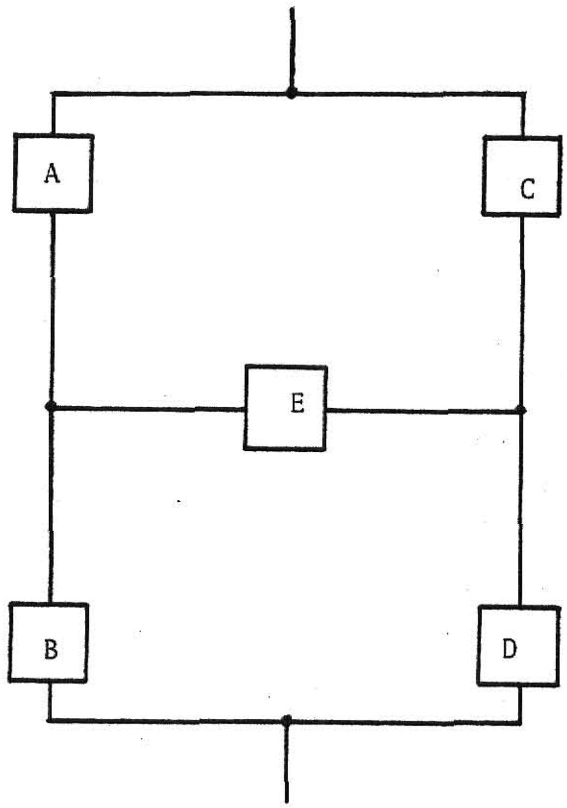


Fig. 3.12.13. A bridge case.

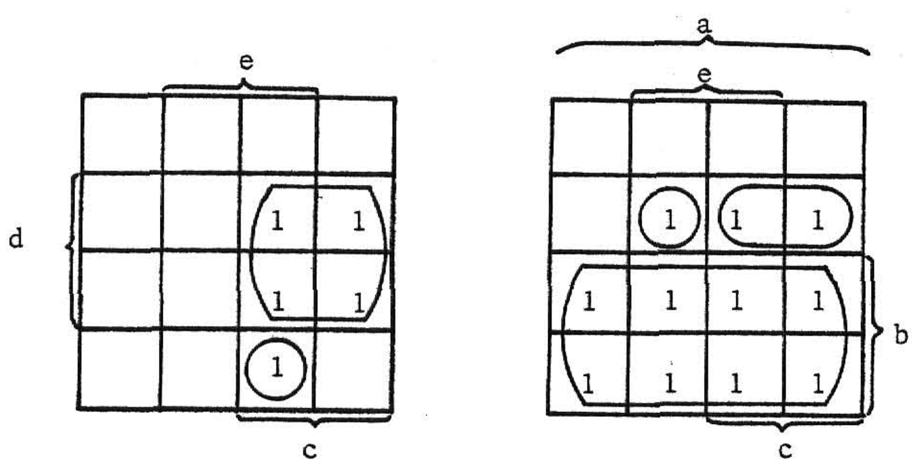


Fig. 3.12.14. Probability map for a bridge case, with an arbitrary selection of groupings.

In Fig. 14, the arbitrarily selected groupings yield

$$Rs = ab + \bar{a}cd + a\bar{b}cd + a\overline{bc}de + \bar{a}bc\bar{d}e. \qquad (6)$$

4. Conclusions

A particularly important property for the probability map method is that all its terms are mutually exclusive because of the nature of loops. The expression in (6) requires only 14 multiplications for numerical evaluation. A reduction in the number of computations will increase the accuracy and computer calculation time will also be reduced.

But this method is a graphical method and hence it is not easy to use this technique on computer. Moreover, the method is convenient only if the number of variables is six or less than six; but become very tedious in case the number of variables is large.

Chapter 4. EVALUATION OF THE LARGE SYSTEM RELIABILITY

4.1 A Computer Program Applied to Approximating Large System Reliability--Success Paths and Cut Sets Approach

1. Introduction

In the last decade, several papers have been written on the subject of reliability approximations and bounds by using the concepts of success paths (or tie sets) and cut sets.

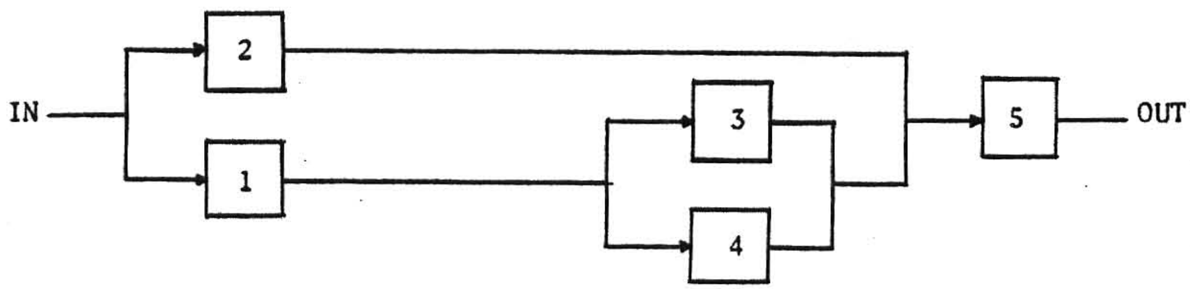
Further discussion of bounds and approximations is given by Messinger and Shooman [46]. Jensen and Bellmore [36] provide an algorithm for determining the reliability of a complex system in which the components or elements all have two terminals. Shooman [62] provides further mathematical background material and an entire chapter on combinatorial reliability.

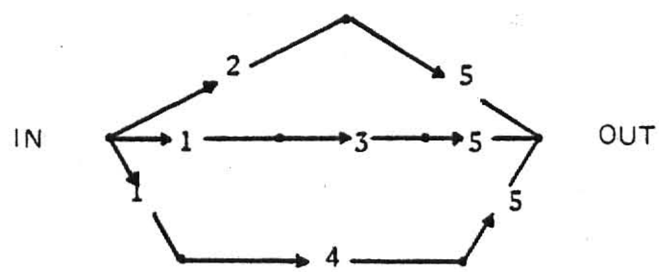
One of the difficulties associated with obtaining reliability estimates for complex systems is that of deriving a prediction equation which expresses all possible events of interest. One way to alleviate this difficulty is to obtain a sequence of prediction equations which provide increasingly closer bounds on the system reliability. A method for doing this and a computer program for performing the tedious computations are described here.

2. Formulation of the problem

The success probability of a system, typically called the system reliability, is defined as the probability of successful function of all of the elements in at least one tie set or as the probability that all cut sets are good. A tie set (success path) is a directed path from input to output as indicated in the simple system in Fig. 4.1.1(b). The tie sets are (2,5), (1,3,5), (1,4,5).

A cut set is a set of elements which literally cuts all success





(b)

Fig. 4.1.1. (a) Simple functional logic diagram. (b) Reliability graph corresponding to functional logic diagram

paths; that is, it severs the line of communication between input and output. One is usually interested in the minimal cut set, i.e., the smallest (minimal) set of elements such that the elimination of any one element would no longer make it a cut. In the preceding example, the minimal cut sets are (1,2), (2,3,4), (5).

The system failure probability (system unreliability) is the probability that all tie sets have a failure, that is,for each set at least one element fails, or is the probability that at least one cut occurs, that is, all the elements of the cut set fail.

Hereafter, the cut set will mean the minimal cut set.

Let T_i, i = 1, ..., I, denote the tie sets, I in number.

Let C_j, j = 1, ...,J, denote the cut sets, J in number.

Then, the system reliability R can be expressed as follows:

$$\begin{aligned} R &\equiv \Pr\{T_1 + T_2 + \ldots + T_1\} \\ &= \Pr\ \{\text{at least one tie set is good}\}. \qquad (1) \end{aligned}$$

Expressed in terms of the cut sets

$$\begin{aligned} R &\equiv \Pr\ \{C_1.C_2\ldots C_J\} \\ &= \Pr\ \{\text{all cut sets are good, viz., contain at least} \\ &\qquad \text{one element of the set which is operative}\}. \qquad (2) \end{aligned}$$

Equivalently, the unreliability is expressed as

$$\begin{aligned} 1 - R &\equiv \{\bar{T}_1.\bar{T}_2\ldots\bar{T}_1\} \\ &= \Pr\ \{\text{all tie sets have a failure}\} \qquad (3) \end{aligned}$$

Or $$1 - R \equiv \Pr\{C_1 + C_2 + \ldots + C_J\}$$

$$= \Pr\{\text{at least one cut occurs}\} \tag{4}$$

where $\overline{T}_i$ and $\overline{C}_j$ are the complements of the events T_i and C_j, respectively. Thus $\overline{T}_i$ denotes failure of at least one item in the i^{th} tie set, and $\overline{C}_j$ denotes failure of all items of the j^{th} cut.

Bounds can be obtained by using the basic probabilistic inequalities in the following:

$$R \equiv \Pr\{T_1 + T_2 + \ldots + T_I\} \leq \sum \Pr\{T_i\} \tag{5}$$

$$R \equiv \Pr\{T_1 + T_2 + \ldots + T_I\} \geq \sum \Pr\{T_i\}$$

$$- \sum_{i<k} \Pr\{T_i \cdot T_k\}, \quad 1 \leq i, \ k \leq I. \tag{6}$$

Thus an upper bound R_{U1} and a lower bound R_{L1} to the reliability are

$$R_{U1} \equiv \sum \Pr\{T_i\} \tag{7}$$

$$R_{L1} \equiv \sum \Pr\{T_i\} - \sum_{i<k} \Pr\{T_i \cdot T_k\} \tag{8}$$

In the same manner, another upper bound is obtained

$$R_{U2} = \sum \Pr\{T_i\} - \sum_{i<k} \Pr\{T_i \cdot T_k\}$$

$$+ \sum_{i<k<l} \Pr\{T_i \cdot T_k \cdot T_l\}, \quad 1 \leq i, \quad k, l \leq l \tag{9}$$

The last two summations are over all possible combinations of the subscripts taken 2 at a time, 3 at a time. Similarly, the inequalities (5) and (6) can be applied to the cut-set form of the equation for unreliability (4) to obtain

$$1 - R \leq \sum \Pr\{\overline{C}_j\}$$

or $$R \geq 1 - \sum \Pr\{\overline{C}_j\} \equiv R_{L2} \tag{10}$$

and by using two terms

$$R \le 1 - \Sigma \Pr\{\overline{C}_j\} + \sum_{j<m} \Pr\{\overline{C}_j\overline{C}_m\} \equiv R_{U3},$$

$$1 \le j,\ m \le J. \tag{11}$$

As stated in [46], the bounds based on the cut sets are best in the high reliability region, and those based on the tie sets are best in the low reliability region.

Example 1

Consider the reliability graph given in Fig. 4.1.1(b). Assume statistical independence between items and let the probabilities of success for each of the items be

$$\Pr\{1\} \equiv 0.93,\ \Pr\{2\} \equiv 0.86,\ \Pr\{3\} \equiv 0.92,\ \Pr\{4\} \equiv 0.95,\ \Pr\{5\} \equiv 0.98$$

$$(\Pr\{\overline{1}\} \equiv 1 - \Pr\{1\} = 0.07,\ \Pr\{\overline{2}\} = 0.14,\ \text{etc})$$

The probabilities for the ties and cuts are as follows:

$$\Pr\{T_1\} = \Pr\{2\cdot 5\} = \Pr\{2\}\ \Pr\{5\} = 0.8428$$

$$\Pr\{T_2\} = \Pr\{1\cdot 3\cdot 5\} = 0.8385$$

$$\Pr\{T_3\} = \Pr\{1\cdot 4\cdot 5\} = 0.8658$$

and

$$\Pr\{C_1\} = \Pr\{1 + 2\} = 1 - \Pr\{\overline{1}\}\ \Pr\{\overline{2}\}$$

$$= 1 - 0.0098 = 0.9902$$

$$\Pr\{C_2\} = \Pr\{2 + 3 + 4\} = 1 - \Pr\{\overline{2}\}\ \Pr\{\overline{3}\}\ \Pr\{\overline{4}\}$$

$$= 0.99944$$

$$\Pr\{C_3\} = \Pr\{5\} = 1 - \Pr\{\overline{5}\} = 0.98$$

Upper and lower bounds for the reliability are obtained by using (7), (8), (9), (10), and (11), respectively,

$$R_{U1} = \Sigma\ Pr\{T_i\} = 0.8428 + 0.83849 + 0.86583$$
$$= 2.54712 > 1$$

not useful, as obviously 1 is an upper bound..

$$R_{L1} = 0.8428 + 0.8385 + 0.8658 - Pr\{1\cdot 2\cdot 3\cdot 5\}$$
$$- Pr\{1\cdot 2\cdot 4\cdot 5\} - Pr\{1\cdot 3\cdot 4\cdot 5\} = 0.28484$$

$$R_{U2} = 0.28484 + Pr\{1\cdot 2\cdot 3\cdot 4\cdot 5\} = 0.28484 + 0.68504$$
$$= 0.96988 = R$$

this result being equal to the system reliability.

$$R_{L2} = 1 - \sum Pr\{\overline{C}_j\}$$

$$= 1 - (0.07)(0.14) - (0.14)(0.08)(0.05) - 0.02 = 0.96964$$

$$R_{U3} = 1 - 0.03036 + 0.00024 = 0.96988$$

where 0.96988 is the system reliability. Hence the bounds R_{L2} and R_{U3} are the preferred bounds in the preceding example, and R_{U3} in this example saves no computation as it is the exact probability of system success. In more general problems in which there are J cut sets, the number of terms in the lower and upper bound computations R_{L2} and R_{U3} are J and $J(J + 1)/2$, respectively. This is compared to $2^J - 1$ terms obtained by expanding (4). In order to perform the computations, a reliability prediction program has been written to obtain the tie sets from a listing of the items in the system and their predecessors. Then it obtains the cut sets and the sequence of probability bounds using only the cut sets until a value of desired precision is obtained. This program is described in the following section.

3. Program Description (Refer to the program in Appendix A1)

The bounds for system reliability in the high-reliability region are obtained from calculations which are based on cut sets (best in high-reliability region). This program calculates upper and lower bounds using the probabilities of success of each item in the system. It is written in a FORTRAN dialect, and flow diagram is given in Fig. 4.1.2.

Input simplicity is one of the features of this program. The user need only supply the success probabilities and a list of predecessors for each element in the system. The list of predecessors is established by feeding to the computer a card for each element. Each card identifies the items in the system preceding that element in a directed flow diagram. Table 4.1.1 shows an example corresponding to the reliability logic flow diagram in Fig. 4.1.1(a).

The algorithom is not complex but is rather a series of simple steps. These steps in order are: read the list of elements, develop the tie sets, develop the cut sets, and calculate the bounds.

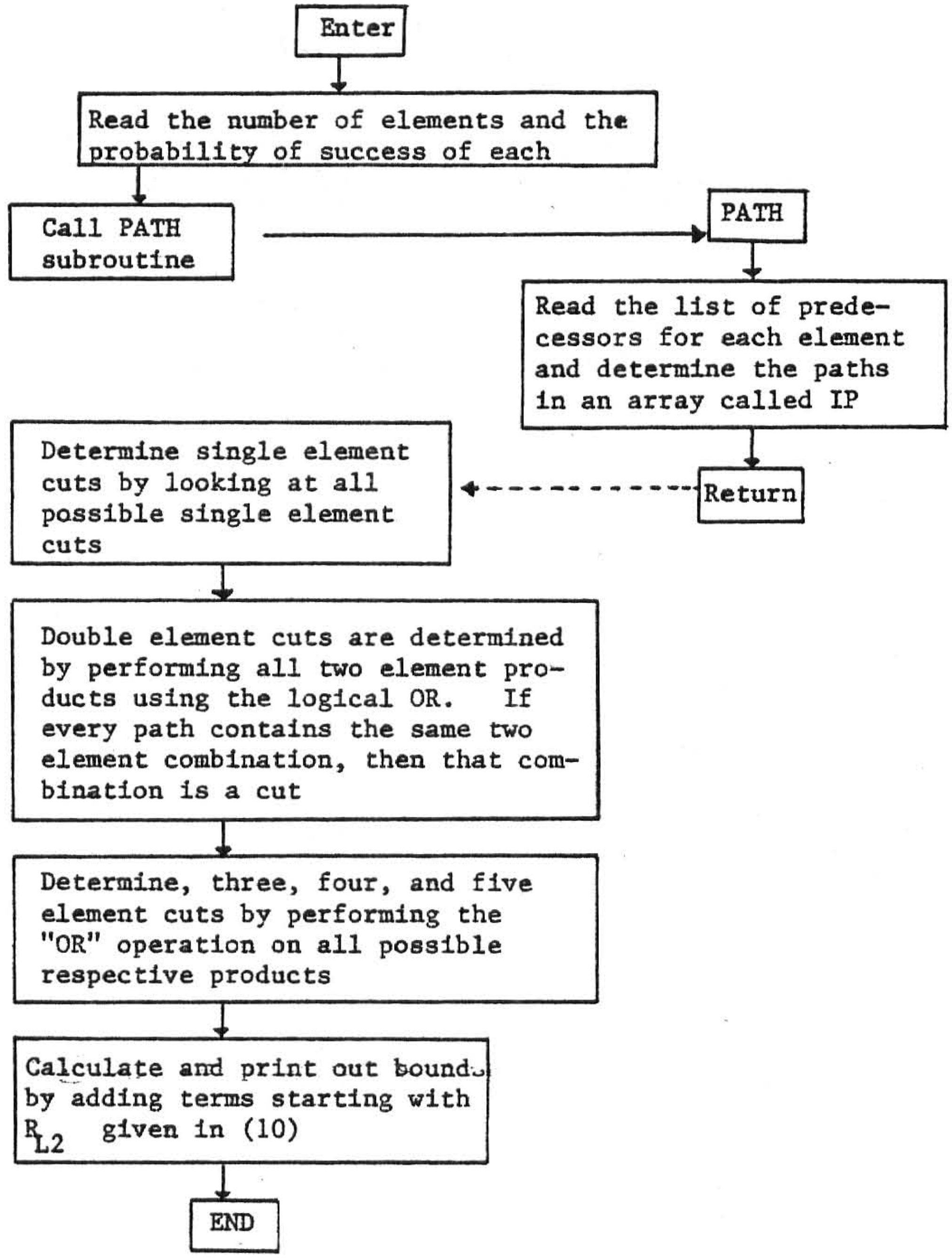
The success paths (tie sets) are developed by a subroutine called PATH. The arguments are number of items N in the system, number of success paths NP found by the subroutine, the array of the success paths IP found by the subroutine. The list of predecessors is read by the PATH subroutine. After being printed, the paths are converted to a Boolean array of zeros and ones, and the cut sets are developed by the procedure given in the following subsection.

(1) Generation of Cut Sets

A simple procedure using Boolean logic is used for obtaining a matrix identifying the minimal cuts of the system from the matrix containing the paths.

Let the path matrix be

$$P \equiv \begin{bmatrix} P_1 \\ P_2 \\ P_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix}$$



Comment: The paths are determined by element number in reverse order; therefore, the program corrects the order for output purposes and forms a Boolean matrix whereby the paths are the rows.

Fig. 4.1.2. Flow diagram for computer program: bounds for reliability [52].

Table 4.1.1 List of Predecessors for Example 1, Fig. 4.1.1(a).

Item	Predecessors	Card Code
1	IN	-1
2	IN	-1
3	1	1
4	1	1
5	2,3,4	2,3,4
OUT	5	25

Where the tie sets are P_1 = (1,3,5) (indicated by 1's in columns 2, 3, and 5 of the first row); P_2 = (2,5) and P_3 = (1,4,5).

Now consider the column vectors (1,0,1), (0,1,0), etc., of the path matrix P. For a single element to be a cut, it must be in each path, i.e., its column vector in P must be the unit vector (1,1,1). Note that element 5 is the only single element cut. In general, if P_c denotes the C^{th} column vector of an I-path matrix, and if the elements

$$P_{ci} = 1 \qquad \text{for all } i = 1,2,\ldots,I$$

then the corresponding element C is a single element cut. If P_{ci} = 0 for some i in each column vector, then there are no single-element cuts, and one must proceed to look for two-element cuts.

For two-element cuts consider for $c \neq d$

$$P_{ci} + P_{di}$$

where the + indicates the logic sum or union. If

$$P_{ci} + P_{di} = 1, \quad \text{for all } i = 1,2,\ldots,I$$

then elements c and d form a two-element cut. For example, $P_1 + P_2 = 1$ for all i = 1,2,...,I, and hence elements 1,2 form a cut.

This procedure continues until all possible cuts of order 1,2,...,n (n being the number of elements in the system) have been exhausted or until only unit vectors are obtained in the vector unions as described. At each stage all the nonminimal cuts are eliminated by using the following approach.

After a possible cut of order M has been identified, it is checked against all cuts of order M-1, M-2, ...,1 by using Boolean logic for intersection, i.e., the AND operation, for the multiplication of two vectors. If the cut of order M contains a cut of smaller order, the vector product would be equal to the order of the smaller cut. In this case the cut of order M would be eliminated because it is nonminimal.

(2) Output

The output is brief and easily read. Input probabilities, tie sets, and cut sets are printed. Since the calculation for bounds is done by adding terms to a series with each new term resulting in a new bound, either lower or upper, the bounds are given at each step with the appropriate last term shown. For small systems the exact system reliability might be calculated at the final step.

Computer results for Example 1

The example in Fig. 4.1.1 is used. The path matrix is given by the following:

$$\begin{array}{cc} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} & \text{Paths} \\ P = & \begin{bmatrix} 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 \end{bmatrix} & \begin{array}{l} 1,3,5 \\ 2,5 \\ 1,4,5 \end{array} \end{array}$$

and the cut matrix by

$$\begin{array}{cc} & \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} & \text{Cuts} \\ C = & \begin{bmatrix} 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix} & \begin{array}{l} 5 \\ 1,2 \\ 2,3,4 \end{array} \end{array}$$

The three cuts are thus (5), (1,2), and (2,3,4) as indicated by the 1's in the corresponding positions in rows 1, 2, and 3. The upper and lower bounds are obtained as indicated in the previous section. The program results from the computer printout are shown in Table 4.1.2.

4. Example

The system shown in Fig. 4.1.3 is a relatively complex series-parallel network. As can be seen, there are many possible success paths through the

Table 4.1.2. Bounds for System Reliability Example

Circuit Contains 5 Elements	
Element Numbers	Probability of Success
1	0.9300
2	0.8600
3	0.9200
4	0.9500
5	0.9800
Tie Sets or Success Paths (3)	Element Numbers
1	2 5
2	1 3 5
3	1 4 5
Cuts Sets (3)	Element Numbers
1	5
2	1 2
3	2 3 4
Lower bound is 0.96964E 0	Last term 0.30361E - 1
Upper bound is 0.96988E 0	Last term 0.24541E - 3
Lower bound is 0.95988E 0	Last term 0.78407E - 6
System reliability 0.96988E 0	

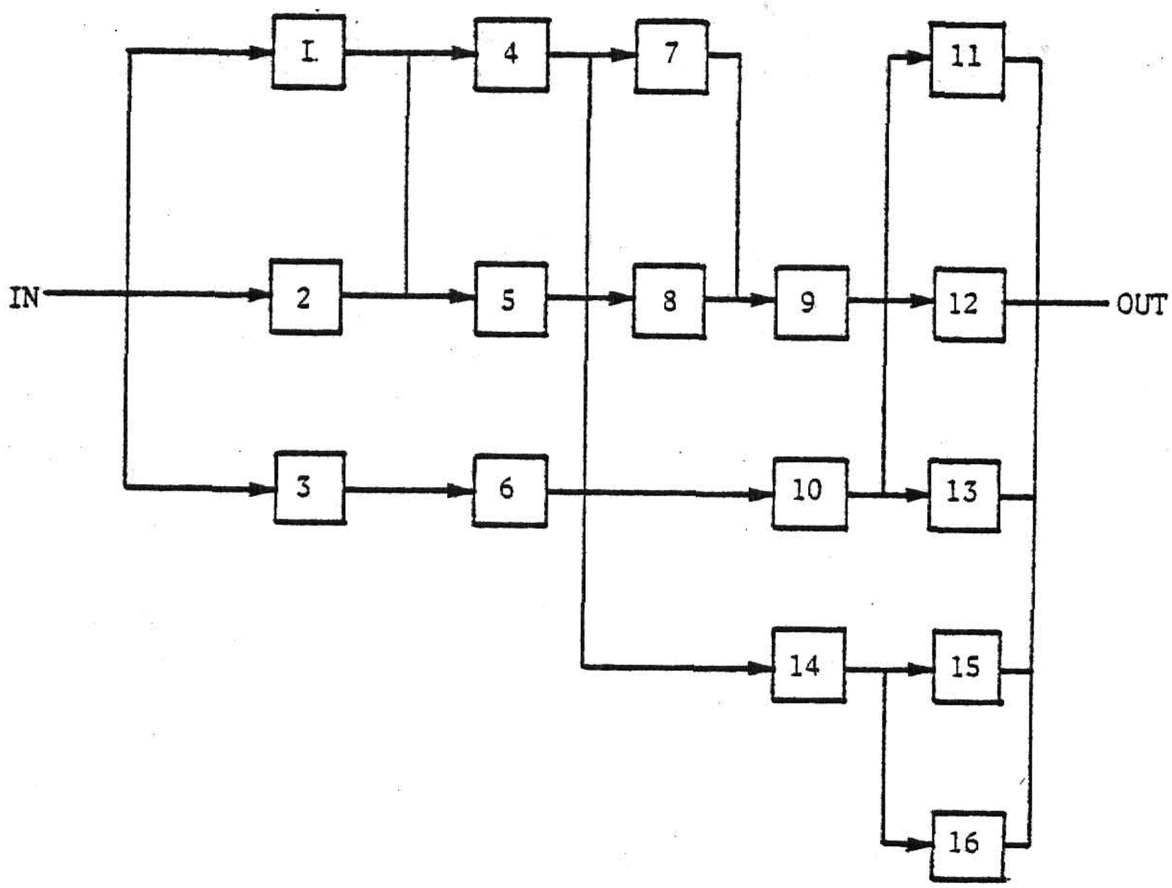


Fig. 4.1.3. System diagram for bounds program [52].

system and hand calculation of system reliability would be at best very tedious.

The reliability of each element is given in Table 4.1.3. As required by the program, element failures are assumed statistically independent. Some of the 55 tie sets and all of the 10 cut sets are shown in Table 4.1.4 The bounds program printout is given at the bottom of Table 4.1.4.

As can be seen from the last two lines of the printout, the program has bounded the system reliability. Since the upper and lower bounds have converged to the same value 0.97726, this value is the system reliability to five-place accuracy.

5. Conclusions

A computer program, which provides bounds for system reliability, is described. The algorithms are based on the concepts of success paths and cutsets. A listing of the elements in the system, their predecessors, and the probability of successful operation of each element are the inputs. The outputs are the success paths, the cutsets, and a series of upper and lower reliability bounds. Two examples are used to illustrate the algorithm and features of the computer program.

Table 4.1.3. Probabilities of Success for Each Element of Example 2.

Circuit Contains 16 Elements	
Element Numbers	Probability of Success
1	0.80
2	0.80
3	0.90
4	0.85
5	0.75
6	0.87
7	0.82
8	0.82
9	0.89
10	0.88
11	0.85
12	0.85
13	0.85
14	0.75
15	0.70
16	0.70

Table 4.1.4. Tie Sets and Cut Sets for Example 2.

Tie Sets of Success Paths (55)	Element Numbers				
1	1	4	7	9	11
2	1	4	7	9	12
3	1	4	7	9	13
4	1	4	14	15	
5	1	4	14	16	
6	1	4	10	11	
7	1	4	10	12	
.	.	.	.	.	.
.	.	.	.	.	.
.	.	.	.	.	.
55	2	5	8	9	13

Cut Sets (10)	Element Numbers				
1	1	2	3		
2	1	2	6		
3	3	4	5		
4	4	5	6		
5	9	10	14		
6	7	8	10	14	
7	9	10	15	16	
8	11	12	13	14	
9	7	8	10	15	16
10	11	12	13	15	16

Lower bound is 0.97522E 0	Last term 0.24782E - 1
Upper bound is 0.97738E 0	Last term 0.21627E - 2
Lower bound is 0.97723E 0	Last term 0.14357E - 3
Upper bound is 0.97726E 0	Last term 0.33038E - 4
Lower bound is 0.97726E 0	Last term 0.64429E - 6

4.2 An Algorithm to Determine the Reliability of a Complex System--Minimal Cuts and Coherent Systems Approach

1. Introduction

The method most often suggested for determining the reliability of a system is to construct a reliability network, enumerate from the network all mutually exclusive working states of the system, calculate the probability of occurrence of each working state, and sum these probabilities. For a complex system this is not a practical method because there is a very large number of working states. This computational problem is considerably alleviated by an approximation technique [25, 46, 62] , which by considering a much smaller set of states, called the set of minimal cuts, obtains a lower bound to system reliability. The contribution of this paper is an algorithmic procedure that generates the set of minimal cuts and determines from it the minimal cut approximation to reliability. This algorithm is a slight modification of a general procedure for finding all proper cuts of a linear graph described with proofs in [15].

2. Formulation of the Problem

Reliability Network

The components of the reliability network all have two terminals. Each component is bidirectional in that a path from S to t may traverse it in either direction. Capital letters identify the components as in Fig. 4.2.1.

Components are interconnected at nodes of the network. Integers identify the nodes as in Fig. 4.2.1. The set of all nodes is $N = (1,2,3,\ldots,n)$. Nodes S and t are number 1 and n, respectively. The numbering of all other

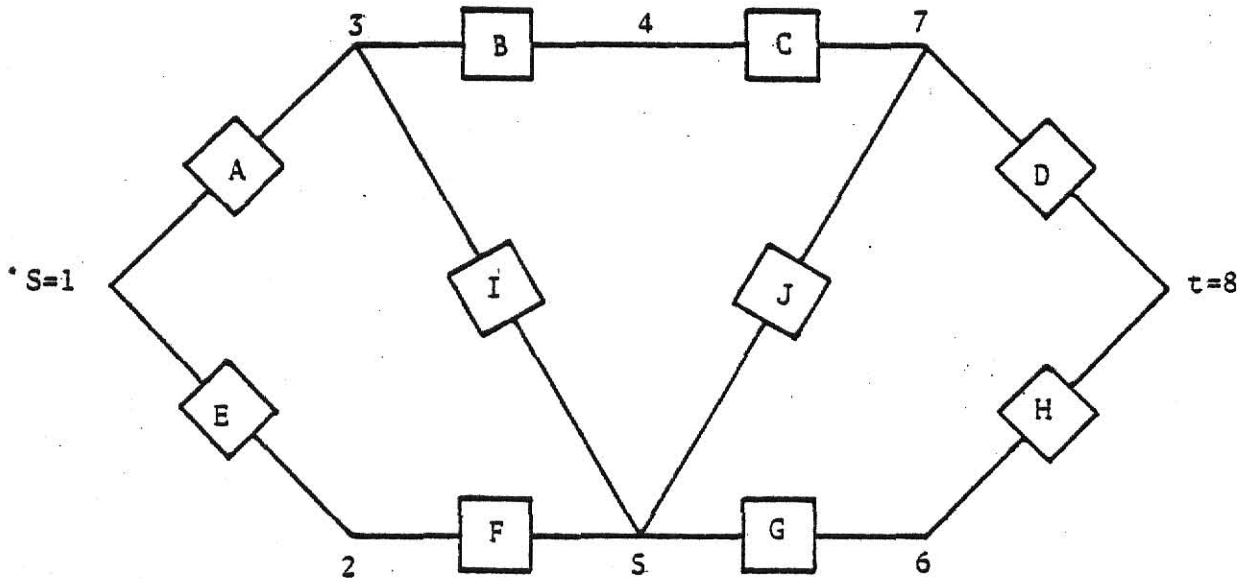


Fig. 4.2.1. Reliability network [36].

nodes is arbitrary. In the course of subsequent discussion, the nodes of the reliability network will be divided into two mutually exclusive subsets with node 1 in one subset and node n in the other. The subset of nodes including 1 is called X and the subset of nodes including n is called $\bar{X}$. So, $X \cup \bar{X} = N$.

A simple path from node a to node b in the reliability network is an ordered set of components such that the first member of the set and no others has a as a terminal, the last member and no others has b as a terminal, adjacent members of the set have common terminals and no more than two members of the set have the same common terminal. Thus a simple path from 1 to 8 in Fig. 4.2.1 is AIJD.

A subnetwork is said to be connected if there exists a simple path between every pair of nodes in the subnetwork. All the reliability networks treated here are assumed to be connected.

Reliability Approximation

Because of the large number of states for a complex system the exact reliability is very difficult to determine. Here a lower bound approximation based on the theory of Esary and Proschan [25] is provided. The approximation is very close when component reliabilities are close to unity. This is most often the case in practice.

Proschan and Esary's analysis is based on the concepts of minimal cuts and coherent systems. These terms refer to the effects of component failures on the operation of the network. A "coherent system" is defined by the following four conditions:

1) when a group of components in the system is failed causing the system to be failed, the occurrence of any additional failure or failures will not return the system to a successful condition:

2) when a group of components in the system is successful and the system is successful, the system will not fail if some of the failed components are returned to the successful condition:

3) when all the components in the system are successful the system is successful:

4) when all the components in the system are failed the system is failed.

If a system fulfills all these conditions, it is a coherent system. The reliability network described here are coherent systems, hence applicable to the Esary and Proschan analysis.

A "cut" is a set of components such that if they fail, the system will be failed regardless of the other components in the system. An example of a coherent system is shown in Fig. 4.2.2.

As long as any path through successful components exists between terminals 1 and 4 of the system, the system is said to be successful. A component failure opens the path between the two terminals of the component.

The cuts of this system are listed in Table 4.2.1. The failure of any of the cuts will cause the network of Fig. 4.2.2 to fail.

A "minimal cut" is defined as a cut in which there is no proper subset of components whose failure alone will cause the system to fail. From Table 4.2.1, the minimal cuts of network in Fig. 4.2.2 can easily be recognized. They are listed in Table 4.2.2 along with their probabilities of occurrence.

The lower bound approximation to reliability depends on the identification of all the minimal cuts in the network. Esary and Proschan found that a lower bound to system reliability is the probability that none of the system's minimal cuts fail.

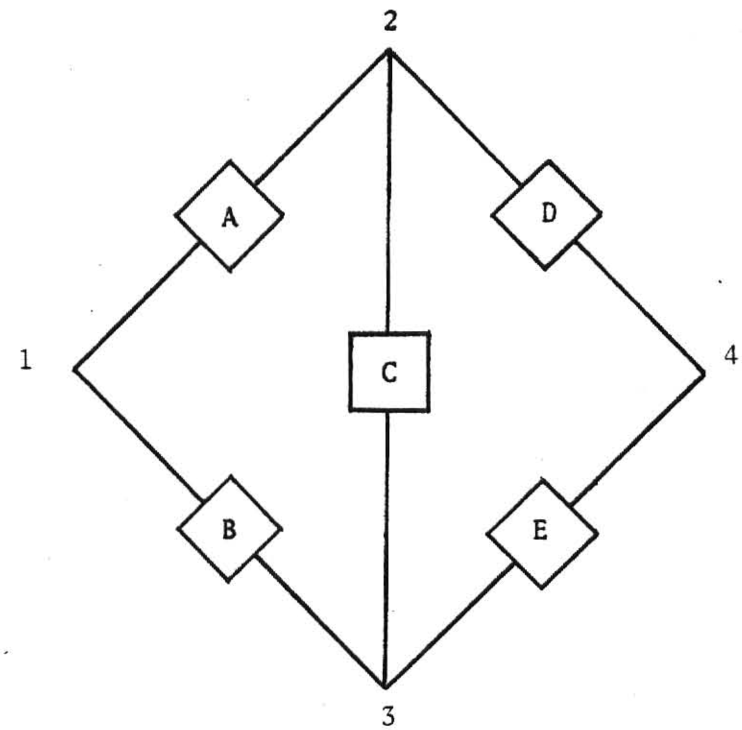


Fig. 4.2.2. Example of coherent system

TABLE 4.2.1

Cuts of the System Between Nodes 1 and 4 of Fig. 4.2.2.

Cut	Components in Cut
1	A, B
2	A, B, C
3	A, B, D
4	A, B, E
5	A, B, C, D
6	A, B, C, E
7	A, B, D, E
8	A, B, C, D, E
9	D, E
10	C, D, E
11	B, D, E
12	A, D, E
13	B, C, D, E
14	A, C, D, E
15	A, C, E
16	B, C, D

TABLE 4.2.2

Minimal Cuts Between Nodes 1 and 4 on Fig. 4.2.2.

Minimal Cut	Components in Minimal Cut	Probability of Failure of Minimal Cut
1	A, B	$q_A q_B$
2	D, E	$q_E q_D$
3	A, C, E	$q_A q_C q_E$
4	B, C, D	$q_B q_C q_D$

For the example, this lower bound R_{LB} is

$$R_{LB} = [1 - (q_A q_B)][1 - (q_D q_E)][1 - (q_A q_C q_E)] \times [1 - (q_B q_C q_D)].$$

This is not the true reliability expression because the failure of minimal cuts is assumed to occur independently. This is not always true of course since one component may appear in several minimal cuts.

This relationship can be written for general systems if the j^{th} minimal cut is denoted by the set S_j. The members of the j^{th} minimal cut are given by $i \varepsilon S_j$.

The lower bound to the system reliability is the probability that none of the system's minimal cuts fail or

$$R_{LB} = \prod_{\text{all}_j} \left[1 - \prod_{i \varepsilon S_j} (q_i) \right]$$

This is the approximation used here.

3. Algorithm to Determine the Set of Minimal Cuts.

The network of Fig. 4.2.1 is used for illustration. Table 4.2.3 lists the 16 minimal cuts of this network. This list is the output of the algorithm of this section.

The algorithm provides for the construction of a tree. The tree for the network of Fig. 4.2.1 appears in Fig. 4.2.3. The tree consists of vertices and edges. Edges are the line segments on the tree and vertices are the points. Vertices are given integer indices for identification. The vertex of the tree indexed 0 is called the root vertex.

Table 4.2.3

Minimal Cuts of the Network of Fig. 4.2.1 Specified by the Set

Cut	Members of the Set X 1	2	3	4	5	6	7	8	Components in the cut
0	x								AE
1	x		x						BEI
2	x		x	x					CEI
3	x		x	x			x		DEIJ
4	x	x							AF
5	x	x			x				AGIJ
6	x	x			x	x			AHIJ
7	x	x	x						BFI
8	x	x	x	x					CFI
9	x	x	x	x			x		DFIJ
10	x	x	x		x				BGJ
11	x	x	x		x	x			BHJ
12	x	x	x	x	x				CGJ
13	x	x	x	x	x	x			CHJ
14	x	x	x	x	x		x		DG
15	x	x	x	x	x	x	x		DH

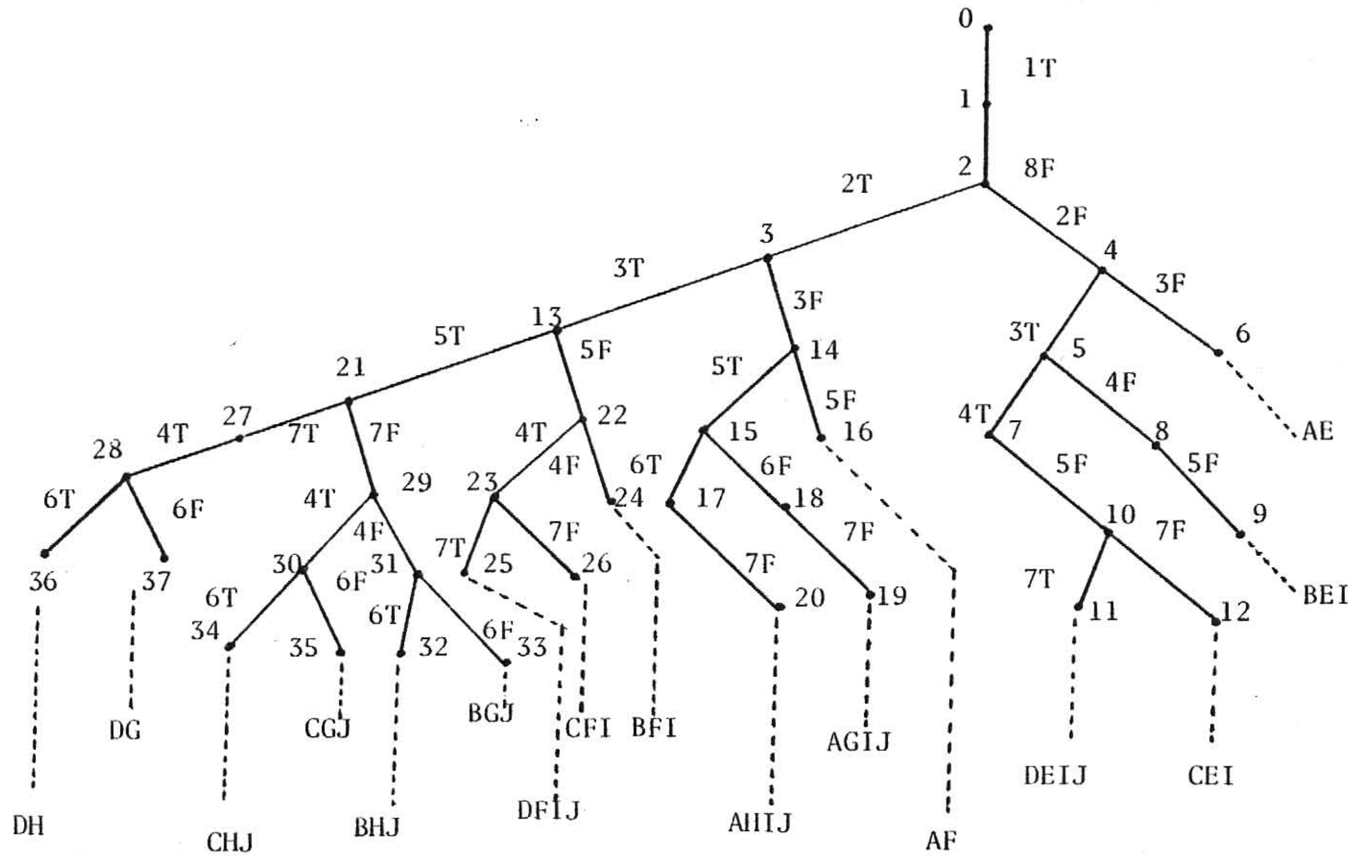


Fig. 4.2.3. Complete minimal-cut generation tree [15,36].

Minimal cuts of the network are represented in the tree by vertices that touch only one edge (excluding vertex0). These are called terminal vertices. For instance in Fig 4.2.3, vertex 20 represents cut AHIJ. The algorithm assures that every such vertex represents a unique minimal cut and that every minimal cut is represented by a terminal vertex of the tree.

Edges of the tree take on labels of the form xT or xF. Here x is an integer representing a node of the reliability network or a member of the set N. T and F are used to indicate members of the sets X and $\bar{X}$ for minimal cut of the reliability network. To illustrate this, note that on the unique path from vertex 0 to vertex 20 the labels 1T, 2T, 5T, 6T appear. These labels identify the set X for the cut represented by vertex 20 as [1, 2, 5, 6]. Table 3 will indicate the correspondence between this set X and the cut AHIJ.

The labels 8F, 3F, and 7F also appear on the path from 0 to 20. Note that nodes 3, 7, and 8 are included in the set $\bar{X}$ = [3, 4, 7, 8]. Generally, the nodes labeled F will not constitute the entire set $\bar{X}$. The set $\bar{X}$ is found for a terminal vertex by the operation $\bar{X}$ = N-X.

The tree is constructed sequentially. It initially consists of no vertices and no edges. The algorithm creates vertices and edges in the course of its accomplishment. Every time the algorithm generates a terminal vertex of the tree, a minimal cut has been found and the contribution of the cut to the unreliability of the network is calculated. The labels are chosen for the edges of the tree in such a way that for every terminal vertex, the sets X and $\bar{X}$ define connected subnetworks (necessary conditions for a minimal cut).

Associated with each vertex i of the tree are four subsets of nodes of the reliability network Y_{1i}, Y_{2i}, Y_{3i}, and W_i, Y_{1i}, Y_{2i}, and Y_{3i} are defined in following manner.

Find the unique simple path ℓ_i that connects the root vertex to vertex i. Identify the sets Y_{1i}, Y_{2i}, and Y_{3i} according to the following definitions:

Node $x \in Y_{1i}$ is an edge in the path ℓ_i is labeled xT.

Node $x \in Y_{2i}$ if an edge in the path ℓ_i is labeled xF.

Node $x \in Y_{3i}$ if it is in the set N but not in the sets Y_{1i} or Y_{2i}.

A node x will be a member of W_i if it is a member of Y_{3i} and if it is a terminal of a component whose other terminal is in the set Y_{1i}.

Initially, the estimate of system unreliability is taken as $Q_o = 0$. This is modified as minimal cuts are discovered.

The algorithm that generates all minimal cuts between given nodes s and t will now be described. One should follow the steps of the algorithm on the example tree of Fig. 4.2.3.

Algorithm for Determining the Set of All Minimal Cuts of a Network Between Nodes s and t.

1) Create three vertices for the tree indexed 0, 1, and 2, and edges (0, 1) and (1, 2) labeled sT and tF, respectively. Let vertices 0 and 1 be scanned and vertex 2 be unscanned. Vertex 0 is called the root vertex. Go to Step 2).

2) Choose the unscanned vertex with the greatest index and mark it scanned. If there are no unscanned vertices, the algorithm terminates for the complete tree has been generated. The vertex chosen will be denoted as vertex i. Find the unique simple path ℓ_i that connects the root to vertex i. Identify the sets Y_{1i}, Y_{2i}, Y_{3i}, and W_i as defined above. Choose y, an element of the set W_i. If W_i has no members go to Step 7) Construct the subnetwork defined by the set of nodes $Y_4 (= Y_{2i} \cup Y_{3i} - y)$. Test to see if it is connected. If not, go to Step 4). If so, go to Step 3).

3) Create two new vertices indexed k and k+1 where k is 1 greater than the number of vertices currently in the tree. Vertices k and k+1 are unscanned. Create two new edges (i, k) and (i, k+1), labeled yT and yF, respectively. Go to Step 2).

4) The subnetwork defined by Y_4 is not connected. Find the set of nodes Y_5 that defines the connected subnetwork that includes node t. If $Y_{2i} \subset Y_5$, go to Step 5) If $Y_{2i} \not\subset Y_5$, go to Step 6)

5) Create vertex k and edge (i, k) labeled yT where k is one greater than the number of vertices currently in the tree. Determine the set $Y_6 = Y_4 - Y_5$. For each number $z \in Y_6$ create a vertex of the tree and an edge labeled zT. If $|Y_6|$ is the number of members in the set Y_6, vertices k + 1, k + 2, ..., k + $|Y_6|$ will be created. Edges (k, k + 1), (k + 1, k + 2). ..., (k + $|Y_6|$ - 1, k + $|Y_6|$) will also be created. Finally, create vertex k + $|Y_6|$ + 1 and edge (i, k + $|Y_6|$ + 1) labeled yF. Go to Step 2)

6) Create one new vertex indexed k and an edge (i, k) labeled yF. Go to Step 2)

7) A minimal cut has been generated at this step. The set X_i for the cut is $X_i = Y_{1i}$ and $\bar{X}_i = N - Y_{1i}$. The components in the minimal cut are those that have one terminal in the set X and one in the set $\bar{X}$. Let these components be the set S. Find the probability of failure of the minimal cut:

$$q_s = \prod_{i \in S} q_i$$

Include this in the system unreliability estimate:

$$Q_i = Q_{i-1} + q_s - Q_{i-1} \cdot q_s$$

where Q_{i-1} is the unreliability estimate before the discovery of this cut.

To generate more minimal cuts, go to Step 2).

A detailed proof of the algorithm appears in [4] and [7]. Roughly speaking, the algorithm sequentially generates [Step 2)] the sets X and $\bar{X}$ (represented by Y_{1i} and Y_{2i}, respectively, in the algorithm). The choice of a member of W_i for expanding the set X guarantees that the subnetwork defined by set X is connected. This fulfills the first requirement for a minimal cut. Each subnetwork thus generated also defines a complementary subnetwork that includes t but not s (the network defined by Y_4).

At every step this subnetwork is tested [Step 2)] to see if it is connected (the second requirement for a minimal cut). If it is, the generation procedure continues with Step 3). If it is not connected and if it includes no members already assigned to $\bar{X}$, then that part of the subnetwork not including t is joined to the subnetwork including s [Step 5)]. If it includes a node already assigned to $\bar{X}$ that avenue of expansion for the set X is closed [Step 6)]. Step 4) decides which of these alternatives is to be taken.

If at some point in the generation process the set X cannot be expanded in such a way to guarantee that the subnetwork defined by X is connected (the way is blocked by nodes already assigned to $\bar{X}$), then the process stops momentarily for a minimal cut has been generated. The unreliability of the network is modified [Step 7)] to include this minimal cut, and the process returns to Step 2 to continue its search for more minimal cuts.

The absence of unscanned nodes signals termination of the algorithm.

4. Space-Saving Modification

The algorithm is designed to generate the entire tree and maintain it in the core memory of the computer. In the computer program implementation of the algorithm, a great deal of space in the core memory of the computer must be set aside to keep all the information concerning the vertices, edges, and labels of the tree.

A very simple modification to the algorithm is possible that at any point in the generation process allows one to keep only that portion of the tree that is necessary to discover the set of minimal cuts that have not yet been generated.

The modification takes place in the first paragraph of Step 2):

> 2) Choose the unscanned vertex with the greatest index. Let this be i. Discard those vertices in the tree with indices greater than i. Mark vertex i scanned. If there are...

The algorithm then continues as before.

With this change the next vertex of the tree to be generated is indexed i + 1. At no time in the generation process will there be more than 2n vertices in the tree.

Note that no minimal cuts are lost by this process because all the vertices discarded have been previously scanned.

Of course in the discard process, terminal vertices representing cuts are also discarded. Since the probability of occurrence of each cut is included in the system unreliability as the cut generated, this does not affect the value of the reliability approximation.

With this change, the process of cut generation is practically limited only by the computer time one is willing to expend, while without this change the process is limited by the core size of the computer.

5. Computational Considerations and Conclusions

The minimal cut generation algorithm has been programmed for the IBM 7094 computer in FORTRAN IV. For the system reliability application, the algorithm with the modification of the last section is of primary interest.

The time required for the program cannot be measured entirely in terms of the number of nodes in the network n.

Depending on the configuration of the network the number of minimal cuts between two nodes varies from n - 1 to 2^{n-2}. There are n - 1 cuts if the network is a simple chain and s and t are at the ends of the chain, and 2^{n-2} cuts if there is a component between every pair of nodes. For n, any reasonable number, this is a very great range. The program with the modification handles problems having up to 125 nodes at the lower bound and problems having perhaps 20 nodes at the upper bound. The lower bound case is space limited and the upper bound case is time limited. It would take the algorithm about one hour to generate 2^{18} minimal cuts.

Table 4.2.4 shows the cut generation time for several randomly generated reliability networks. These times do not include the time required to calculate Q_i for each cut. Of course, the time depends strongly on the number of cuts to be generated.

The space requirement of the program is determined not by the number of cuts it may generate but by the number of nodes in the network. The primary space consumer of the FORTRAN IV program that implements this algorithm is an n x n binary matrix in which a 1 in cell (i,j) indicates the presence of a component between nodes i and j and a 0 indicates its absence. The maximum dimensions on this matrix have been 125 x 125. Other methods of storing the matrix trade space efficiency for time efficiency, and since time is the limitation for most networks of interest, there has been no stimulus to provide a more efficient storage scheme.

This paper has presented a technique for very rapidly enumerating the minimal cuts and hence determining a lower bound to the reliability of networks consisting of bidirectional components. Typical of such networks are communications and relay networks. The methods are extendable to unidirectional

TABLE 4.2.4

Computation Time for Cut Generation

Network	Number of Nodes	Number of Components	Number of Minimal Cuts	Total time (seconds)	Average Time per Cut (ms)
1	10	45	256	1.5	6.0
2	12	66	1024	6.0	5.9
3	13	20	54	0.7	13.0
4	14	19	29	0.5	17.2
5	20	27	1250	21.5	17.2
6	22	31	3255	62.4	19.2

components with some modifications. The algorithm has been programmed for a digital computer and is very efficient. The practicality of the methods depends, however, on the number and interconnection pattern of the components.

4.3 A Boolean Algebra Method for Computing the Terminal Reliability in a Communication Network

1. Introduction

An efficient algorithm for the analysis of unreliable communication networks is proposed. Here a communication network will be represented by an oriented graph with weighted arcs and unweighted nodes. The nodes represent stations, assumed to be completely reliable; the weighted arcs represent direct unreliable connections, that is, connections which are available on the average only for a given percentage of time. The weight assigned to each arc is its probability of existence and is also known as arc reliability. This kind of probabilistic graph may be a meaningful model for some communication networks.

We assume that there does not exist any correlation between failures of different links and that the arc reliabilities are constant during the time interval in which the reliability of the network is being examined.

2. Review of the Existing Methods

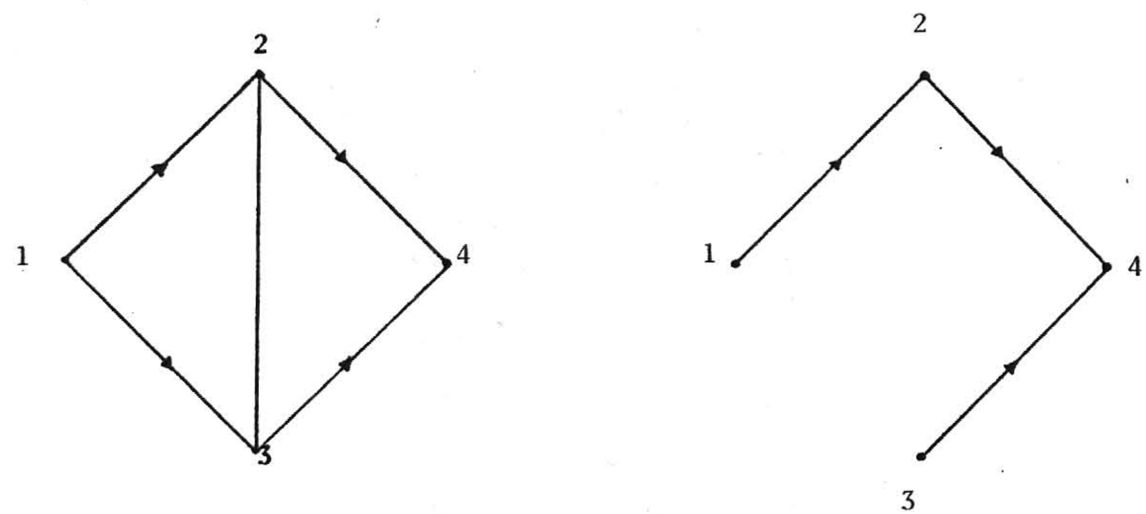
Let us consider the four-node graph represented in Fig. 4.3.1(a), where the probability p_{ij} of existence of the arc (i, j) is given. Suppose we want to compute the terminal reliability between vertices 1 and 4. For each arc (i, j) of the graph, we can now define a stochastic variable X_{ij} having {0, 1} as definition domain. The statistic distribution of X_{ij}, is, of course,

$$P(X_{ij} = 0) = 1 - P_{ij} = q_{ij}$$

$$P(X_{ij} = 1) = P_{ij}$$

Having 5 binary stochastic variables, $2^5 = 32$ elementary events have to be considered. Each event corresponds to a subgraph of the given graph. For instance, the event

$$E = \{X_{12} = 1,\ X_{13} = 0,\ X_{23} = 0,\ X_{24} = 1,\ X_{34} = 1\} \qquad (1)$$



(a) (b)

Fig. 4.3.1. (a) An example of probabilistic graph. (b) A subgraph corresponding to a favorable event for the connectivity between vertex 1 and vertex 4.

corresponds to the subgraph in Fig. 4.3.1(b). Event probabilities are easily computed since all variables are independent by assumption. For example,

$$P_E = P_{12} \cdot q_{13} \cdot q_{23} \cdot p_{24} \cdot p_{34}$$

Since we are interested in computing the terminal reliability between nodes 1 and 4, an event is considered favorable if at least one path exists in its subgraph from node 1 to node 4. For instance, the event E in (1) is favorable. As a consequence, a binary function F can be defined which associates with each event a value 1 or 0 according to whether or not it is favorable.

In Fig. 4.3.2(a) we show a Karnaugh map defining function F for our graph in Fig. 4.3.1(a). Event E corresponds to the shaded square. The terminal reliability is, by definition,

$$p = \sum_{F(E) = 1} P_E \tag{2}$$

Thus in our example from Fig. 4.3.2(a) we get

$$\begin{aligned} P = {} & q_{12}p_{13}q_{23}q_{24}p_{34} + q_{12}p_{13}q_{23}p_{24}p_{34} \\ & + p_{12}p_{13}q_{23}q_{24}p_{34} + p_{12}p_{13}q_{23}p_{24}p_{34} \\ & + p_{12}p_{13}q_{23}p_{24}q_{34} + p_{12}q_{13}q_{23}p_{24}p_{34} \\ & + p_{12}q_{13}q_{23}p_{24}q_{34} + q_{12}p_{13}p_{23}q_{24}p_{34} \\ & + q_{12}p_{13}p_{23}p_{24}p_{34} + q_{12}p_{13}p_{23}p_{24}q_{34} \\ & + p_{12}p_{13}p_{23}q_{24}p_{34} + p_{12}p_{13}p_{23}p_{24}p_{34} \\ & + p_{12}p_{13}p_{23}p_{24}q_{34} + p_{12}q_{13}p_{23}q_{24}p_{34} \\ & + p_{12}q_{13}p_{23}p_{24}p_{34} + p_{12}q_{13}p_{23}p_{24}q_{34} \end{aligned} \tag{3}$$

As we have just shown, it is very easy to compute the probability for each connecting subgraph, but, unfortunately, the number of these subgraphs increases very rapdily with the number of arcs.

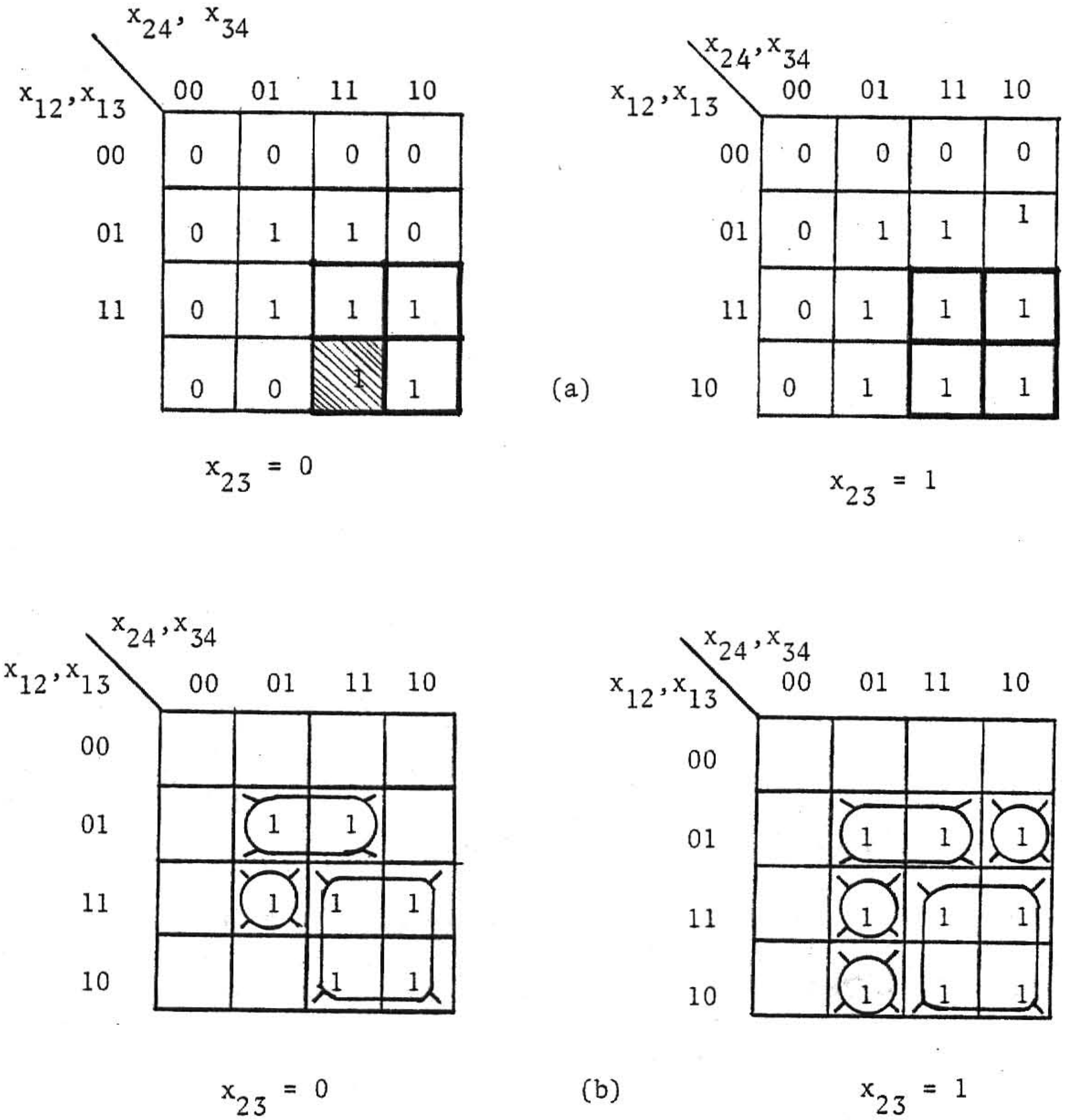


Fig. 4.3.2. (a) Karnaugh map defining the binary function for graph in Fig. 1(a). (b) Set of nonoverlapping implicants covering the Karnaugh map.

The second method described in the literature [39] considers larger events, corresponding to the simple paths[1] between the terminal nodes. For instance, consider the simple path (1, 2, 4) in Fig. 4.3.1(a); the (non-elementary) event

$$E' = \{X_{12} = 1, \quad X_{24} = 1\}$$

can be associated to it. We will have

$$p_{E'} = p_{12} \cdot p_{24}$$

while the corresponding implicant $X_{12}X_{24}$ will cover the eight masked in the Karnaugh map of Fig. 4.3.2(a).

By considering all simple paths between nodes 1 and 4 we completely cover the Karnaugh map i.e., we take into account all the events which contribute to the computation of the terminal reliability.

Unfortunately, these new events, the simple paths, are no longer disjoint and the terminal reliability is given by the probability of the union of the events corresponding to the existence of the paths. This can be computed by applying the inclusion and exclusion principle but the expansion of the union becomes a very difficult task.

To apply the second method to our example we need all simple paths between nodes 1 and 4 in Fig. 4.3.1(a) that can be found here by inspection:

path 1:1,2,4 path 2:1,3,4

path 3:1,2,3,4 path 4: 1,3,2,4

The corresponding probabilities are

$$p_1 = p_{12}p_{24} \qquad p_2 = p_{13}p_{34}$$

$$p_3 = p_{12}p_{23}p_{34} \qquad p_4 = p_{13}p_{23}p_{24}$$

[1]A path is called simple if no node is traversed more than once.

We also have to compute the following joint probabilities:

$$p_{1,2} = p_{12}p_{13}p_{24}p_{34} \quad p_{1,3} = p_{12}p_{23}p_{24}p_{34} \quad p_{1,4} = p_{12}p_{13}p_{23}p_{24}$$

$$p_{2,3} = p_{12}p_{13}p_{23}p_{34} \quad p_{2,4} = p_{13}p_{23}p_{24}p_{34} \quad p_{3,4} = p_{12}p_{13}p_{23}p_{24}p_{34}$$

$$p_{1,2,3} = p_{1,2,4} = p_{1,3,4} = p_{2,3,4} = p_{12}p_{13}p_{23}p_{24}p_{34}$$

$$p_{1,2,3,4} = p_{12}p_{13}p_{23}p_{24}p_{34}$$

The desired terminal reliability is given by the following formula:

$$p = \sum_i p_i - \sum_{i,j} p_{ij} + \sum_{i,j,k} p_{i,j,k} - p_{i,j,k,h}$$

Modifications of the previous methods are presented in [18, 47, 52].

3. Representation of a Set of Simple Paths as a Boolean Sum of Disjoint Products

We can see that the two methods presented in previous section are similar, but start the solution procedure from two different forms of the same Boolean function F. Both forms are sum of Boolean products; the difference is that in the first form the Boolean products represent all the elementary events (obviously disjoint), while in the second case they represent nondisjoint implicants.

The complexity of these methods is caused in the first case by the large number of elementary events and in the second case by the difficult computation of the probability of the sum of nondisjoint events. It is clear that a more efficient method should avoid the difficulties previously seen by representing the Boolean function F with a sum of nonelementary disjoint products, where the number of these products is therefore not so large.

This corresponds to covering the Karnaugh map by nonoverlapping implicants and can be obtained, as shown in Section 4, by performing simple algebraic operations on the initial form obtained from the set of all simple paths.

Once the desired Boolean form is obtained, the arithmetic expression giving the terminal reliability is straightforwardly computed by means of the following correspondences:

$x_{ij} \to p_{ij}$

$\bar{x}_{ij} \to q_{ij}$

Boolean sum $\to$ arithmetic sum

Boolean product $\to$ arithmetic sum (4)

Let us consider the graph in Fig. 4.3.1(a). On the Karnaugh map in Fig. 4.3.2(a) we can easily find by inspection a set of nonoverlapping implicants [marked in Fi 4.3.2(b)] which give the following disjoint form for the Boolean function F:

$$F = X_{12}X_{24} + \bar{X}_{12}X_{13}X_{34} + X_{12}X_{13}\bar{X}_{24}X_{34} + \bar{X}_{12}X_{13}X_{23}X_{24}\bar{X}_{34} + X_{12}\bar{X}_{13}X_{23}\bar{X}_{24}X_{34} \tag{5}$$

Notice that the terms in (5) are only five (and simpler) instead of sixteen as in (3). By translating (5) according to (4), we obtain

$$p = p_{12}p_{24} + q_{12}p_{13}p_{34} + p_{12}p_{13}q_{24}p_{34} + q_{12}p_{13}p_{23}p_{24}q_{34} + p_{12}q_{13}p_{23}q_{24}p_{34} \tag{6}$$

This formula gives us a very simple expression to compute the terminal reliability between node 1 and 4.

4. Computational Procedure

The solution of our original problem is now reduced to finding a sum of products for the Boolean function F in which all terms are disjoint. This computation is schematically described by the following Algorithm A.

Algorithm A

Step 1: Let S be the set of all simple paths between the terminal nodes. Construct a Boolean sum of products f, where each product corresponds to a path ε S and whose factors are exactly the noncomplemented variables corresponding to the arcs in the path.

Let P = 0.

Step 2: If f has no terms, stop.

Step 3: Select any term A of f.

Step 4. Let A' be the arithmetic monomial which is equivalent to A according to (4).

Let P = P + A'.

Step 5: Let $f \leftarrow \bar{A} \cdot f$ and reduce f to a sum of products.

Go to Step 2.

Let us apply this algorithm to our example:

Step 1:

$$f = X_{12}X_{24} + X_{13}X_{34} + X_{12}X_{23}X_{34} + X_{13}X_{23}X_{24} \quad P = 0.$$

Step 2: f has at least one term.

Step 3: $A = X_{12}X_{24}$

Step 4: $A' = p_{12}p_{24} \quad P = p_{12}p_{24}$

Step 5:

$$\begin{aligned} f &= \overline{X_{12}X_{24}} \cdot (X_{13}X_{34} + X_{12}X_{23}X_{34} + X_{13}X_{23}X_{24}) \\ &= (\bar{X}_{12} + \bar{X}_{24}) \cdot (X_{13}X_{34} + X_{12}X_{23}X_{34} + X_{13}X_{23}X_{24}) \\ &= \bar{X}_{12}X_{13}X_{34} + X_{13}\bar{X}_{24}X_{34} + X_{12}X_{23}\bar{X}_{24}X_{34} + \bar{X}_{12}X_{13}X_{23}X_{24}) \end{aligned}$$

Step 2: f has at least one term

Step 3: $A = \bar{X}_{12}X_{13}X_{34}$

Step 4:

$$A' = q_{12}p_{13}p_{34} \qquad P = p_{12}p_{24} + q_{12}p_{13}p_{34}$$

and so on until the final result is obtained.

$$P = p_{12}p_{24} + q_{12}p_{13}p_{34} + p_{12}p_{13}q_{24}p_{34} + q_{12}p_{13}p_{23}p_{24}q_{34} + p_{12}q_{13}p_{23}q_{24}p_{34}$$

Note:

At Step 3 the selection of the implicant A can be performed according to different criteria. A very good one could be to choose the implicant whose probability is maximum.[2] That gives as a result an algorithm with the fastest convergence but requires at each iteration the computation of the probability corresponding to all terms of the Boolean form.

Other criteria are to select the implicant with the minimum number of factors (as we did in the example) or the minimum number of complemented factors. The latter takes advantage of the fact that the q_i are always (at least in the communication networks) much smaller than the p_i and thus it works very often as the maximum probability criterion.

Another important remark about Algorithm A concerns the simplification of the Boolean form f computed in Step 5. The simplest way of reducing f to a sum of products consist of applying the well-known identity

$$\overline{x_1x_2} = \bar{x}_1 + \bar{x}_2$$

for computing $\bar{A}$ and then using distributivity.

[2]This criterion is clearly not applicable if the result must be given in symbolic form.

However, the resulting sum may be very clumsy, so that some form of simplification may be convenient. If F is the Boolean function represented by f (which changes at every step), we can proceed in one of the following ways.

1) Absorption Law: Erase all the terms which imply some other term of f.

2) Prime Implicant Form: Starting from f and using the consensus algorithm generate all prime implicants of f.

3) Irredundant Form: Select a set of prime implicants covering F such that no one of them is redundant.

4) Minimal Form: Select a set of prime implicants covering F and having the minimal total number of factors.

Note that the initial form of f (obtained from the set of all simple paths) is minimal since it contains only essential prime implicants.

The more sophisticated the selection criteria and the simplification methods are, the simpler the final expression for P should be. However, this is only an heuristic rule: the problem of finding an optimal expression for P (for instance in the sense of minimal number of multiplications) is probably very difficult.

Algorithm B (Approximation Techniques)

When we want to compute the exact value of the terminal reliability in quite large networks, we must face the large amount of computations required and there is no way to avoid it even applying the method previously discussed. Moreover, in many applications it is enough to know an approximate value; for instance, in network synthesis the terminal reliability is assumed as a measure of the efficiency of a given topology and it is evaluated many times in an optimization procedure.

If we examine Algorithm A, we notice that at each iteration the current value of P is increased by a positive quantity corresponding to the probability of the selected implicant. This means that at each iteration the procedure gives an estimate with positive error of the terminal reliability. We call error e_i the difference between the exact value of the terminal reliability and the current value of p at the i^{th} iteration. By definition, e_i is non-negative and monotonically decreasing with i.

Knowing the value of the error or at least an upper bound of it, we could evaluate how good the current estimate of the terminal reliability is and then implement a technique which stops the Algorithm A when the required precision is achieved.

An approximation technique is described in the following.

Step 1: Let S be the set of all simple paths between the terminal nodes. Construct a Boolean sum of products f, where each product corresponds to a path ϵ S and whose factors are exactly the noncomplemented variables corresponding to the arcs in the path.

Let P = 0

Step 2: Let h = 0

Step 3: If f has no terms, stop.

Step 4: Let f* = f + h. Transform from f* into an arithmetic form f' using correspondences (4). Evaluate f' using the given probabilities. If this value R_i does not exceed a given error ϵ, stop.

Step 5: Select any term A of f.

Step 6: Let A' be the arithmetic monomial which is equivalent to A according to (4).

Let P = P + A'.

Step 7: Let $g \leftarrow \bar{A} \cdot f$ and reduce g to a sum of products. Represent g as the sum of two functions $g = g' + g''$ such that g' contains exactly all terms of g with a number of complemented variables smaller than a given threshold T and g'' all the other terms. Let $f = g'$ and $h = h + g''$. Go to Step 3.

The desired upper bound to the error e_i is given by the residue R_i defined at the i^{th} iteration of the Algorithm B as the value of the arithmetic expression obtained transforming f* according to (4). (See Step 4.)

As the implicants of f* are not disjoint we have

$$R_i \geq e_i, \qquad \forall i.$$

An important property of Algorithm B is described by Theorem 1.

Theorem 1

The polynomial P computed by Algorithm B contains exactly all terms of polynomial P computed by Algorithm A which have a number of q-variables smaller than the threshold T.

Proof: It is easy to verify that during the multiplication performed in Step 7, every term of f generates only terms with a number of complemented variables not smaller than its own. Therefore, terms which have been erased from f in Algorithm B cannot generate in Algorithm A terms with a number of complemented variables smaller than T.

Taking into account that usually the values of q-variables are much smaller than the values of p-variables ($q = 0.01 \div 0.1$ $p = 0.9 \div 0.99$), meaning of Theorem 1 is that the terms computed by Algorithm B are the usually exactly the largest terms among those computed by Algorithm A.

The preceding statement allows us to believe that the approximation technique converges quite rapidly. This is corroborated by the experimental results shown in section 5.

It is interesting to observe that Algorithm B can stop either at Step 3 or 4. If it stops at Step 4, the constraint on the error is satisfied;

otherwise, it is necessary to repeat the procedure assuming a higher value of T.

Given an error ε, the optimum value of T is the minimum which let the Algorithm B stop at Step 4. In fact in that condition Algorithm B is as fast as possible.

5. Examples

A computer program implementing Algorithms A and B has been written in FORTRAN IV on an IBM 360/67 computer. A block diagram of this program is given in Fig. 4.3.3.

After reading a matrix specifying graph topology and arc probabilities (with information about directedness of arcs), the program simplifies the graph as much as possible using the series-parallel reduction shown in Fig. 4.3.4.

The path finding algorithm is then applied. In this algorithm a path is represented as a sequence of nodes and therefore a conversion to the "set of arcs" representation is necessary in the following stage. In this stage the Boolean variables x_{ij} are also created. The rest of the program follows Algorithms A and B in detail.

This program has been applied to the computation of the terminal reliability between UCLA and CMU in the ARPA computer network shown in Fig. 4.3.5.

The program was completely executed in 112s. The network after series-parallel reduction is shown in Fig. 4.3.6 and all simple paths in the vertex and arc representation are listed in Fig. 4.3.7(a) and (b), respectively.

The 53 final disjoint terms are shown in Fig. 4.3.8 where a + (-) in row k and column (i,j) appears as a factor.

Finally, the upper and lower bounds given at different iterations of the algorithm in the case of arc reliability 0.9 and 0.99 are shown in Table 4.3.1.

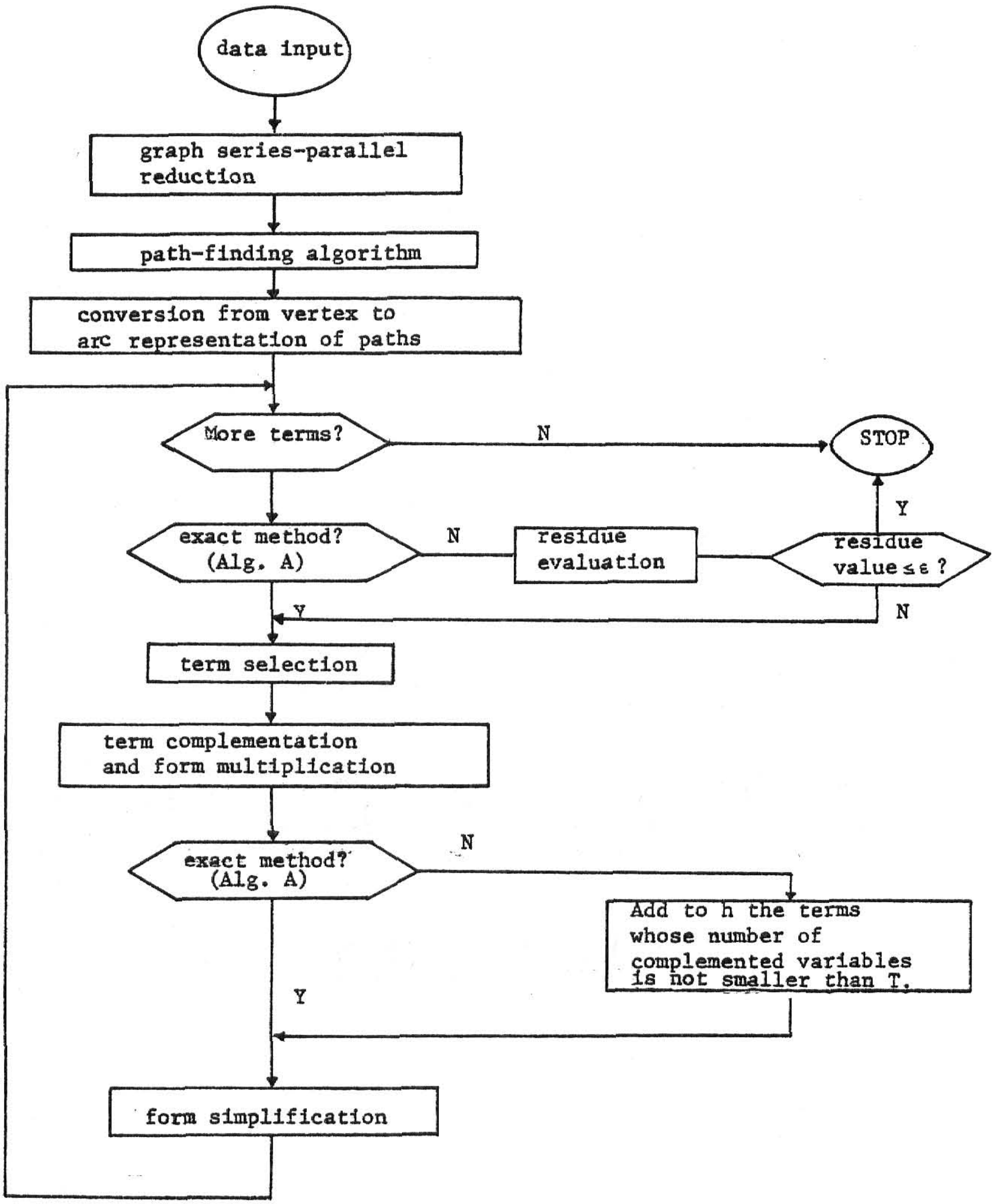


Fig. 4.3.3. Block diagram of the program implementing both Algorithm A and B. [28]

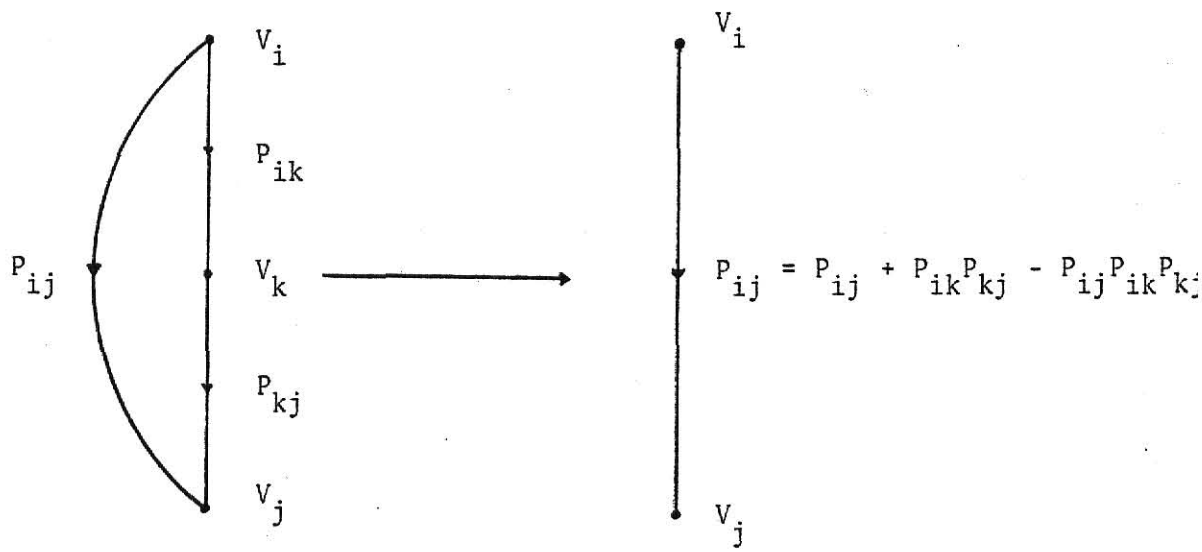


Fig. 4.3.4. Example of series-parallel reduction.

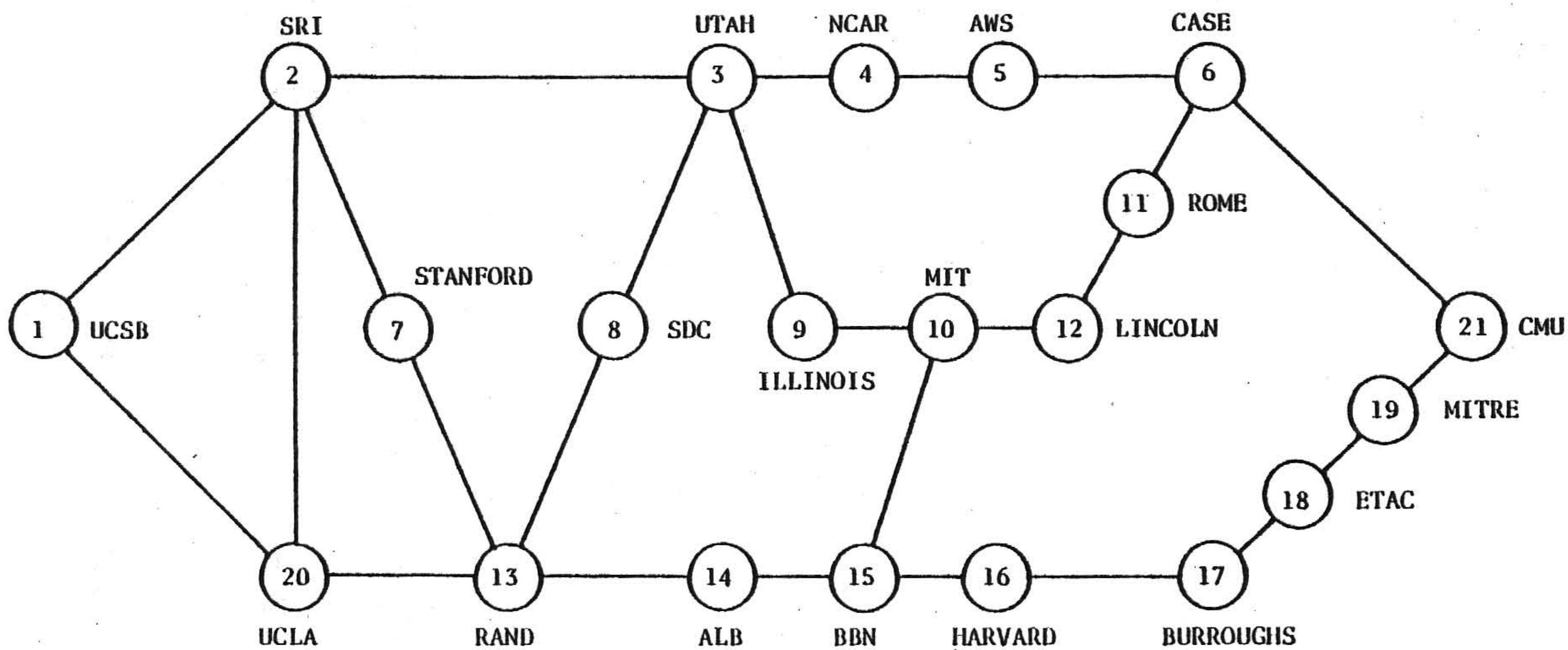


Fig. 4.3.5. Topology of the ARPA computer network [28].

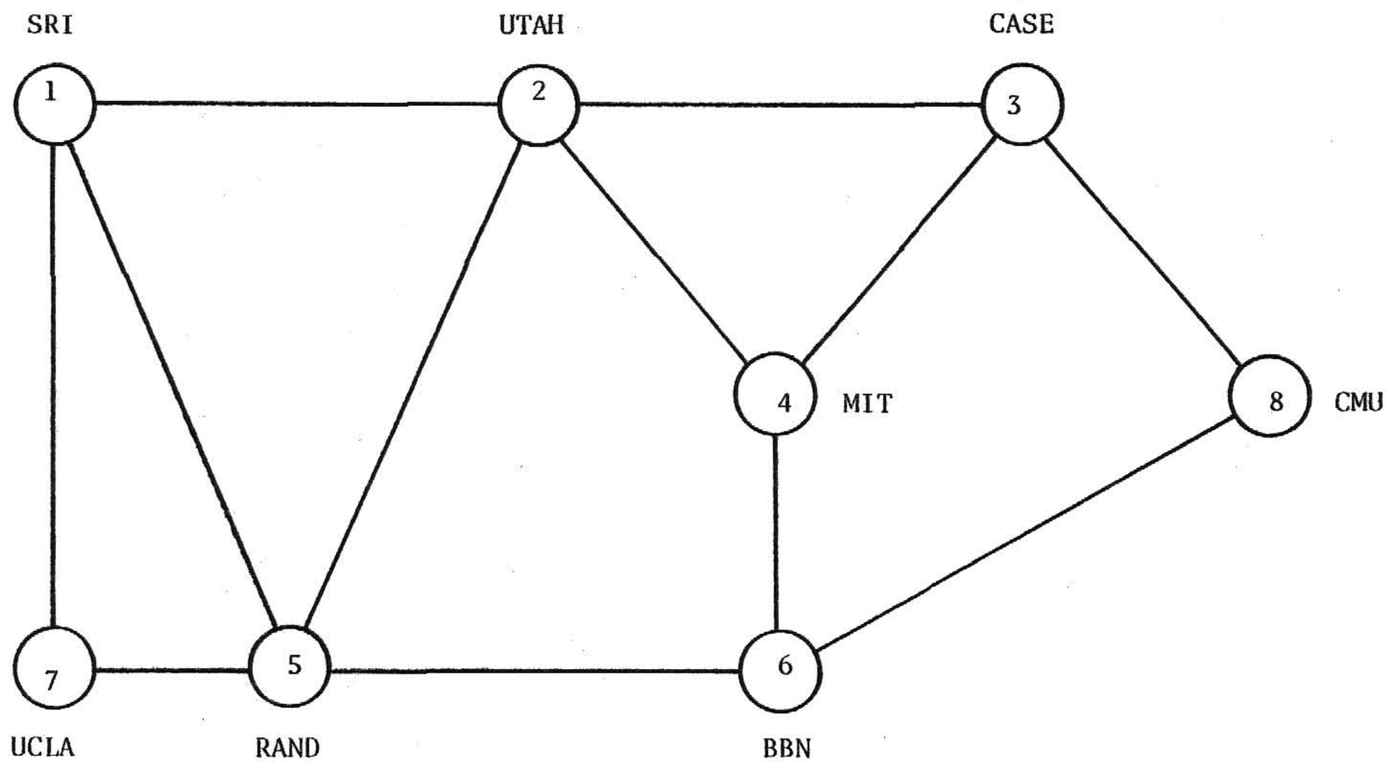
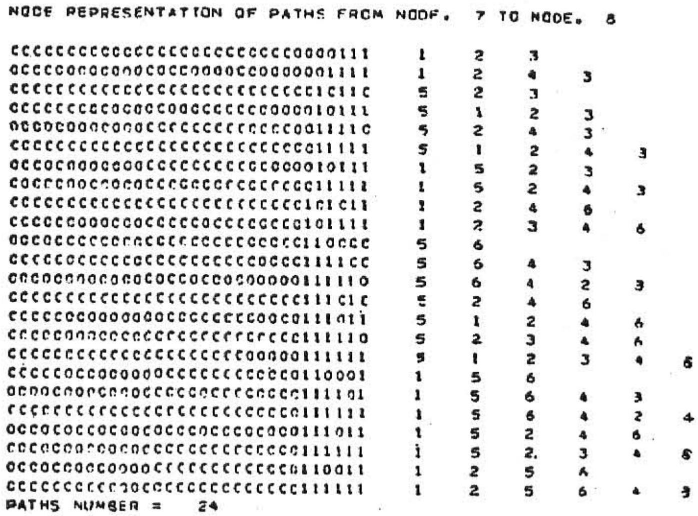
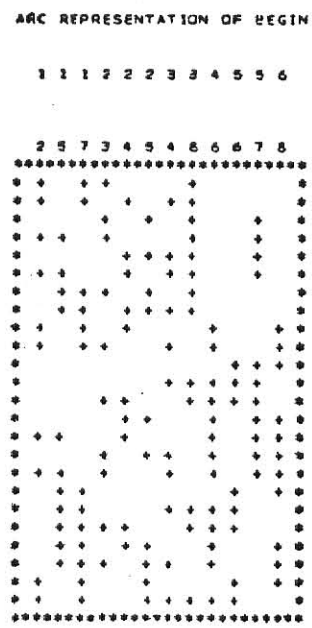


Fig. 4.3.6. ARPA computer network after series-parallel reduction.



(a)



(b)

Fig. 4.3.7. All simple paths between UCLA and CMU in the network of Fig. 4.3.6. They are represented as sequences of vertices with the pattern word (a) and as sets of arcs in (b) [28].

12	15	17	23	24	25	34	38	46	56	57	68
									+	+	+
+		+	+				+		-		
	+	+							+	-	+
			+		+		+		+	+	-
		-	+		+		+		-	+	
+		+	-	+		+	+		-		
-		+	+		+		+		-	+	
+		+	+		-		+		+		-
			-			+	+	+	+	+	-
				+	+		-	+	-	+	+
+	-	+			+				+	-	+
	+	+	+		+		+		+	-	-
+	+	-	+		-		+			+	-
-			-	+	+	+	+		-	+	
+		+		+	-		-	+	-		+
-	+	+	+		+		+		-	-	
+	+	-	-	+		+	+		-	+	
-			+	+	-		+	+	+	+	-
+	-	+	+		+		+		+	-	-
+	+	-	+		-		+		-	+	+
+	-	+	+		-		+		+	-	+
	+	+	-			+	+	+	+	-	-
			+	-	+	+	-	+	-	+	+
+		+	-	+		-	+	+	-		+
+		+	-	+		+	+	-	+		-
	+	+		+	+		-	+	-	-	+
+	-	-	-	+	+	+	+		-	+	
-			+	-	-	+	+	+	+	+	-
+	+	-		+	-		-	+	-	+	+
-			-	+	+	-	+	+	-	+	+
+	-	+		+	-		-	+	+	-	+
-	+	+	-	+	+	+	+		-	-	
+		+	+	-	-	+	-	+	-		+
-			-	+	+	+	+	-	+	+	-
+	-	+		+	+		-	+	-	-	+
-	+	+	+	+	-		+	+	+	-	-
+	+	-	-	+		-	+	+	-	+	+
+	-	-	+	+	-		+	+	+	+	-
	+	+	+	-	+	+	-	+	-	-	+
+	-	+	-	+	-	+	+	+	+	-	
+	+	-	-	+		+	+	-	+	+	-
+	-	+	-		+	+	+	+	+	-	-
-	+	+	-	+	+	-	+	+	-	-	+
+	+	-	+	-	-	+	-	+	-	+	+
-	+	+	+	-	-	+	+	+	+	-	-
+	-	-	-	+	+	-	+	+	-	+	+
-	+	+	-	+	+	+	+	-	+	-	-
+	-	+	+	-	-	+	-	+	+	-	+
+	-	-	+	-	-	+	+	+	+	+	-
+	-	+	-	+	-	-	+	+	+	-	+
+	-	-	-	+	+	+	+	-	+	+	-
+	-	+	+	-	+	+	-	+	-	-	+
+	-	+	-	+	-	+	+	-	+	-	+

Fig. 4.3.8. Final disjoint terms pertaining the terminal reliability between UCLA and CMU in the network of Fig. 4.3.6. [28].

TABLE 4.3.1. The upper and lower bounds given at different iterations of the algorithm [28].

Number of terms represented in the P polynomial form	p = .9 P = .912911 (Exact Value) Lower bound to P	Upper bound to P	p = .99 P = .999443 (Exact Value) Lower bound to P	Upper bound to P
18	.890595	.962134	.999382	.999638
24	.904461	.936912	.999427	.999535
30	.910147	.915520	.999443	.999445
36	.911304	.914429	.999443	.999444
42	.912640	.912912	.999443	.999443
48	.912776	.912911	.999443	.999443
53	.912911	———	.999443	———

In much more complicated example specified in Figs. 4.3. 9 through 11, the program did not find the exact probability expression after 10 min of computation time.

In this case Algorithm B (with T = 3 and T = 4) gave the approximate results shown in Fig. 4.3.12. The computing times were 99s for T = 3 and 546s for T = 4.

The numerical results (computed using double-precision arithmetic) for different arc reliabilities (0.9, 0.99, 0.999) and threshold T = 4 are shown in Table 4.3.2. Note that a meaningful bound is obtained for p = 0.999 and p = 0.99 while p = 0.9 gave an upper bound larger than 1.

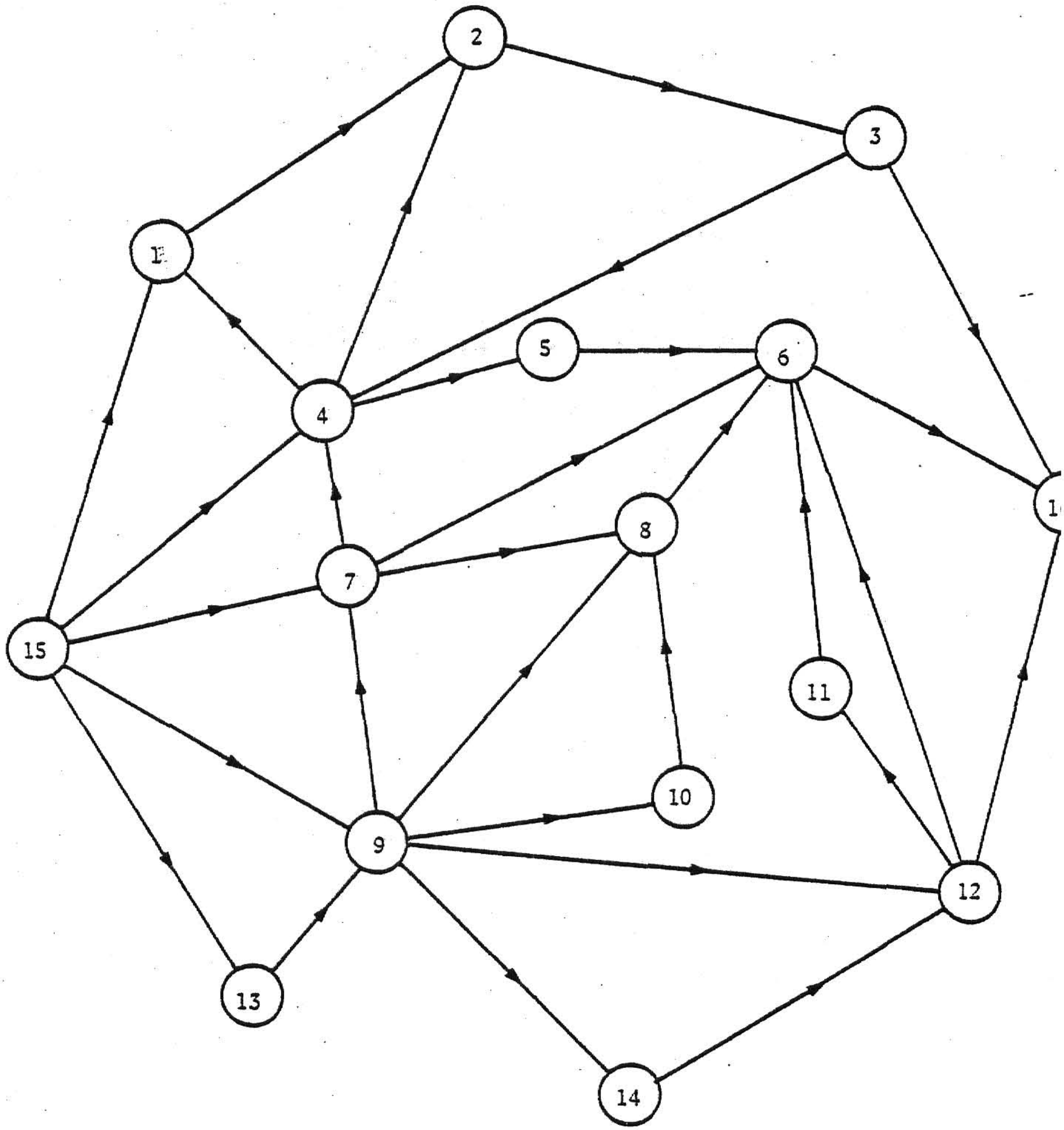


Fig. 4.3.9. Topology of a more connected network [28].

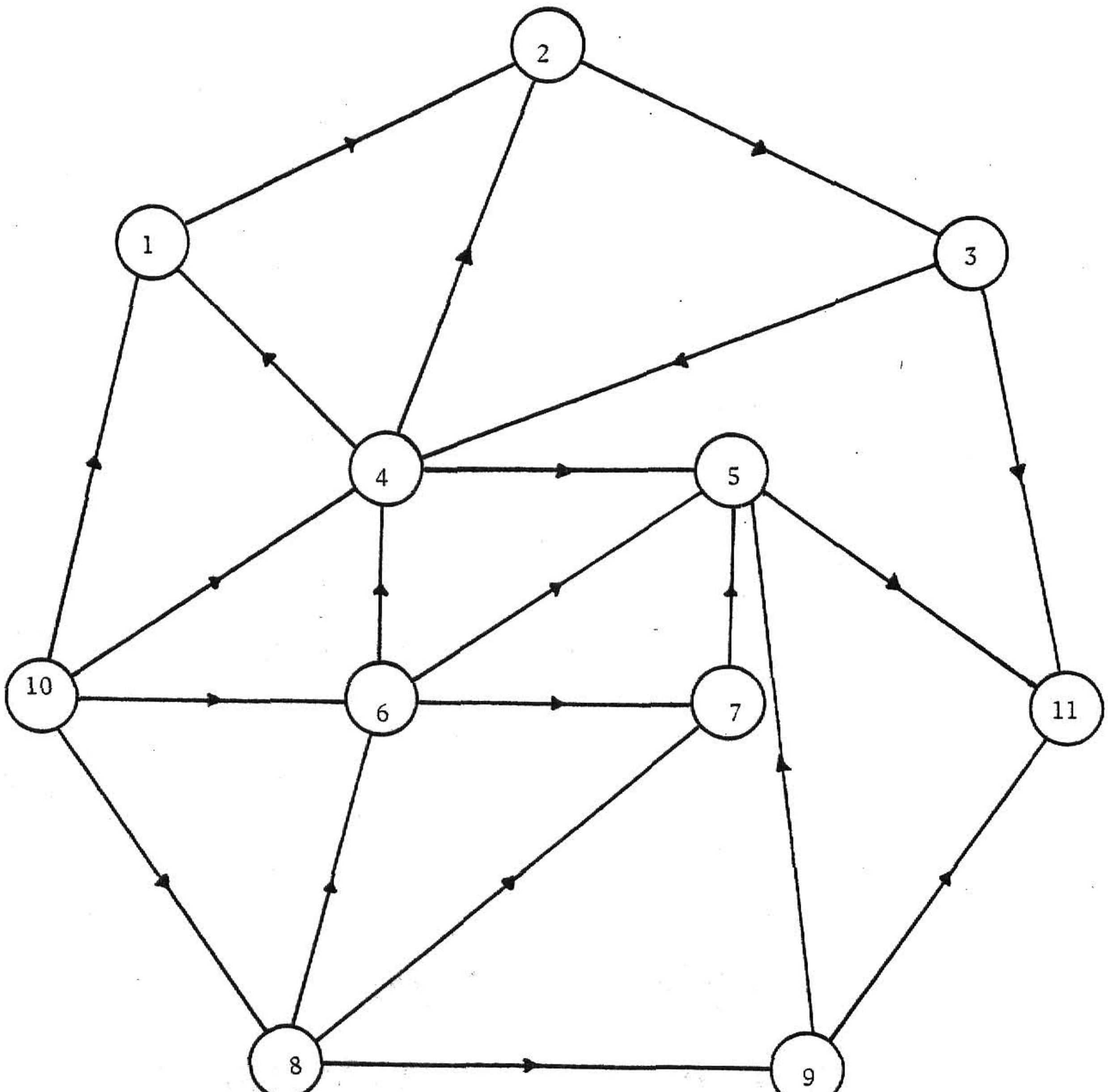


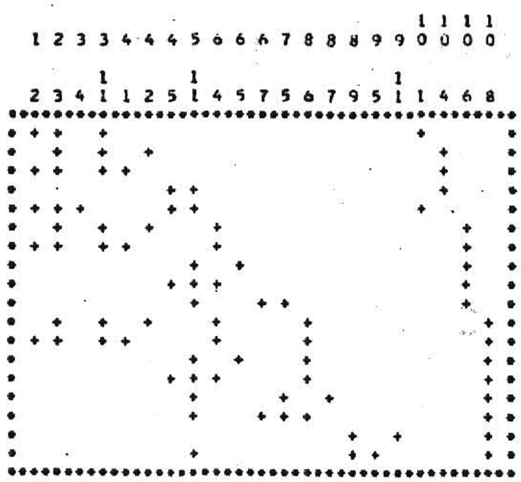
Fig. 4.3.10. Network of Fig. 4.3.9 after series-parallel reduction.

```
NODE REPRESENTATION OF PATHS FROM NODE. 10 TO NODE. 11

000000000000000000000000000000111   1   2   3
000000000000000000000000000001110   4   2   3
000000000000000000000000000001111   4   1   2   3
000000000000000000000000000011000   4   5
000000000000000000000000000011111   1   2   3   4   5
000000000000000000000000000101110   6   4   2   3
000000000000000000000000000101111   6   4   1   2   3
000000000000000000000000000110000   6   5
000000000000000000000000000111000   6   4   5
000000000000000000000000001110000   6   7   5
000000000000000000000000010101110   8   6   4   2   3
000000000000000000000000010101111   8   6   4   1   2   3
000000000000000000000000010110000   8   6   5
000000000000000000000000010111000   8   6   4   5
000000000000000000000000011010000   8   7   5
000000000000000000000000011110000   8   6   7   5
000000000000000000000000110000000   8   9
000000000000000000000000110010000   6   9   5
PATHS NUMBER =   18
```

(a)

ARC REPRESENTATION OF BEGIN PATHS.



(b)

Fig. 4.3.11. All simple paths represented as sets of arc between vertex 10 and vertex 11 in the network of Fig. 4.3. 10 [28].

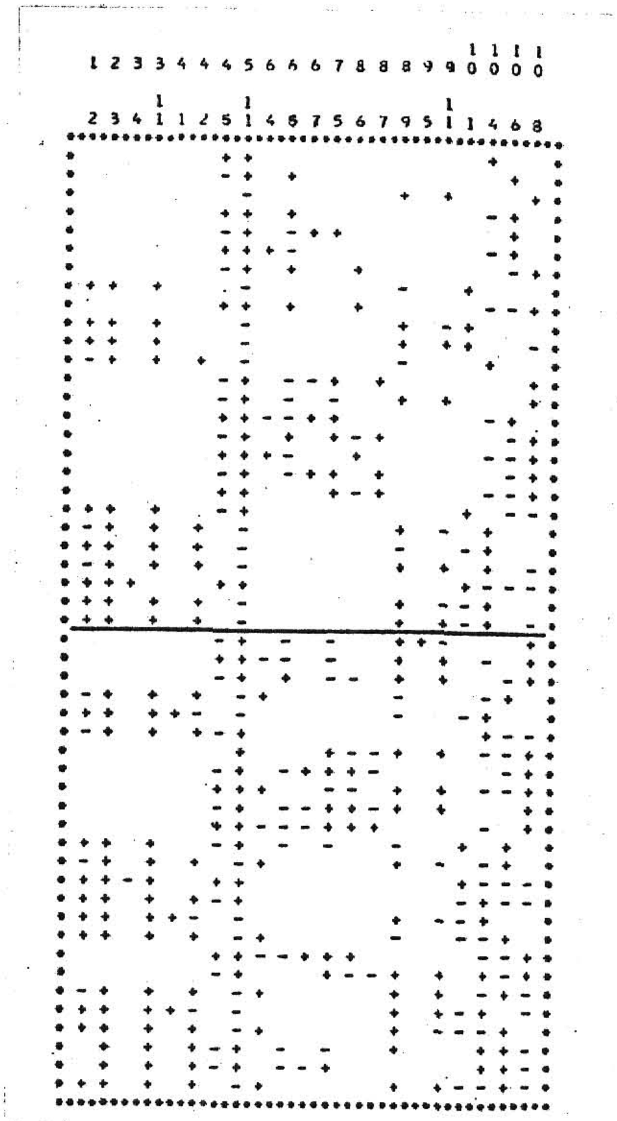


Fig. 4.3.12. Final disjoint terms pertaining the terminal reliability between vertex 10 and vertex 11 in the network of Fig. 4.3.10 computed assuming T = 4. The terms over the continuous line are those obtained by assuming T = 3 [28].

TABLE 4.3.2. The numerical results for different arc reliabilities and threshold T = 4 [28].

arc reliability	Number of terms represented in the P polynomial form	Lower bound to P	Number of terms represented in the h polynomial form	Numerical value of h	Numerical value of f	Upper bound to the error
.999	6	.9999959964	0	0	.448896 E-04	.448896 E-04
	12	.9999999850	597	.452096 E-12	.975947 E-07	.975951 E-07
	18	.9999999950	1361	.710370 E-12	.289591 E-07	.289598 E-07
	24	.9999999970	2902	.134437 E-11	.604517 E-08	.604651 E-08
	30	.9999999980	4747	.197565 E-11	.330525 E-10	.350281 E-10
	36	.9999999980	5917	.231420 E-11	.270374 E-10	.293516 E-10
	42	.9999999980	6520	.244582 E-11	.706301 E-11	.950883 E-11
	48	.9999999980	6864	.249059 E-11	.304888 E-11	.553947 E-11
	51	.9999999980	6932	.249478 E-11	0	.249478 E-11
.99	6	.9995971677	0	0	.43837186E-02	.43837186E-02
	12	.9999851723	597	.44219869E-07	.93838609E-04	.93882228E-04
	18	.9999948242	1361	.70122448E-07	.28524495E-04	.28594617E-04
	24	.9999968391	2902	.13316312E-06	.64278785E-05	.65610416E-05
	30	.9999978574	4747	.19615143E-06	.33263845E-06	.52878988E-06
	36	.9999978685	5917	.23022975E-06	.27161862E-06	.50184837E-06
	42	.9999978979	6520	.24380386E-06	.75209834E-07	.31901369E-06
	48	.9999979177	6864	.24886011E-06	.34170635E-07	.28303074E-06
	51	.9999979186	6932	.24942804E-06	0	.24942804E-06
.9	6	.9586421242	0	0	.3338110576	.3338110576
	12	.9869309623	597	.0033341457	.0619494660	.0652836117
	18	.9936697483	1361	.0056610330	.0235136810	.0291747140
	24	.9954711074	2902	.0109723135	.0087925397	.0197648532
	30	.9965185311	4747	.0163667574	.0028807640	.0192475214
	36	.9966672561	5917	.0193707570	.0023025090	.0216732660
	42	.9968873964	6520	.0207271040	.0008308054	.0215519094
	48	.9970443695	6864	.0214113369	.0004378014	.0218491383
	51	.9970893735	6932	.0215247944	0	.0215247944

Two algorithms were developed for determining symbolic expressions giving an exact and an approximate vlaue for the terminal reliability in a probabilistic network. These algorithms reduce the problem to the computation of a disjoint form of a Boolean function f, defined by the set of all simple paths between the initial and the final node.

The number of variables of this function is the number N of arcs in the network. Since each step of the algorithms computes a disjoint implicant of function f, the number of steps is bounded by the number m of fundamental products of f. Actually, these algorithms trade off the number of steps with the complexity of the single step by applying the concepts of Boolean algebra which is largely used in many fields (like switching and coding theory) for dealing with this kind of problem.

It is pointed out that the reduction of the number of steps is important even without regard to the complexity; in fact, the number of steps is equal to the number of terms in the final symbolic expression, and thus a smaller number of steps implies a more concise form of the terminal reliability. Of course this is particularly important when this form must be evaluated for a number of different sets of values for the arc reliabilities.

4.4 Efficient Evaluation of System Reliability by Monte Carlo Method

1. Introduction

It presents a new Monte Carlo method for calculating the reliability of a large complex system represented by a reliability block diagram or by a fault tree. The usual term-wise calculation (cut or path sets) becomes impractical for large systems since the reliability involves a large number of terms.

Although several approximations have been proposed, they yield only lower and upper bounds of the reliability.

When the crude (straight forward) Monte Carlo method is used, a large number of trials is required to obtain reasonably precise estimates of the reliability.

Mazumdar [81] proposed a Monte Carlo method with variance-reducing techniques [74] in order to decrease the variance of the Monte Carlo estimates of the reliability. There is, however, no guarantee that the method always reduces the variance.

In here, a better Monte Carlo method is obtained by applying variance-reducing techniques.

2. Statement of the Problem

Assumptions:

1) The system has k components, numbered 1,...,k.
2) Each component is either functioning or failed.
3) States of components are s-independent.
4) The system is either functioning or failed. The system is s-coherent.
5) Some path & cut sets are known.

Notation

x_i component state (r.v.)

$$\equiv \begin{cases} 1, \text{ if component } i \text{ is functioning,} \\ 0, \text{ otherwise.} \end{cases} \tag{2.1}$$

x $(x_1,\ldots,x_k)$ is a component state vector.

b $(b_1,\ldots,b_k)$ is a sample vector of x.

$\phi(x)$ s-coherent function of x

$$\equiv \begin{cases} 1, \text{ if system is functioning,} \\ 0, \text{ if system is failed.} \end{cases} \tag{2.2}$$

N sample size.

The elimination of trivial components gives the inequality,

$$0 \quad \Pr\{x = b\} = \prod_{i=1}^{k} \Pr\{x_i = b_i\} < 1 \tag{2.3}$$

The problem is to calculate the system reliability,

$$R \equiv \Pr\{\phi(x) = 1\} \tag{2.4}$$

$$= \Sigma_b \ \phi(b)\Pr\{x = b\} \tag{2.5}$$

$$= E_x \ \{\phi(x)\} \tag{2.6}$$

Crude Monte Carlo Method

Generate N s-independent samples $C_1,\ldots, C_N$ of x. Evaluate R by the s-unbiased binomial estimator $\hat{R}c$ (the subscript c stands for "crude"),

$$\hat{R}c \equiv N^{-1} \sum_{\nu=1}^{N} \phi(C_\nu) \tag{2.7}$$

$$\text{Var } \{\hat{R}c\} = N^{-1}R(1 - R) \tag{2.8}$$

3. New Monte Carlo Method

Let ϕ_L and ϕ_u be two binary functions satisfying (3.1) and (3.2).

$$\phi_L(b) \leq \phi(b) \leq \phi_U(b), \text{ for all b.} \tag{3.1}$$

$$\phi_L(b) \not\equiv 0, \quad \phi_U(b) \equiv 1 \tag{3.2}$$

For any given i, $0 \leq i \leq k$ and $(b_1,\ldots,b_i)$ define

$$R_{L,i}(b_1,\ldots,b_i) \equiv \sum_{b_{i+1},\ldots,b_k} \phi_L(b)\Pr\{x = b\}, \tag{3.3}$$

$$R_{U,i}(b_1,\ldots,b_i) \quad \sum_{b_{i+1},\ldots,b_k} \phi_U(b)\Pr\{x = b\}. \tag{3.4}$$

$R_{L,i}$ and $R_{u,i}$ are used to generate random samples in the new Monte Carlo method. ϕ_L, ϕ_u, and $R_{L,i}$ $R_{u,i}$ can be obtained by the method given in Section 4.

$R_{L,0}$ and $R_{U,0}$ are the reliabilities of the system represented by ϕ_L and ϕ_U, respectively, and are abbreviated respectively by R_L and R_U. The following inequalities hold:

$$0 < R_L \leq R \leq R_U < 1. \tag{3.5}$$

If the equality $R_U = R_L$ holds, then $R = R_L = R_U$ and the problem is trivial; R can be obtained without the use of the Monte Calro methods. In the discussion that follows we assume the inequality

$$R_U - R_L > 0 \tag{3.6}$$

Apply the straight-forward control variate method [74] to (2.5); we have

$$R = \sum_b [\phi(b) - \phi_L(b)]\Pr\{x = b\} + \sum_b \phi_L(b)\Pr\{x = b\} \tag{3.7}$$

$$= \sum_b [\phi(b) - \phi_L(b)]\Pr\{x = b\} + R_L \tag{3.8}$$

We now consider generating the random samples with probability different from $\Pr\{x = b\}$ according to the importance sampling method [74].

Define the sets

$$X \equiv \{b \mid \phi(b) - \phi_L(b) = 1\} \tag{3.9}$$

$$Y \equiv \{b \mid \phi_U(b) - \phi_L(b) = 1\} \tag{3.10}$$

Using (3.6) and since $X \varepsilon Y$, we rewrite (3.8) as follows:

$$R = \sum_{b\varepsilon X} [\phi(b) - \phi_L(b)]\Pr\{x = b\} + R_L$$

$$= [R_U - R_L] \sum_{b\varepsilon Y} [\phi(b) - \phi_L(b)]\Pr\{y = b\} + R_L\ , \tag{3.11}$$

where $y \equiv (y_1,\ldots,y_k) \varepsilon\ Y$ is a random vector and

$$\Pr\{y = b\} \equiv \Pr\{x = b\}/[R_U - R_L] \tag{3.12}$$

Since $\phi_L(b) \equiv 0$ for all $b\varepsilon Y$, we rewrite (3.11) as follows:

$$R = [R_U - R_L] \sum_{b\varepsilon Y} \phi(b)\Pr\{y = b\} + R_L$$

$$= [R_U - R_L] \cdot E_y\{\phi(y)\} + R_L \tag{3.13}$$

The new Monte Carlo method is obtained from (3.13). Generate N s-independent samples $S_1,\ldots,S_N$ of y. Evaluate R by the s-unbiased binomial estimator $\hat{R}_N$ (the subscript N stands for "New"):

$$\hat{R}_N \equiv N^{-1}[R_U - R_L] \sum_{\nu=1}^{N} \phi(s_\nu) + R_L \tag{3.14}$$

$S_1,\ldots,S_N$ can be generated easily by the method given in Note 1.

The following theorem shows that the new Monte Carlo method estimates the reliability with a smaller variance than the crude Monte Carlo method does.

Theorem: Let ϕ_L and ϕ_U satisfy (3.1) and (3.2). Assume $R_U - R_L > 0$ as in (3.6). Define $\hat{R}_N$ as is (3.14). Then,

$$\mathrm{Var}\{\hat{R}_N\} = N^{-1}(R_U - R)(R - R_L) \tag{3.15}$$

$$< \mathrm{Var}\{\hat{R}_C\} = N^{-1}R(1 - R) \tag{3.16}$$

This theorem is proved in Note 2.

4. $(\phi_L, R_{L,i})$ AND $(\phi_U, R_{U,i})$

4.1 $(\phi_L, R_{L,i})$

Take some, say m, path sets $P_1, \ldots, P_m$ of the s-coherent structure ϕ. Define

$$\phi_L(b) \equiv 1 - \prod_{j=1}^{m} [1 - \prod_{i\varepsilon P_j} b_i] \tag{4.1}$$

As shown in Note 3, $\phi_L(b)$ satisfies (3.1) and (3.2). Obtain a reliability function h_L of ϕ_L by the method of Note 3. The value of $R_{L,i}$ of (3.3) can be calculated by

$$R_{L,i}(b_1,\ldots,b_i)$$

$$= h_L(b_1,\ldots b_i\ ,Pr\{x_{i+1} = 1\},\ldots,Pr\{x_k = 1\})\ x$$

$$\prod_{\ell=1}^{i} Pr\{x_\ell = b_\ell\} \tag{4.2}$$

Note 3 refers to the more compact type ϕ_L and h_L.

4.2 $(\phi_U,\ R_{U,i})$

Take some, say n, cut sets $K_1,\ldots,K_n$ of ϕ .

Define

$$\phi_U(b) \equiv \prod_{j=1}^{n} [1 - \prod_{j\varepsilon K_j} (1 - b_i)] \tag{4.3}$$

In the same way as in 4.1, we see that ϕ_U satisfies (3.1) and (3.2)

Obtain a reliability function h_U of ϕ_U by using (C.1). The value of $R_{U,i}$ of (3.4) can be calculated by

$R_{U,i}(b_1,\ldots,b_i)$

$$= h_U(b_1,\ldots,b_i,\Pr\{x_{i+1} = 1\},\ldots,\Pr\{x_k = 1\}) \times \prod_{\ell=1}^{i} \Pr\{x_\ell = b\} \quad . \tag{4.4}$$

5. Numerical Example

The system is represented by the reliability block diagram in Fig. 4.4.1. The reliabilities of the components are $\Pr\{x_i = 1\} = 0.9$ for $i = 1,\ldots,18$.

Let us take the path sets (m=6)

$P_1 = \{2,15\}$, $P_2 = \{3, 16\}$, $P_3 = \{4, 17\}$

$P_4 = \{1, 6, 14\}$, $P_5 = \{5, 11, 18\}$, $P_6 = \{1, 7, 15\}$,

and the cut sets (n=1)

$K_1 = \{1, 2, 3, 4, 5\}$.

then

$\phi_U(b)$

$$= h_U(b) = 1 - (1 - b_1)(1 - b_2)(1 - b_3)(1 - b_4)(1 - b_5). \tag{5.1}$$

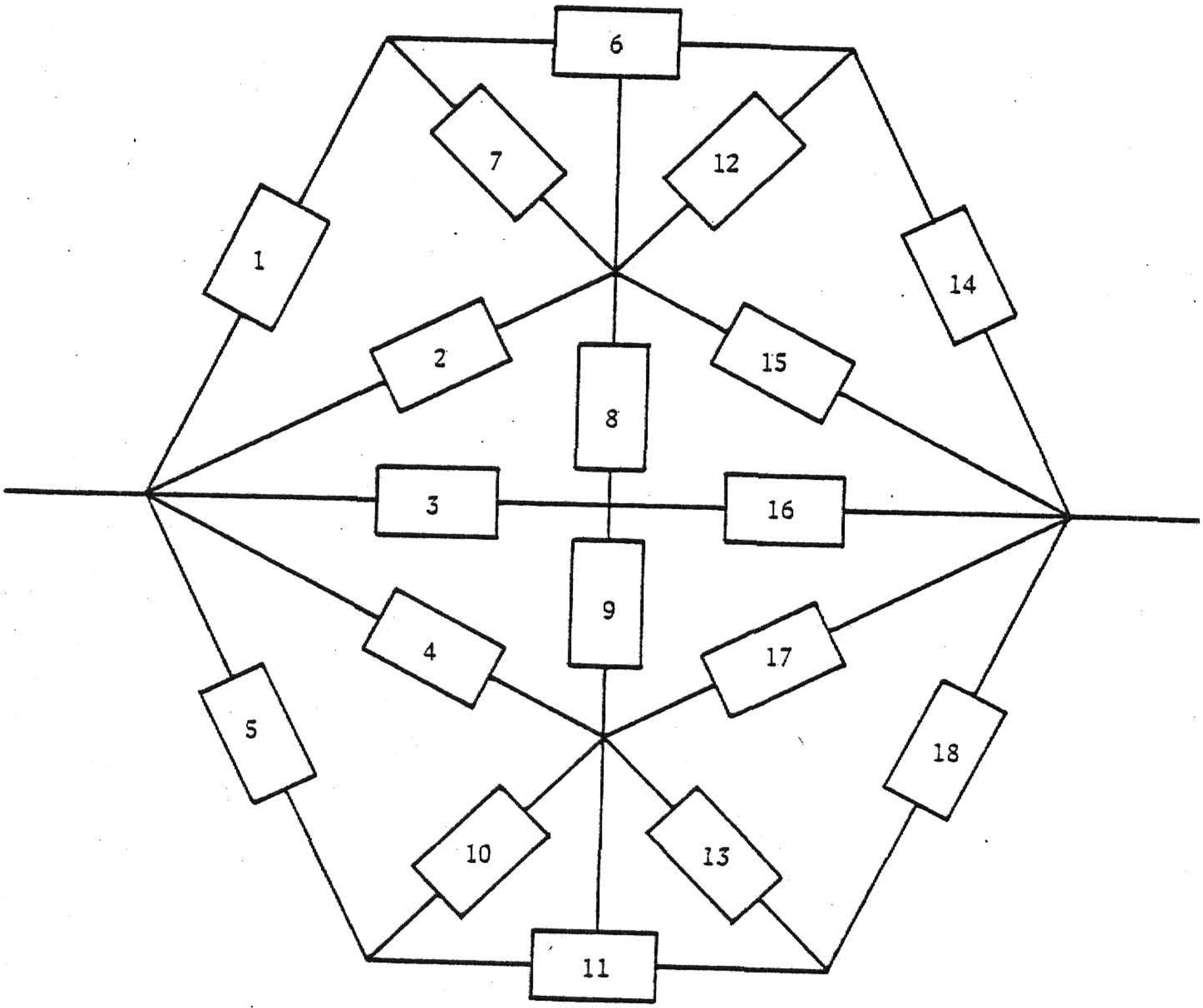


Fig. 4.4.1. A hypothetical 18-component S-coherent complex system [78].

The identity of (C.1) yields

$$\phi_L(b) = h_L(b) = 1 - (\ 1\ b_3 b_{16})(1 - b_4 b_{17})(1 - b_5 b_{11} b_{18})$$

$$\times \Big[b_1(1 - b_6 b_{14})[b_{15}(1 - b_2)(1 - b_7) + (1 - b_{15})]$$

$$+ (1 - b_1)(1 - b_2 b_{15})\Big] \quad . \tag{5.2}$$

The results are clearer in terms of the system failure probability. From (5.1) and (5.2), we obtain

$$1 - R_L = 368 \times 10^{-6}, \tag{5.3}$$

$$1 - R_U = 10 \times 10^{-6}\ . \tag{5.4}$$

The inequality of (3.5) ensures that the system failure probability $1 - R$ lies in the interval $[10 \times 10^{-6}, 368 \times 10^{-6}]$. The reliability R is the sum of $2^{18} = 262144$ terms. The termwise calculation gives the exact system failure probability

$$1 - R = 29.1 \times 10^{-6}. \tag{5.5}$$

We see from (2.8) and (3.15) that the estimators $1 - \hat{R}_c$ and $1 - \hat{R}_N$ with $N = 3000$ have standard deviations of 98.5×10^{-6} and 1.47×10^{-6}, respectively.

The standard deviation of $1 - \hat{R}_N$ is much less than the standard deviation of $1 - \hat{R}_c$, and also much less than 358×10^{-6}, the length of the interval $[10 \times 10^{-6}, 368 \times 10^{-6}]$ of the failure probability obtained from (3.5). Fig. 4.4.2 shows the results of the new and crude Monte Carlo methods. For $N = 3000$, we have

$$1 - \hat{R}_N = 27.9 \times 10^{-6},\ 1 - \hat{R}_C = 0.0 \tag{5.6}$$

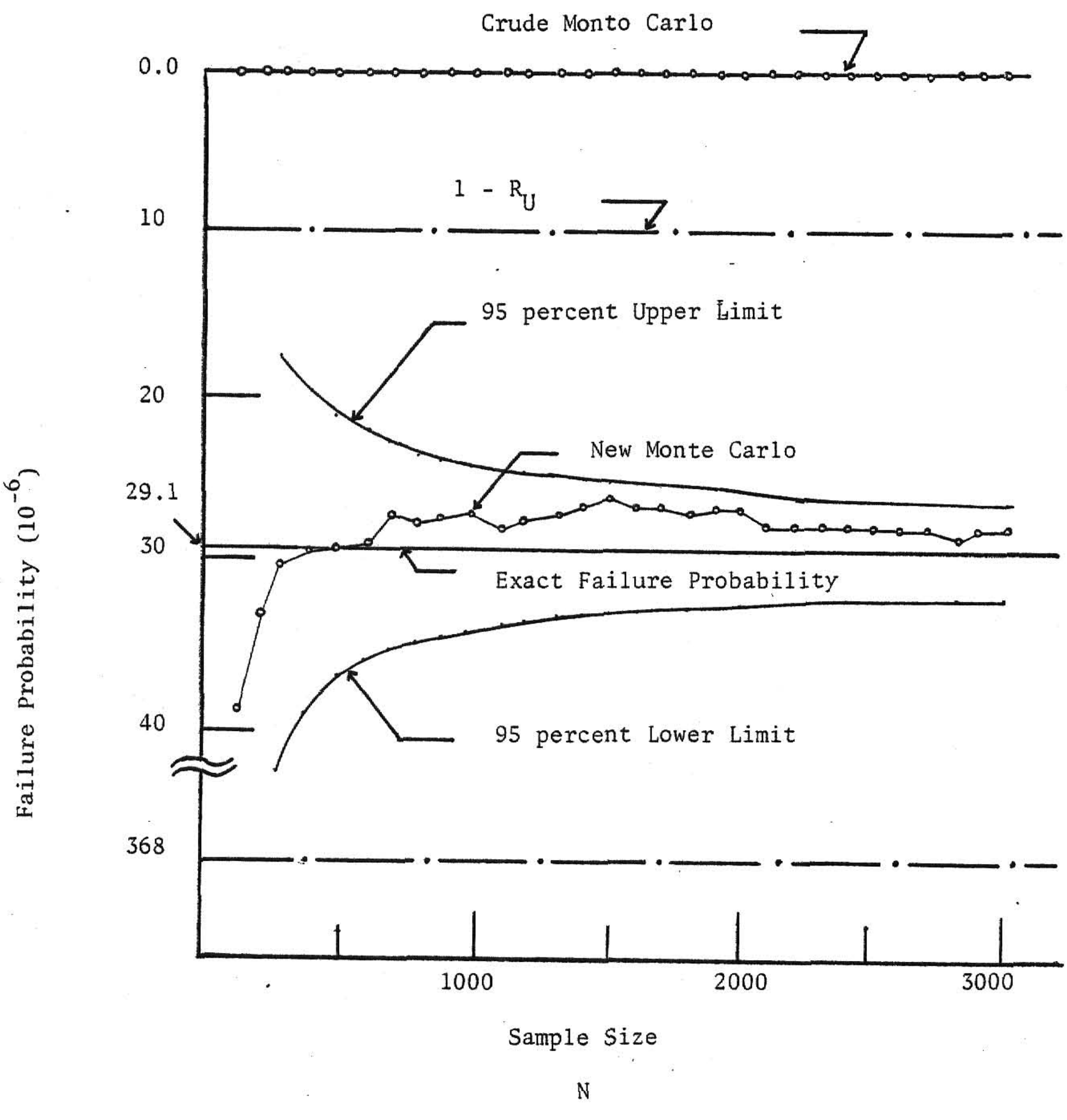


Fig. 4.4.2. Result of New and Crude Monte Carlo Methods [78].

$\hat{R}_c$ gave the zero failure probability since all the samples $\phi(C_\nu)$, $\nu = 1,\ldots,$ 3000, became 1 (unity).

Figure 2 includes the 95 percent upper and lower s-confidence limit of $1 - \hat{R}_s$ for different values of N.

6. Conclusions

It is proved that the new Monte Carlo method gives a reliability estimate with a smaller variance than that of the crude Monte Carlo method.

Note 1: Generation of Samples $S_i,\ldots,S_N$.

$\Pr\{y = b\}$ is represented as

$$\Pr\{y = b\} \equiv \Pr\{y_1 = b_1\} \Pr\{y_2 = b_2 | y_1 = b_1\} \times \ldots$$
$$\times \Pr\{y_k = b_k | y_1 = b_1,\ldots,y_{k-1} = b_{k-1}\} . \tag{A.1}$$

This identity shows that the generation of $s_\nu \equiv (s_{1,\nu},\ldots,s_{i,\nu},\ldots,s_{k,\nu})$ is reduced to the sequential generation of $s_{1,\nu},\ldots,s_{i,\nu},\ldots,s_{k,\nu}$ with the probabilities $\Pr\{y_i | y_1 = s_{1,\nu},\ldots,y_{i-1} = s_{i-1,\nu}\}$, $i = 1,\ldots,k$, respectively.

Denote $(y_1,\ldots,y_{i-1})$ and $(s_{1,\nu},\ldots,s_{i-1,\nu})$ by Y_{i-1} and $S_{i-1,\nu}$, respectively. Assume that the first (i - 1) elements $S_{i-1,\nu}$ have already been generated. Assume further that we can calculate the value of $\Pr\{y_i | Y_{i-1} = S_{i-1,\nu}\}$ for $y_i = 1$ and 0. Then, element $i, s_{i,\nu}$, can be generated by using a random number with rectangular distribution between 0 and 1 [74].

We give now the method of the calculation of $\Pr\{y_i | Y_{i-1} = S_{i-1,\nu}\}$. The following identity holds:

$$\Pr\{y_i | Y_{i-1} = S_{i-1,\nu}\} \equiv \frac{\Pr\{Y_{i-1} = S_{i-1,\nu}, y_i\}}{\Pr\{Y_{i-1} = S_{i-1,\nu}\}} \tag{A.2}$$

Therefore, using (3.3), (3.4), and (3.12), we obtain

$$\Pr\{y_i | Y_{i-1} = S_{i-1,\nu}\} = \frac{R_{U,i}(S_{i-1,\nu}, y_i) - R_{L,i}(S_{i-1,\nu}, y_i)}{R_{U,i-1}(S_{i-1,\nu}) - R_{L,i-1}(S_{i-1,\nu})} \ . \tag{A.3}$$

The probability $\Pr\{y_i | Y_{i-1} = S_{i-1\ \nu}\}$ can be calculated easily by (A.3), (4.2) and (4.4).

Note 2: Proof of Theorem

$$\mathrm{Var}\{\hat{R}_N\} = \mathrm{Var}\{\hat{R}_N - R_L\} \tag{B.1}$$

$$= \mathrm{Var}\left\{N^{-1} \sum_{\nu=1}^{N} (R_U - R_L)\phi(s\ )\right\} \ . \tag{B.2}$$

(B.2) is the variance of the arithmetical mean of N i.i.d. samples of $(R_U - R_L)\phi(y)$. Hence, we have

$$\mathrm{Var}\{\hat{R}_N\} = N^{-1}\ \mathrm{Var}\{R_U - R_L)\phi(y)\} \tag{B.3}$$

$$= N^{-1}E_y\{(R_U - R_L)^2\phi^2(y)\}$$

$$- N^{-1}(E_y\{R_U - R_L)\phi(y)\})^2 \ . \tag{B.4}$$

Since the value of $\phi(b)$ is either 0 or 1, the identity,

$$\phi^2(b) \equiv \phi(b) \tag{B.5}$$

holds. Using (4.13) and (B.5), we have

$$E_y\{(R_U - R_L)\phi(y)\} = R - R_L \tag{B.6}$$

$$E_y\{(R_U - R_L)^2\phi^2(y)\} = (R_U - R_L)\ (R - R_L) \tag{B.7}$$

The substitution of (B.6) and (B.7) into (B.4) yields (3.15).

The variances of (2.8) and (3.15) give

$$\mathrm{Var}\{\hat{R}_C\} - \mathrm{Var}\{\hat{R}_N\} = N^{-1}[R(1 - R_U) + R_L(R_U - R)]\ , \tag{B.8}$$

yielding (3.16) by (3.5).

Note 3 : ϕ_L and h_L .

Suppose $\phi_L(b) = 1$. Then, from (4.1), at least one path set among $P_1, \ldots, P_m$ is functioning. Hence $\emptyset(b) = 1$, resulting in (3.1). (3.2) is satisfied by (4.1).

The following identity holds for any binary function $\zeta(b)$ [71].

$$\zeta(b) \equiv b_i\zeta(1_i, b) + (1 - b_i)\zeta(0_i, b), \tag{C.1}$$

where the symbol $(\cdot_i, b)$ denotes $(b_1, \ldots, b_{i-1}, \cdot, b_{i+1}, \ldots, b_k)$.

From repeated applications of (C.1), we can rewrite (4.1), obtaining the reliability function h_L:

$$\phi_L(b) \equiv 1 - \sum_{j=1}^{\tilde{m}} L_j(b) \equiv h_L(b), \tag{C.2}$$

where each argument b_i appears at most once in each polynomial $L_j(b)$.

Example 1 Suppose $m = 3$, $P_1 = \{1,2,3\}$, $P_2 = \{2,3,4\}$, and $P_3 = \{3,5,6\}$. Using (C.1) we have

$$\phi_L(b) = 1 - (1 - b_1b_2b_3)\ (1 - b_2b_3b_4)\ (1 - b_3b_5b_6) \tag{C.3}$$

$$= 1 - b_3(1 - b_1b_2)\ (1 - b_2b_4)\ (1 - b_5b_6) - (1 - b_3) \tag{C.4}$$

$$= 1 - b_2b_3(1 - b_1)\ (1 - b_4)\ (1 - b_5b_6)$$

$$- (1 - b_2)\ b_3\ (1 - b_5b_6) - (1 - b_3) \tag{C.5}$$

$$= h_L(b)\ . \tag{C.6}$$

If necessary, we can obtain more compact type ϕ_L and h_L by replacing some factors by 1 (unity). This is illustrated by the following example.

Example 2 If we replace $(1 - b_2b_4)$ by 1 in (C.4), then we have

$$\phi_L(b) \geq 1 - b_3(1 - b_1b_2)\ (1 - b_5b_6) - (1 - b_3) \tag{C.7}$$

We can use the r.h.s. of (C.7) as compact type ϕ_L and h_L because of the inequality of (C.7).

CHAPTER 5 SELECTION OF A SUITABLE RELIABILITY EVALUATION TECHNIQUE BY MULTIPLE ATTRIBUTE DECISION MAKING METHOD

1. Introduction

The problem of a reliability analysis of a complex system has generally been of considerable interest. Although there are several methods which can be used in evaluating the complex system reliability, it is not easy to analyze relevant alternative methods for their capability. The multiple attribute decision making (MADM) methods are for evaluating and selecting a desired alternative from a small, explicit list of alternatives [89].

The evaluation and selection of a system reliability evaluation technique is a multiple criteria decision making process. The criteria are often in conflict with each other, and the measurements of their attributes are nonhomogeneous. That is, such important criteria as total computational time, accuracy of the solution, ease of use of the method, ease of understanding the logic of the method, etc., arise.

The purpose of this paper is to demonstrate the decision making process through the applications of the MADM methods in the selection of a suitable system reliability evaluation technique for the corresponding system configuration.

2. Formulation of the Problem

All problem solving must begin with establishing goals (objectives) [90]. And the objectives should be nonconflicting, coherent, and logical as a set. One way this may be achieved is to hierarchically derive the goals from some supergoal. We will follow the hierarchical goal structure approach [88, 89, 94]

Essentially we are trying to depict the worth relationships between overall performance objectives and successively lower levels of increasingly more specific performance attributes relevant to the selection of a specified alternative designed to achieve a given policy.

To render the discussion concrete, let us take a general evaluation technique in the system reliability problem. A system reliability evaluation technique may be evaluated through the following measures of performance:

1) CPU - time on a computer
2) Accuracy of the solution with respect to the error in the component values and system reliability expression.
3) Ease of understanding the logic of the technique
4) Ease of use of the technique

After generating a hierarchical tree of overall objective and the attribute a next logical step is to select a unit of measurement of physical performance for each lowest-level attribute on the tree. Selecting physical performance measures must be done judgmentally.

Consider a communication system with five nodes and seven branches (two of these being interconnecting) shown in Fig. 5.1. It is assumed that all the nodes are perfectly reliable. There are nine feasible methods which can be employed to evaluate the reliability of this general network. These alternative methods are as follows:

A_1: Exhaustive Search

A_2: Direct Canonical Expansion

A_3: Probability Calculus

A_4: Bayes' Theorem

A_5: Flow Graph Method

A_6: Algebraic Extraction

A_7: Fast Algorithm

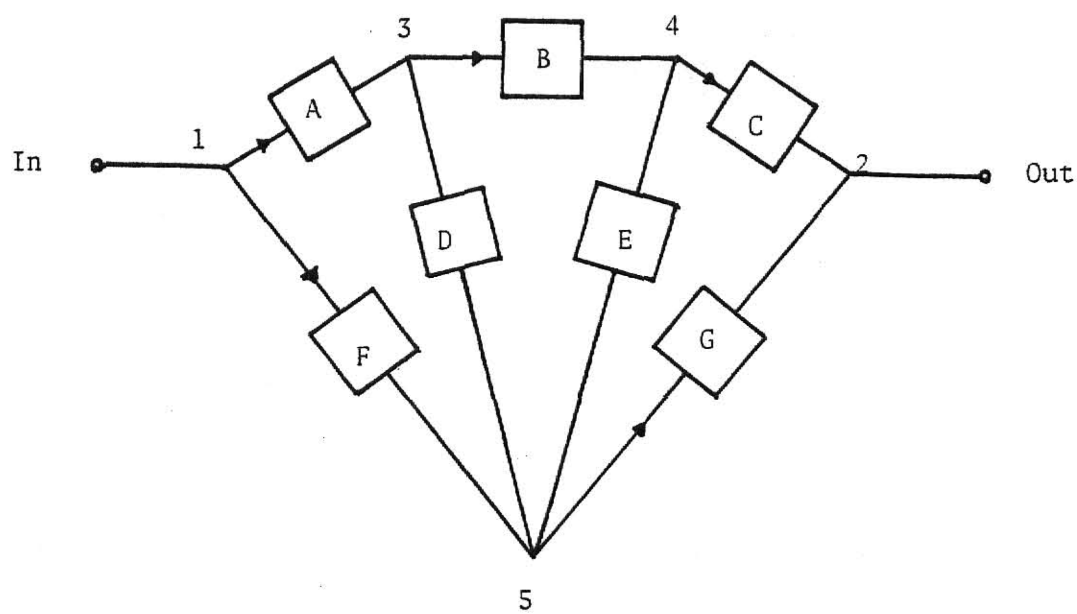


Fig. 5.1. A General network.

A_8: An Efficient Method for a General Network

A_9: Symbolic Reliability Evaluation Using Logical Signal Relations

As mentioned above, the alternatives will be examined by applying four major criteria. All the information of nine methods available [3, 5, 6, 7, 8, 55] are summarized in Table 5.1.

Note 1: Let m be the number of multiplications required

Let a be the number of additions required

Let T_m be the time for one multiplication on computer

Let T_a be the time for one addition on computer

Let T_c be the total computational time

In a typical Digital computer [95], T_m is about ten times more than T_a. (In IBM 1620, T_m = 12,512 μsec, while T_a = 1,200 μsec). Using $T_m = 10\ T_a$, total computational time T_c will be (10m + a) T_a. That is, T_c/T_a is equal to (10m + a).

In case, m = 354 and a = 58, $T_c = 3598\ T_a$

Note 2: The error analysis of any complex mathematical expression is based on the following theorems [95].

a) The absolute error in the summation (subtraction) of certain terms is equal to the summation of the absolute error in these terms

b) The relative error in the multiplication of certain terms is equal to the summation of the relative error in these terms.

In the analysis, it is assumed that all component reliability values have an absolute error of Δ_e. All unreliability values (q's) are also therefore having an absolute error of Δ_e. For simplification, all reliability (or unreliability) values may be assumed equal to p (or q). It is further assumed that the value of the error Δ_e is quite small.

TABLE 5.1. The System Reliability Evaluation Method Selection Problem

Attributes X_j	Alternatives A_i								
	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9
1. Total Computational time X_1	3598[1]	3598	1188	1188	1188	398	398	398	623
2. Accuracy, X_2 $\frac{\Delta R}{\Delta e}$	14	14	178	178	178	4	4	4[2]	9
3. Easy to understand the logic X_3(very easy → very difficult)	very easy	easy	easy	average	difficult	average	difficult	difficult	very difficult
4. Easy to use X_4(very easy → very difficult)	difficult	average	average	easy	difficult	average	easy	average	difficult

*1: Refer to NOTE 1

2: Refer to NOTE 2

Using above theorems for reliability expression by A_6 and A_7;

$$\Delta_R = \Delta_e \,[(2/p)p^2 + 2(3/p + 1/q)p^3q + 2(4/p + 2/q)p^4q^2 + (3/p + 2/q)p^3q^2 + 3(4/p + 3/q)p^4q^3]$$

or

$$\Delta_R = \Delta_e \,(2p + 2p^3 + 6p^2q + 8p^3q^2 + 4p^4q + 3p^2q^2 + 2p^3q + 12p^3q^3 + 9p^4q^2)$$

In the high reliability region ($p \simeq 1$); this error is approximately given as:

$$\Delta_R \simeq 4\Delta_e$$

where, Δ_R Absolute error in system reliability

Δ_e Absolute error in component reliability

3. Application of MADM methods

As to which method(s) we should use, the selection of MADM method(s) itself is a MADM problem. We suggest a general choice rule represented by a tree diagram in Fig. 5.2(a)[89]. Through those various situational judgments, some MADM methods--conjunctive constraints, simple additive weighting, linear assignment method, ELECTRE, and TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) are utilized for the solution of the reliability evaluation technique selection problem. This problem has nine alternatives with four attributes. First some transformations of qualitative (fuzzy) attributes to quantitative ones are given in order to use the method of simple additive weighting, ELECTRE, and TOPSIS, then solutions by several methods are illustrated.

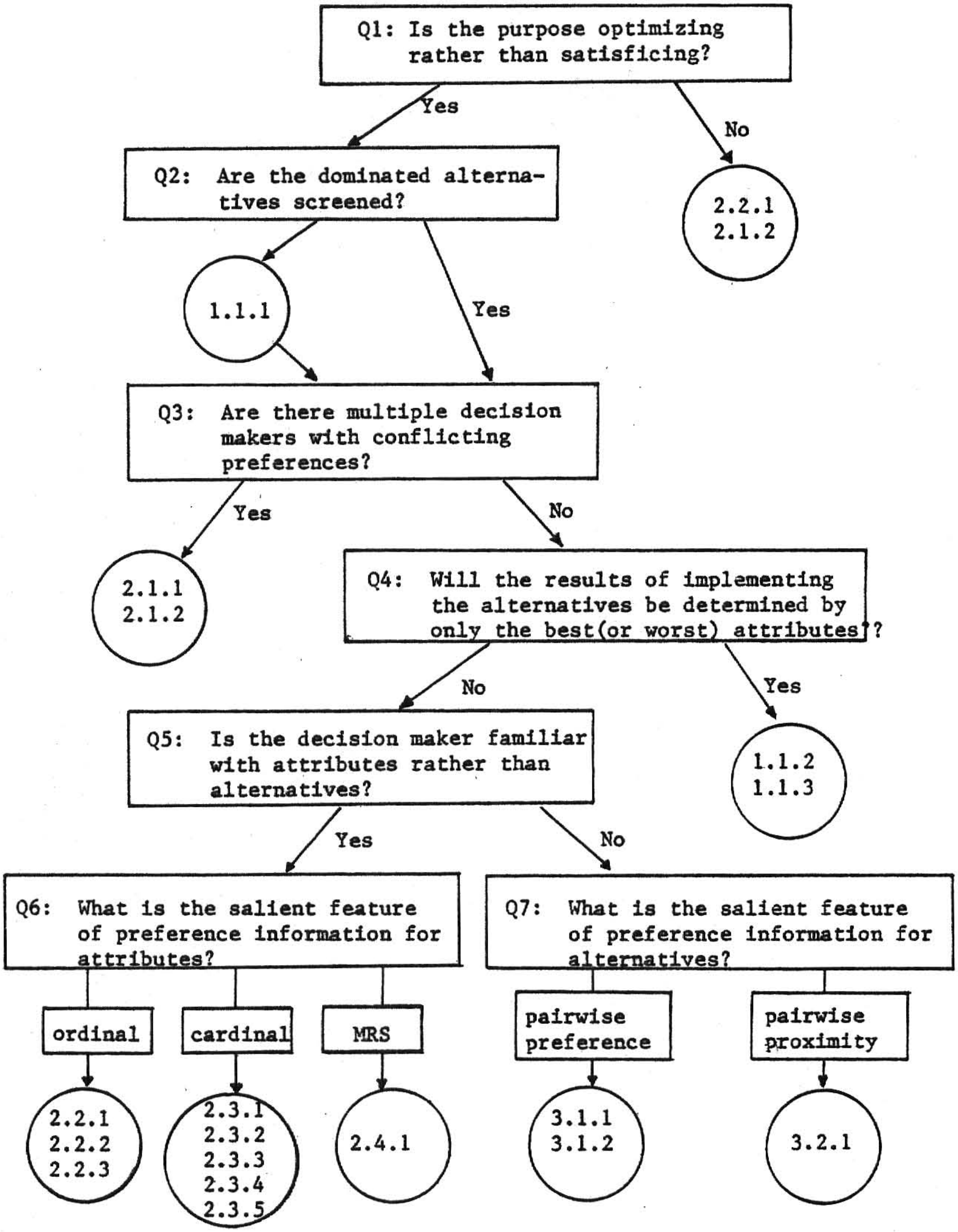


Fig. 5.2(a). MADM method specification chart [89].

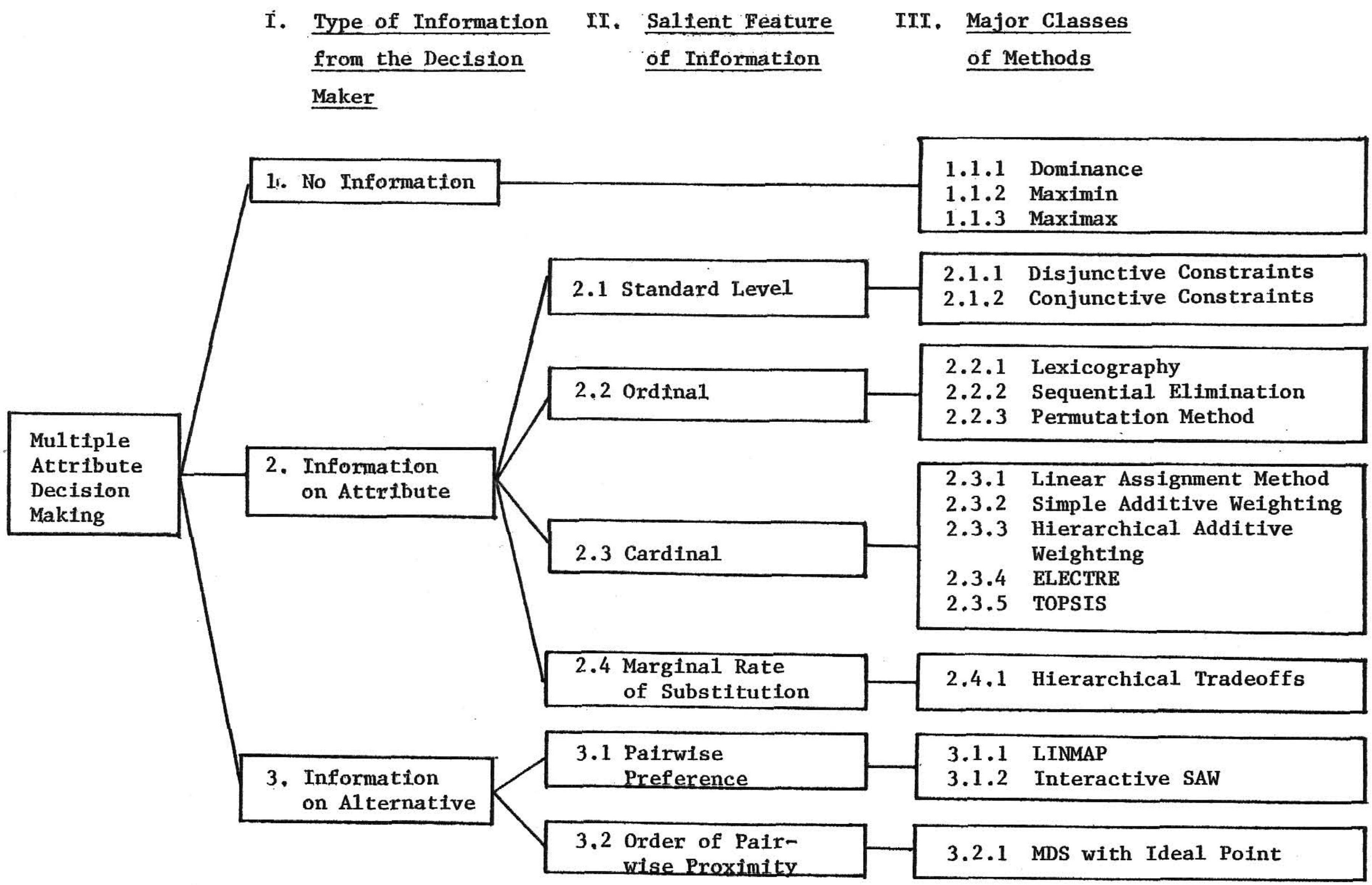


Fig. 5.2(b). A taxonomy of methods for multiple attribute decision making [89].

3.1 Transformation of Attributes

Qualitative attributes are quantified using the bipolar scale as shown in Fig. 5.3. Some qualitative attributes such as 'ease of use' or 'ease of understanding' need empirical judgments.

Next the quantitative attributes with different units are converted into comparable scale. A simpler procedure is to divide the outcome of a certain criterion by its maximum value, provided that the criteria are defined as benefit criteria (the larger X_j, the greater preference); then the transformed outcome of X_{ij} is,

$$r_{ij} = \frac{X_{ij}}{X_j^*}$$

$$\text{where } X_j^* = \max_i X_{ij} \tag{1}$$

Since all the qualitative attributes (X_3, X_4) have been transformed into the benefit scale, cost attributes (X_1, X_2) are also changed into benefit scale using

$$r_{ij} = \frac{1/X_{ij}}{\max_i (1/X_{ij})} = \frac{\min_i X_{ij}}{X_{ij}} = \frac{X_j^{\min}}{X_{ij}} \tag{2}$$

The comparable numerical decision matrix becomes

$$D = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \\ A_9 \end{array} \begin{array}{c} \begin{array}{cccc} X_1 & X_2 & X_3 & X_4 \end{array} \\ \begin{bmatrix} 0.11 & 0.29 & 0.9 & 0.3 \\ 0.11 & 0.29 & 0.7 & 0.5 \\ 0.34 & 0.02 & 0.7 & 0.5 \\ 0.34 & 0.02 & 0.5 & 0.7 \\ 0.34 & 0.02 & 0.3 & 0.3 \\ 1 & 1 & 0.5 & 0.5 \\ 1 & 1 & 0.3 & 0.7 \\ 1 & 1 & 0.3 & 0.5 \\ 0.64 & 0.44 & 0.1 & 0.3 \end{bmatrix} \end{array}$$

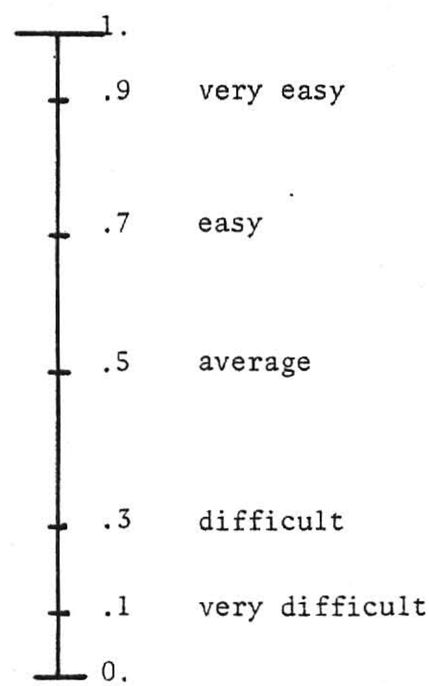


Fig. 5.3. Bipolor scale for the qualitative attributes.

3.2 Solution by the Conjunctive Constraints

In conjunctive decision making, all the standards must be met. Hence, standards will be set and then increased or decreased in an iterative fashion until only one alternative meets all of the standards. This solution is very simple because it requires little complicated analysis.

Iteration #1:

The decision maker will take any alternatives which possess the following properties:

1. Total computational time X_1 at most 1010
2. Accuracy X_2 $(= \Delta_R/\Delta_e)$ at most 10
3. Ease of understanding X_3 at least average
4. Ease of use X_4 at least average

Property	Alternative possessing that property							
1					A_6	A_7	A_8	A_9
2					A_6	A_7	A_8	A_9
3	A_1	A_2	A_3	A_4	A_6			
4		A_2	A_3	A_4	A_6	A_7	A_8	

It can easily be seen that only A_6 possesses all of the four properties. Hence select A_6 (Algebraic extraction method).

3.3 Solution by Simple Additive Weighting Method

This method [86, 87, 91] takes an alternative which has the maximum weighted averages. The DM assesses the weight using the given hierarchical structure of attributes. First he assesses the weight about the four major classes, then he judges within each class. The complete assessment is given in Fig. 5.4.

With the assessed weight and the comparable attribute values given in the D matrix, the weighted average values for the alternatives are:

$$A_1 = \sum_{j=1}^{4} w_j x_{1j} = 0.361 \qquad A_6 = 0.75$$

$$A_2 = 0.381 \qquad A_7 = 0.77$$

$$A_3 = 0.396 \qquad A_8 = 0.69$$

$$A_4 = 0.416 \qquad A_9 = 0.39$$

$$A_5 = 0.256$$

therefore, A_7 (Fast algorithm) is selected.

3.4 Solution by Linear Assignment Method

This solution [85] employs a linear compensatory process for attribute interaction and combination. In the process only ordinal data, rather than cardinal data, are used as the input. And the qualitative attributes do not need to be scaled. The alternatives are ranked as shown in Table 5.2.

Now, let us define a product-attribute matrix Π as a square (9 x 9) nonnegative matrix whose elements Π_{ik} represent the frequency (or number) that A_i is ranked the K^{th} attributewise ranking. For the different weight $\underline{w} = (w_1, w_2, w_3, w_4) = (.3, .2, .2, .3)$, each entry of Π matrix is the sum of the weights of all of the entries with that ranking for that alternative. This is shown in Fig. 5.5.

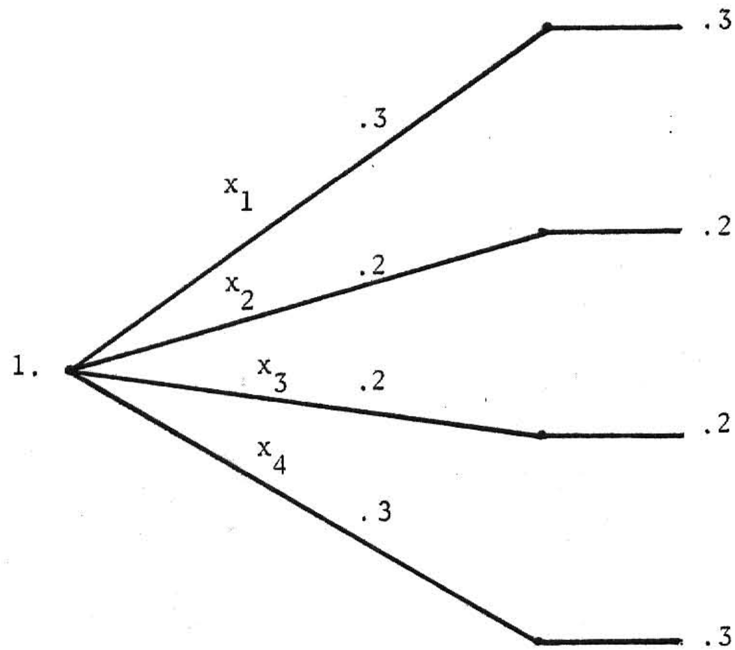


Fig. 5.4. Assessment of attribute weights.

TABLE 5.2. Ranking of Alternatives

Attribute / Rank	$X_1(W_1)$	$X_2(W_2)$	$X_3\ (W_3)$	$X_4\ (W_4)$
1st	A_6, A_7, A_8	A_6, A_7, A_8	A_1	A_4, A_7
2nd			A_2,A_3	
3rd				A_2,A_3,A_6,A_8
4th	A_9	A_9	A_4,A_6	
5th	A_3,A_4,A_5	A_1,A_2		
6th			A_5,A_7,A_8	
7th		A_3,A_4,A_5		A_1,A_5,A_9
8th	A_1,A_2			
9th			A_9	

For the different weight $\underline{w} = (w_1, w_2, w_3, w_4) = (.3, .2, .2, .3)$, Π matrix becomes

$$
\Pi = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \\ A_9 \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1^{st} & 2^{nd} & 3^{rd} & 4^{th} & 5^{th} & 6^{th} & 7^{th} & 8^{th} & 9^{th} \end{array} \\
\left[\begin{array}{ccccccccc}
w_3 & & & & \frac{w_2}{2}, & \frac{w_2}{2} & \frac{w_4}{3} & \frac{w_1}{2} + \frac{w_4}{3}, & \frac{w_1}{2} + \frac{w_4}{3} \\
 & \frac{w_3}{2} & \frac{w_3}{2} + \frac{w_4}{4}, & \frac{w_4}{4} & \frac{w_2}{2} + \frac{w_4}{4}, & \frac{w_2}{2} + \frac{w_4}{4} & & \frac{w_1}{2} & \frac{w_1}{2} \\
 & \frac{w_3}{2} & \frac{w_3}{2} + \frac{w_4}{4} & \frac{w_4}{4} & \frac{w_1}{3} + \frac{w_4}{4}, & \frac{w_1}{3} + \frac{w_4}{4} & \frac{w_1}{3} + \frac{w_2}{3} & \frac{w_2}{3} & \frac{w_2}{3} \\
\frac{w_4}{2} & \frac{w_4}{2} & & \frac{w_3}{2} & \frac{w_1}{3} + \frac{w_3}{2}, & \frac{w_1}{3} & \frac{w_1}{3} + \frac{w_2}{3} & \frac{w_2}{3} & \frac{w_2}{3} \\
 & & & & \frac{w_1}{3} & \frac{w_1}{3} + \frac{w_3}{3} & \frac{w_1}{3} + \frac{w_2}{3} + \frac{w_2}{3} + \frac{w_4}{3} & \frac{w_2}{3} + \frac{w_3}{3} + \frac{w_4}{3} & \frac{w_2}{3} + \frac{w_4}{3} \\
\frac{w_1}{3} + \frac{w_2}{3}, & \frac{w_1}{3} + \frac{w_2}{3}, & \frac{w_1}{3} + \frac{w_3}{3}, + \frac{w_4}{4} & \frac{w_3}{2} + \frac{w_4}{4} & \frac{w_3}{2} + \frac{w_4}{4} & \frac{w_4}{4} & & & \\
\frac{w_1}{3} + \frac{w_2}{3} + \frac{w_4}{2}, & \frac{w_1}{3} + \frac{w_2}{3} + \frac{w_4}{2}, & \frac{w_1}{3} + \frac{w_2}{3} & & & \frac{w_3}{3} & \frac{w_3}{3} & \frac{w_3}{3} & \\
\frac{w_1}{3} + \frac{w_2}{3}, & \frac{w_1}{3} + \frac{w_2}{3}, & \frac{w_1}{3} + \frac{w_2}{3} + \frac{w_4}{4}, & \frac{w_4}{4} & \frac{w_4}{4} & \frac{w_3}{3} + \frac{w_4}{4} & \frac{w_3}{3} & \frac{w_3}{3} & \\
 & & & w_1 + w_2 & & & \frac{w_4}{3} & \frac{w_4}{3} & w_3 + \frac{w_4}{3}.
\end{array}\right]
\end{array}
$$

Fig. 5.5. Product-attribute matrix Π

$$
= \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \\ A_9 \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\
\begin{bmatrix}
.2 & 0 & 0 & 0 & .1 & .1 & .1 & .25 & .25 \\
0 & .1 & .175 & .075 & .175 & .175 & 0 & .15 & .15 \\
0 & .1 & .175 & .075 & .175 & .175 & .176 & .067 & .067 \\
.15 & .15 & 0 & .1 & .2 & .1 & .167 & .067 & .067 \\
0 & 0 & 0 & 0 & .1 & .167 & .333 & .233 & .167 \\
.167 & .167 & .242 & .175 & .175 & .075 & 0 & 0 & 0 \\
.317 & .317 & .167 & 0 & 0 & 0 & .067 & .067 & 0 \\
.167 & .167 & .242 & .075 & .075 & .142 & .067 & .067 & 0 \\
0 & 0 & 0 & .5 & 0 & 0 & .1 & .1 & .3
\end{bmatrix}
\end{array}
$$

Fig. 5.5. Product-attribute matrix Π (continued)

The larger Π_{ik} indicates the more concordance in assigning A_i to the K^{th} overall rank. Hence the problem is to find A_i for each K, K = 1,2,...,9 which maximize

$$\sum_{k=1}^{9} \Pi_{ik}$$

Let us define permutation matrix P as (9 x 9) square matrix whose element P_{ik} = 1 if A_i is assigned to overall rank K, and P_{ik}=0 otherwise. The linear assignment method can be written by the following LP format,

$$\max \sum_{i=1}^{9} \sum_{k=1}^{9} \Pi_{ik} P_{ik}$$

$$\text{s.t.} \quad \sum_{k=1}^{9} P_{ik} = 1, \qquad i = 1,2,\ldots,9$$

$$\sum_{i=1}^{9} P_{ik} = 1, \qquad k = 1,2,\ldots,9$$

$$P_{ik} \geq 0, \qquad \forall\ i, k$$

The solution (by computer output; refer to Appendix 2) is

$P_{61} = 1$ $\qquad$ $P_{36} = 1$

$P_{72} = 1$ $\qquad$ $P_{57} = 1$

$P_{83} = 1$ $\qquad$ $P_{28} = 1$

$P_{94} = 1$ $\qquad$ $P_{19} = 1$

$P_{45} = 1$ $\qquad$ the others are all zero.

The optimal permutation matrix P* is

$$P^* = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \\ A_9 \end{array} \begin{array}{c} \begin{array}{ccccccccc} \text{1st} & \text{2nd} & \text{3rd} & \text{4th} & \text{5th} & \text{6th} & \text{7th} & \text{8th} & \text{9th} \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

That is, the final ranking of $(A_6, A_7, A_8, A_9, A_4, A_3, A_5, A_2, A_1)$ is obtained.

3.5 Solution by ELECTRE (Elimination et Choice Translating Reality) Method

This method [92, 93, 96] uses the concept of 'Outranking relationship'. Outranking relationship of $A_k \rightarrow A_\ell$ says that even though two alternatives k and ℓ are nondominated each other mathematically, the DM accepts the risk of regarding A_k as almost surely better than A_ℓ [97]. Through the successive assessment of outranking relationship of the other alternatives, the dominated alternatives by the outranking relationship can be eliminated. ELECTRE sets the criteria for the mechanical assessment of the outranking relationships.

The ELECTRE method takes the following steps:

1. Calculate the normalized decision matrix:

$$R = \begin{array}{c c} & \begin{array}{cccc} X_1 & X_2 & X_3 & X_4 \end{array} \\ \begin{array}{c} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \\ A_9 \end{array} & \begin{bmatrix} 0.6463 & 0.0453 & 0.5614 & 0.2 \\ 0.6463 & 0.0453 & 0.4366 & 0.3333 \\ 0.2134 & 0.5758 & 0.4366 & 0.3333 \\ 0.2134 & 0.5758 & 0.3119 & 0.4667 \\ 0.2134 & 0.5758 & 0.1871 & 0.2 \\ 0.0715 & 0.0129 & 0.3119 & 0.3333 \\ 0.0715 & 0.0129 & 0.1871 & 0.4667 \\ 0\ 0715 & 0.0129 & 0.1871 & 0.3333 \\ 0.1119 & 0.0291 & 0.0624 & 0.2 \end{bmatrix} \end{array}$$

2. Calculate the weighted normalized decision matrix:

$V = RW$

$$= \begin{array}{c c} & \begin{array}{cccc} X_1 & X_2 & X_3 & X_4 \end{array} \\ \begin{array}{c} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \\ A_9 \end{array} & \begin{bmatrix} 0.6463 & 0.0453 & 0.5614 & 0.2 \\ 0.6463 & 0.0453 & 0.4366 & 0.3333 \\ 0.2134 & 0.5758 & 0.4366 & 0.3333 \\ 0.2134 & 0.5758 & 0.3119 & 0.4667 \\ 0.2134 & 0.5758 & 0.1871 & 0.2 \\ 0.0715 & 0.0129 & 0.3119 & 0.3333 \\ 0.0715 & 0.0129 & 0.1871 & 0.4667 \\ 0.0715 & 0.0129 & 0.1871 & 0.3333 \\ 0.1119 & 0.0291 & 0.0624 & 0.2 \end{bmatrix} \end{array} \begin{bmatrix} 0.3 & 0 & 0 & 0 \\ 0 & 0.2 & 0 & 0 \\ 0 & 0 & 0.2 & 0 \\ 0 & 0 & 0 & 0.3 \end{bmatrix}$$

$$
\begin{array}{cc}
 & \begin{array}{cccc} X_1 & X_2 & X_3 & X_4 \end{array} \\
= \begin{array}{c} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \\ A_9 \end{array} &
\begin{bmatrix}
0.1939 & 0.0091 & 0.1123 & 0.06 \\
0.1939 & 0.0091 & 0.0873 & 0.10 \\
0.0640 & 0.1152 & 0.0873 & 0.10 \\
0.0640 & 0.1152 & 0.0624 & 0.14 \\
0.0640 & 0.1152 & 0.0374 & 0.06 \\
0.0215 & 0.0026 & 0.0624 & 0.10 \\
0.0215 & 0.0026 & 0.0374 & 0.14 \\
0.0215 & 0.0026 & 0.0374 & 0.10 \\
0.0336 & 0.0058 & 0.0125 & 0.06
\end{bmatrix}
\end{array}
$$

Recall that X_1 and X_2 are cost attributes, and X_3 and X_4, benefit attributes.

3. Determine concordance and discordance set:

The concordance set $C_{k\ell}$ of A_k and A_ℓ is composed of all criteria for which A_k is preferred to A_ℓ. i. e.,

$C_{k\ell} = \{j | X_{kj} \geq X_{\ell j}\}$ and the discordance set is $D_{k\ell} = \{j | X_{kj} < X_{\ell j}\}$

$= J - C_{k\ell}$

$C_{12} = \{1, 2, 3\}$	$D_{12} = \{4\}$
$C_{13} = \{2, 3\}$	$D_{13} = \{1, 4\}$
$C_{14} = \{2, 3\}$	$D_{14} = \{1, 4\}$
$C_{15} = \{2, 3, 4\}$	$D_{15} = \{1\}$
$C_{16} = \{3\}$	$D_{16} = \{1, 2, 4\}$
$C_{17} = \{3\}$	$D_{17} = \{1, 2, 4\}$
$C_{18} = \{3\}$	$D_{18} = \{1, 2, 4\}$
$C_{19} = \{3, 4\}$	$D_{19} = \{1, 2\}$
$C_{21} = \{1, 2, 4\}$	$D_{21} = \{3\}$
$C_{23} = \{2, 3, 4\}$	$D_{23} = \{1\}$
$C_{24} = \{2, 3\}$	$D_{24} = \{1, 4\}$
$C_{25} = \{2, 3, 4\}$	$D_{25} = \{1\}$
$C_{26} = \{3, 4\}$	$D_{26} = \{1, 2\}$
$C_{27} = \{3\}$	$D_{27} = \{1, 2, 4\}$
$C_{28} = \{3, 4\}$	$D_{28} = \{1, 2\}$
$C_{29} = \{3, 4\}$	$D_{29} = \{1, 2\}$

$C_{31} = \{1, 4\}$
$C_{32} = \{1, 3, 4\}$
$C_{34} = \{1, 2, 3\}$
$C_{35} = \{1, 2, 3, 4\}$
$C_{36} = \{3, 4\}$
$C_{37} = \{3\}$
$C_{38} = \{3, 4\}$
$C_{39} = \{3, 4\}$
$C_{41} = \{1, 4\}$
$C_{42} = \{1, 4\}$
$C_{43} = \{1, 2, 4\}$
$C_{45} = \{1, 2, 3, 4\}$
$C_{46} = \{3, 4\}$
$C_{47} = \{3, 4\}$
$C_{48} = \{3, 4\}$
$C_{49} = \{3, 4\}$
$C_{51} = \{1, 4\}$
$C_{52} = \{1\}$
$C_{53} = \{1, 2\}$
$C_{54} = \{1, 2\}$
$C_{56} = \emptyset$
$C_{57} = \{3\}$
$C_{58} = \{3\}$
$C_{59} = \{3, 4\}$
$C_{61} = \{1, 2, 4\}$
$C_{62} = \{1, 2, 4\}$
$C_{63} = \{1, 2, 4\}$
$C_{64} = \{1, 2, 3\}$
$C_{65} = \{1, 2, 3, 4\}$
$C_{67} = \{1, 2, 3\}$
$C_{68} = \{1, 2, 3, 4\}$
$C_{69} = \{1, 2, 3, 4\}$

$D_{31} = \{2, 3\}$
$D_{32} = \{2\}$
$D_{34} = \{4\}$
$D_{35} = \emptyset$
$D_{36} = \{1, 2\}$
$D_{37} = \{1, 2, 4\}$
$D_{38} = \{1, 2\}$
$D_{39} = \{1, 2\}$
$D_{41} = \{2, 3\}$
$D_{42} = \{2, 3\}$
$D_{43} = \{3\}$
$D_{45} = \emptyset$
$D_{46} = \{1, 2\}$
$D_{47} = \{1, 2\}$
$D_{48} = \{1, 2\}$
$D_{49} = \{1, 2\}$
$D_{51} = \{2, 3\}$
$D_{52} = \{2, 3, 4\}$
$D_{53} = \{3, 4\}$
$D_{54} = \{3, 4\}$
$D_{56} = \{1, 2, 3, 4\}$
$D_{57} = \{1, 2, 4\}$
$D_{58} = \{1, 2, 4\}$
$D_{59} = \{1, 2\}$
$D_{61} = \{3\}$
$D_{62} = \{3\}$
$D_{63} = \{3\}$
$D_{64} = \{4\}$
$D_{65} = \emptyset$
$D_{67} = \{4\}$
$D_{68} = \emptyset$
$D_{69} = \emptyset$

$C_{71} = \{1, 2, 4\}$ $D_{71} = \{3\}$
$C_{72} = \{1, 2, 4\}$ $D_{72} = \{3\}$
$C_{73} = \{1, 2, 4\}$ $D_{73} = \{3\}$
$C_{74} = \{1, 2, 4\}$ $D_{74} = \{3\}$
$C_{75} = \{1, 2, 3, 4\}$ $D_{75} = \emptyset$
$C_{76} = \{1, 2, 4\}$ $D_{76} = \{3\}$
$C_{78} = \{1, 2, 3, 4\}$ $D_{78} = \emptyset$
$C_{79} = \{1, 2, 3, 4\}$ $D_{79} = \emptyset$
$C_{81} = \{1, 2, 4\}$ $D_{81} = \{3\}$
$C_{82} = \{1, 2, 4\}$ $D_{82} = \{3\}$
$C_{83} = \{1, 2, 4\}$ $D_{83} = \{3\}$
$C_{84} = \{1, 2\}$ $D_{84} = \{3, 4\}$
$C_{85} = \{1, 2, 3, 4\}$ $D_{85} = \emptyset$
$C_{86} = \{1, 2, 4\}$ $D_{86} = \{3\}$
$C_{87} = \{1, 2, 3\}$ $D_{87} = \{4\}$
$C_{89} = \{1, 2, 3, 4\}$ $D_{89} = \emptyset$
$C_{91} = \{1, 2, 4\}$ $D_{91} = \{3\}$
$C_{92} = \{1, 2\}$ $D_{92} = \{3, 4\}$
$C_{93} = \{1, 2\}$ $D_{93} = \{3, 4\}$
$C_{94} = \{1, 2\}$ $D_{94} = \{3, 4\}$
$C_{95} = \{1, 2, 4\}$ $D_{95} = \{3\}$
$C_{96} = \emptyset$ $D_{96} = \{1, 2, 3, 4\}$
$C_{97} = \emptyset$ $D_{97} = \{1, 2, 3, 4\}$
$C_{98} = \emptyset$ $D_{98} = \{1, 2, 3, 4\}$

4. Calculate the concordance matrix

$$C_{12} = \sum_{j \varepsilon C_{12}} w_j = w_1 + w_2 + w_3 = 0.3 + 0.2 + 0.2 = 0.7$$

$$C_{13} = \sum_{j \varepsilon C_{13}} w_j = w_1 + w_3 = 0.2 + 0.2 = 0.4$$

.
.
.
etc.,

then the concordance matrix is,

$$C = \begin{array}{c} 1\\2\\3\\4\\5\\6\\7\\8\\9 \end{array}\left[\begin{array}{ccccccccc}
— & 0.7 & 0.4 & 0.4 & 0.7 & 0.2 & 0.2 & 0.2 & 0.5 \\
0.8 & — & 0.7 & 0.4 & 0.7 & 0.5 & 0.2 & 0.5 & 0.5 \\
0.6 & 0.8 & — & 0.7 & 1 & 0.5 & 0.2 & 0.5 & 0.5 \\
0.6 & 0.6 & 0.8 & — & 1 & 0.5 & 0.5 & 0.5 & 0.5 \\
0.6 & 0.3 & 0.5 & 0.5 & — & 0 & 0.2 & 0.2 & 0.5 \\
0.8 & 0.8 & 0.8 & 0.7 & 1 & — & 0.7 & 1 & 1 \\
0.8 & 0.8 & 0.8 & 0.8 & 1 & 0.8 & — & 1 & 1 \\
0.8 & 0.8 & 0.8 & 0.5 & 1 & 0.8 & 0.7 & — & 1 \\
0.8 & 0.5 & 0.5 & 0.5 & 0.8 & 0 & 0 & 0 & —
\end{array}\right]$$

5. Calculate the discordance matrix.

An element $d_{k\ell}$ of the D_x matrix is obtained as,

$$d_{k\ell} = \frac{\max\limits_{j\varepsilon D_{k\ell}} | v_{kj} - v_{\ell j}|}{\max\limits_{j\varepsilon J} |v_{kj} - v_{\ell j}|}$$

$$d_{12} = \frac{0.04}{0.04} = 1$$

$$d_{21} = \frac{0.025}{0.04} = 0.625$$

$$d_{13} = \frac{0.1299}{\max \{0.1299, 0.1061, 0.025, 0.04\}} = \frac{0.1299}{0.1299} = 1$$

$$d_{31} = \frac{0.1061}{0.1299} = 0.8168$$

$$d_{14} = \frac{0.1299}{\max\{0.1299,\ 0.1061,\ 0.0499,\ 0.08\}} = 1$$

$$d_{41} = \frac{0.1061}{0.1299} = 0.8168$$

$$d_{15} = \frac{0.1299}{\max\{0.1299,\ 0.1061,\ 0.0749,\ 0\}} = 1$$

$$d_{51} = \frac{0.1061}{0.1299} = 0.8168$$

$$d_{16} = \frac{0.1724}{\max\{0.1724,\ 0.0065,\ 0.0499,\ 0.04\}} = 1$$

$$d_{61} = \frac{0.0499}{0.1724} = 0.2894$$

.
.
.
etc.,

The discordance matrix is,

$$D_x = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \begin{array}{c} \begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\ \begin{bmatrix} - & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ .625 & - & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ .8168 & .8168 & - & 1 & 0 & 1 & 1 & 1 & 1 \\ .8168 & .8168 & .6225 & - & 0 & 1 & 1 & 1 & 1 \\ .8168 & .8168 & 1 & 1 & - & 1 & 1 & 1 & 1 \\ .2894 & .1444 & .2211 & .3552 & 0 & - & 1 & 0 & 0 \\ .4345 & .2894 & .4432 & .2220 & 0 & .625 & - & 0 & 0 \\ .4345 & .2894 & .4432 & .3552 & 0 & 1 & 1 & - & 0 \\ .6226 & .4666 & .6837 & .7313 & .2276 & 1 & 1 & 1 & - \end{bmatrix} \end{array}$$

6. Determine the concordance dominance matrix: if we take the threshold value of $c_{k\ell}$ as the average concordance index, then

$$\bar{c} = \frac{(.7 + .4 + .4 + \cdots + .5 + .8)}{9 \times 8} = \frac{43.5}{72} = 0.6042$$

The concordance dominance matrix is,

$$F = \begin{array}{cc} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} & \begin{bmatrix} - & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & - & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & - & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & - & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & - & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & - & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & - & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & - & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & - \end{bmatrix} \end{array}$$

7. Determine the discordance dominance matrix: the value of $\bar{d}$ is taken as the average discordance index, then

$$\bar{d} = \frac{(1 + 0.8168 + \cdots + 1)}{9 \times 8} = \frac{49.4266}{72} = 0.6865$$

The discordance dominance matrix is,

$$G = \begin{array}{cc} & \begin{matrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{matrix} \\ \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{matrix} & \begin{bmatrix} - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & - & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & - & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & - & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 & - & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & - & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & - & 1 \\ 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & - \end{bmatrix} \end{array}$$

8. Determine the aggregate dominance matrix; Combine matrices of F and G, the aggregate matrix is obtained as,

$$
E = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \end{array} \\
\begin{bmatrix}
- & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
1 & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & - & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & - & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & - & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & - & 0 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & - & 1 & 1 \\
1 & 1 & 1 & 0 & 1 & 0 & 0 & - & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -
\end{bmatrix}
\end{array}
$$

9. Eliminate the less favorable alternatives; the E matrix renders the following over-ranking relationships;

$A_2 \to A_1$	$A_6 \to A_1$	$A_7 \to A_1$	$A_8 \to A_1$
$A_3 \to A_5$	$A_6 \to A_2$	$A_7 \to A_2$	$A_8 \to A_2$
$A_4 \to A_3$	$A_6 \to A_3$	$A_7 \to A_3$	$A_8 \to A_3$
$A_4 \to A_5$	$A_6 \to A_4$	$A_7 \to A_4$	$A_8 \to A_5$
	$A_6 \to A_5$	$A_7 \to A_5$	$A_8 \to A_9$
	$A_6 \to A_8$	$A_7 \to A_6$	$A_9 \to A_1$
	$A_6 \to A_9$	$A_7 \to A_8$	$A_9 \to A_5$
		$A_7 \to A_9$	

Combining above relations as much as possible $\to$

$$
A_7 \to A_6 \to A_8 \begin{cases} \to A_1 \\ \to A_5 \\ \to A_2 \to A_1 \\ \to A_3 \to A_5 \end{cases}
$$

$$
A_9 \begin{cases} \to A_1 \\ \to A_5 \end{cases}
$$

$$
A_7 \to A_6 \to A_4 \begin{cases} \to A_3 \\ \to A_5 \end{cases}
$$

We can see that either $A_7 \to A_6 \to A_8$ relationship or $A_7 \to A_6 \to A_4$ hold predominantly. Also some other relations can be identified partly. But we cannot tell the streamlined relation.

By tightening the threshold value: (Steps 1 to 5 are the same as before)

$\bar{c}$ increased to 0.8, $\bar{d}$ lowered to 0.3

6. The concordance dominance matrix becomes:

$$F = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \begin{bmatrix} - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & - & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & - & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & - & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & - & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & - & 1 & 1 \\ 1 & 1 & 1 & 0 & 1 & 1 & 0 & - & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & - \end{bmatrix}$$

7. The discordance dominance matrix is

$$G = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \begin{bmatrix} - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & - & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & - & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & - & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & - & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & - & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & - & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & - \end{bmatrix}$$

8. Determine the aggregate dominance matrix;

$$E = \begin{array}{c} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \\ 8 \\ 9 \end{array} \begin{bmatrix} - & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & - & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & - & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & - & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & - & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & - & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 & 1 & 0 & - & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & - & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & - \end{bmatrix}.$$

9. The E matrix renders the following overranking relationships;

$A_3 \rightarrow A_5$	$A_4 \rightarrow A_5$	$A_6 \rightarrow A_1$	$A_7 \rightarrow A_2$	$A_8 \rightarrow A_2$
		$A_6 \rightarrow A_2$	$A_7 \rightarrow A_4$	$A_8 \rightarrow A_5$
		$A_6 \rightarrow A_3$	$A_7 \rightarrow A_5$	$A_8 \rightarrow A_9$
		$A_6 \rightarrow A_5$	$A_7 \rightarrow A_8$	
		$A_6 \rightarrow A_8$	$A_7 \rightarrow A_9$	$A_9 \rightarrow A_5$
		$A_6 \rightarrow A_9$		

By the graphical representation,

$$\begin{tikzcd}
& A_1 & & \\
A_6 \arrow[ur] \arrow[r] \arrow[dr] & A_3 \arrow[r] & A_5 & \\
& A_8 \arrow[r] \arrow[dr] & A_9 \arrow[r] & A_5 \\
& & A_2 &
\end{tikzcd}$$

$$\begin{tikzcd}
A_7 \arrow[r] \arrow[dr] & A_4 \arrow[r] & A_5 & \\
& A_8 \arrow[r] \arrow[dr] & A_9 \arrow[r] & A_5 \\
& & A_2 &
\end{tikzcd}$$

We can easily see that A_6 and A_7 dominate others.

3.6. Solution by Technique for Order Preference by Similarity to Ideal Solution (TOPSIS)

TOPSIS [89] considers the distances to both the ideal and the negative-ideal solutions simultaneously by taking the relative closeness to the ideal solution. It is shown that the simple additive weighting (SAW) is a special case of TOPSIS. A favorable reliability evaluation technique will be selected after a series of successive steps of TOPSIS.

1. Calculate the normalized decision matrix.

$$R = \begin{array}{c} A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \\ A_9 \end{array} \begin{bmatrix} 0.6463 & 0.0453 & 0.5614 & 0.2 \\ 0.6463 & 0.0453 & 0.4366 & 0.3333 \\ 0.2134 & 0.5758 & 0.4366 & 0.3333 \\ 0.2134 & 0.5758 & 0.3119 & 0.4667 \\ 0.2134 & 0.5758 & 0.1871 & 0.2 \\ 0.0715 & 0.0129 & 0.3119 & 0.3333 \\ 0.0715 & 0.0129 & 0.1871 & 0.4667 \\ 0.0715 & 0.0129 & 0.1871 & 0.3333 \\ 0.1119 & 0.0291 & 0.0624 & 0.2 \end{bmatrix}$$

2. Calculate the weighted decision matrix;

Assume that the relative importance of attribute is given by the DM as $\underline{w} = (w_1, w_2, w_3, w_4) = (0.3, 0.2, 0.2, 0.3)$. The weighted decision matrix is then

$$V = \begin{array}{c} \\ A_1 \\ A_2 \\ A_3 \\ A_4 \\ A_5 \\ A_6 \\ A_7 \\ A_8 \\ A_9 \end{array} \begin{array}{c} \begin{array}{cccc} X_1 & X_2 & X_3 & X_4 \end{array} \\ \begin{bmatrix} 0.1939 & 0.0091 & 0.1123 & 0.06 \\ 0.1939 & 0.0091 & 0.0873 & 0.10 \\ 0.0640 & 0.1152 & 0.0873 & 0.10 \\ 0.0640 & 0.1152 & 0.0624 & 0.14 \\ 0.0640 & 0.1152 & 0.0374 & 0.06 \\ 0.0215 & 0.0026 & 0.0624 & 0.10 \\ 0.0215 & 0.0026 & 0.0374 & 0.14 \\ 0.0215 & 0.0026 & 0.0374 & 0.10 \\ 0.0336 & 0.0058 & 0.0125 & 0.06 \end{bmatrix} \end{array}$$

3. Determine the ideal and negative-ideal solutions:
   Two artificial alternatives A^* and A^- are defined as

$$A^* = (\min_i V_{i1}, \quad \min_i V_{i2}, \quad \max_i V_{i3}, \quad \max_i V_{i4})$$
$$= (0.0215, \quad 0.0026, \quad 0.1123, \quad 0.14)$$
$$A^- = (\max_i V_{i1}, \quad \max_i V_{i2}, \quad \min_i V_{i3}, \quad \min_i V_{i4})$$
$$= (0.1939, \quad 0.1152, \quad 0.0125, \quad 0.06)$$

4. Calculate the separation measures:

The separation between each alternative can be measured by the n-dimensional Euclidean distance. The separation of each alternative to ideal one is then given by

$$S_{i*} = \sqrt{\sum_{j=1}^{4} (V_{ij} - V_j^*)^2}, \qquad i = 1,2,\ldots,9$$

$S_{1*} = 0.1902$, $\quad S_{4*} = 0.1303$ $\quad S_{7*} = 0.0749$

$S_{2*} = 0.1789$ $\quad S_{5*} = 0.1628$ $\quad S_{8*} = 0.0849$

$S_{3*} = 0.1293$ $\quad S_{6*} = 0.0640$ $\quad S_{9*} = 0.1285$

Similarly the separation to the negative-ideal one is given by

$$S_{i-} = \sqrt{\sum_{j=1}^{4} (V_{ij} - V_j^-)^2}, \qquad i = 1,2,\ldots,9$$

$S_{1-} = 0.1457$	$S_{4-} = 0.1605$	$S_{7-} = 0.2223$
$S_{2-} = 0.1358$	$S_{5-} = 0.1323$	$S_{8-} = 0.2112$
$S_{3-} = 0.1551$	$S_{6-} = 0.2156$	$S_{9-} = 0.1941$

5. Calculate the relative closeness to the ideal solution:

The relative closeness of A_i with respect to A^* is

$C_{1*} = S_{1-}/(S_{1*} + S_{1-}) = 0.4338$
$C_{2*} = 0.4315$
$C_{3*} = 0.5454$
$C_{4*} = 0.5519$
$C_{5*} = 0.4483$
$C_{6*} = 0.7711$
$C_{7*} = 0.7480$
$C_{8*} = 0.7133$
$C_{9*} = 0.6017$

It is clear that $C_{i*} = 1$ if $A_i = A^*$ and $C_{i*} = 0$ if $A_i = A^-$. An alternative A_i is closer to A^* as C_{i*} approaches to 1.

6. Rank the preference order: According to the descending order of C_{i*}, the preference order is:

A_6, A_7, A_8, A_9, A_4, A_3, A_5, A_1, A_2

4. Comparison of results by the MADM methods

To select a suitable reliability evaluation technique, such MADM methods as the Conjunctive Constraints, Simple Additive Weighting Method, Linear Assignment Method, ELECTRE Method and TOPSIS are employed. Comparison of results by these methods is shown in Table 5.3.

TABLE 5.3. Comparison of Results by MADM Methods

MADM Methods	Favorably selected alternatives or Ranking 1st	2nd	3rd	4th	5th	6th	7th	8th	9th	Remarks
Conjunctive Constraints	A_6									
Simple Additive Weighting	A_7	A_6	A_8	A_4	A_3	A_9	A_2	A_1	A_5	The weighted average value of A_7 and A_6 are 0.77 and 0.75
Linear Assignment	A_6	A_7	A_8	A_9	A_4	A_3	A_5	A_2	A_1	
ELECTRE	A_7	A_6	A_8 →		A_2 A_3 → A_9			A_1 A_5		When the value of $\bar{c}$ and $\bar{d}$ is taken as the average of corresponding index i.e., $\bar{c} = 0.6042$, $\bar{d} = 0.6863$
	or									
	A_7	A_6 →		A_4 →			A_3 A_5			
	A_6 →	A_1 A_3 → A_8 →			A_5 A_9 → A_5 A_2					When tightening the threshold value $\bar{c}$ 0.8, $\bar{d}$ 0.3
	or									
	A_7 →	A_4 → A_8 →			A_5 A_9 → A_5 A_2					
TOPSIS	A_6	A_7	A_8	A_9	A_4	A_3	A_5	A_1	A_2	Very similar ranking to Linear Assignment Method

5. Concluding Remarks

As Table 5.3 shows, A_6 (Algebraic Extraction method) appears to be most favorable alternative when Conjunctive Constraints, Linear Assignment Method, or TOPSIS are applied, while Simple Additive Weighting and ELECTRE recommends A_7 (Fast Algorithm) as a most favorable one. It can easily be seen that A_6 or A_7 (Fast Algorithm is just the extension of Algebraic Extration method i.e., A_6) is selected as a desirable method in this problem.

The goals (or objectives) of the system reliability evaluation techniques may be: a) Computational time saving on a computer, b) Accuracy of the solution, c) Ease of understanding the logic of the technique, and d) Ease of use of the technique. These goals (or objectives) are often in conflict with each other. As a result the solution to the multicriteria decision problem is a compromised one, not necessarily an optimum one. In general the basic problem is to decide what kind of evaluation technique should be employed depending on the size and configuration of the system. The selection of a particular evaluation method will depend upon the degree of achieving the Criteria (objectives) mentioned above, and the compromises required.

The use of MADM methods are demonstrated through the evaluation and selection of a system reliability evaluation technique from several alternatives. The procedure can be applied to the selection of relevant evaluation technique for the other system configurations in systems reliability.

REFERENCES

[1] J. A. Abraham, "An improved algorithm for network reliability," IEEE Trans. Reliability, vol R-20, 1979 Apr, pp 58-61.

[2] K. K. Aggarwal, J. S. Gupta, K. B. Misra, "A new method for system reliability evaluation," Microelectronics and Reliability, vol 12, 1973, pp 435-440.

[3] K. K. Aggarwal, J. S. Gupta, K. B. Misra, "Computational time and absolute error comparison for reliability expression derived by various methods," Microelectronics and Reliability, vol 14, 1975, pp 465-467.

[4] K. K. Aggarwal, J. S. Gupta, K. B. Misra, "A simple method for reliability evaluation of a communication system," IEEE Trans. Reliability, 1975 May, pp 563-566.

[5] K. K. Aggarwal, K. B. Misra, J. S. Gupta, "Reliability evaluation: A comparative study of different techniques," Microelectronic and reliability, vol 14, 1975 Feb, pp 49-56.

[6] K. K. Aggarwal, K. B. Misra, J. S. Gupta, "A fast algorithm for reliability evaluation," IEEE Trans. Reliability, vol R-24, 1975 Apr, pp 83-85.

[7] K. K. Aggarwal, S. Rai, "Symbolic reliability evaluation using logical signal relations," IEEE Trans. Reliability, vol R-27, 1978 Aug, pp 202-205.

[8] ARINC Res. Corp., Reliability Engineering, Englewood Cliffs, N.J., Prentice Hall, 1964.

[9] S. Arunkumar, S. H. Lee, "Enumeration of all minimal cut-sets for a node pair in a graph," IEEE Trans. Reliability, vol R-28, 1979 Apr, pp 51-55.

[10] M. O. Ball, "Computing network reliability," Operations Research, vol 27, no 4, 1979 Jul-Aug, pp 823-838.

[11] S. K. Banerjee, K. Rajamani, "Parametric representation of probability in two dimensions - A new approach in system reliability evaluation," IEEE Trans. Reliability, vol R-21, 1972 Feb, pp 56-60.

[12] S. K. Banerjee, K. Rajamani, "Closed form solutions for delta-star and star-delta conversions of reliability networks," IEEE Trans. Reliability, vol R-25, 1976 Jun, pp 118-119.

[13] J. R. Batts, "Computer program for approximating system reliability - part II," IEEE Trans. Reliability, vol R-20, 1971 May, pp 88-90.

[14] I. Bazovsky, Reliability Theory and Practice, Prentice Hall, New Jersey, 1961.

[15] M. Bellmore, P. A. Jensen, "An implicit enumeration scheme for proper cut generation," Technometrics, vol 12, 1970 Nov, pp 775-788.

[16] R. G. Bennetts, "On the analysis of fault trees," IEEE Trans. Reliability, vol R-24, 1975 Aug, pp 175-185.

[17] J. E. Biegel, "Determination of tie sets and cut sets for a system without feedback," IEEE Trans. Reliability, vol R-26, no 1, 1977 Apr, pp 39-42.

[18] D. B. Brown, "A computerized algorithm for determining the reliability of redundant configurations," IEEE Trans. Reliability, vol R-20, 1971 Aug, pp 121-124.

[19] J. L. Burris, Model for the Analysis of Probabilities of Systems, MBA, Research Report, Department of Administrative Sciences, Oklahoma State University, 1972.

[20] Y. D. Burtin, B. G. Pittel, "Asymptotic estimates of the reliability of a complex system," Engineering Cybernetics, no 3, 1972, pp 445-451.

[21] T. Case, "A reduction technique for obtaining a simplified reliability expression," IEEE Trans. Reliability, vol R-26, 1977 Oct, pp 248-249.

[22] W. K. Chung, W. C. Chan, "A new approach to reliability prediction," IEEE Trans. Reliability, vol R-23, 1974 Oct, pp 252-255.

[23] J. W. Cooley, Simulation Program for Assessing the Reliabilities of Complex Systems (SPARCS), Ph.D., Dissertation, Oklahoma State University, 1976.

[24] S. Devamanoharan, "A note on 'determination of tiesets and cutsets for a system without feedback'," IEEE Trans. Reliability, vol R-20, 1979 Apr, pp 67-69.

[25] J. D. Esary, F. Proschan, "Coherent structures of non-identical components," Technometrics, vol 5, 1963 May, pp 191-209.

[26] W. Feller, An Introduction to Probability Theory and Its Applications, 3rd. ed., New York: Wiley, vol 1, 1968.

[27] H. Frank, I. T. Frisch, Communication, Transmission and Transportation Networks, Reading, Mass.: Addison-Wesley, 1971.

[28] L. Fratta, U. G. Montanari, "A boolean algebra method for computing the terminal reliability in a communication network," IEEE Trans. Circuit Theory, vol CT-20, 1973 May, pp 203-211.

[29] L. B. Groysberg, M. D. Lindenbaum, "Calculation of the reliability of hierarchical systems with majority redundancy," Engineering Cybernetics, no 5, 1972, pp 819-827.

[30] H. Gupta, J. Sharma, "A delta-star transformation approach for reliability evaluation," IEEE Trans. Reliability, vol R-27, 1978 Aug, pp 212-214.

[31] H. Gupta, J. Sharma, "A method of symbolic steady-state availability evaluation of K-out-of-n; G system," IEEE Trans. Reliability, vol R-28, 1979 Apr, pp 56-57.

[32] E. Hansler, G. K. McAuliffe, R. S. Wilkov, "Exact calculation of computer network reliability," Networks, vol 4, 1974, pp 95-112.

[33] F. J. Henley, R. A. Williams, Graph Theory in Modern Engineering, New York: Academic Press, 1973.

[34] R. B. Hurley, "Probability maps," IEEE Trans. Reliability, 1963 Sep, pp 39-44.

[35] R. K. Iyer, T. Downs, "A moment approach to evaluation and optimization of complex system reliability," IEEE Trans Reliability, vol R-27, 1978 Aug, pp 226-229.

[36] P. A. Jensen, M. Bellmore, "An algorithm to determine the reliability of a complex system," IEEE Trans. Reliability, vol R-18, 1969 Nov, pp 169-174.

[37] Y. H. Kim, K. E. Case, P. M. Ghare, "A method for computing complex system reliability," IEEE Trans. Reliability, vol R-21, 1972 Nov, pp 215-219.

[38] E. V. Krishnamurthy, G. Komissar, "Computer-aided reliability analysis of complicated networks," IEEE Trans. Reliability, vol R-21, 1972 May, pp 86-89.

[39] C. Y. Lee, "Analysis of switching networks," The Bell System Technical Journal 1955 Nov, pp 1287-1315.

[40] K. K. Lee, Rejection Methods for Generating Random Deviates and Their Application in System Reliability, MS Dissertation, Oklahoma State University, 1977.

[41] P. M. Lin, B. J. Leon, T. C. Huang, "A new algorithm for symbolic system reliability analysis," IEEE Trans. Reliability, vol R-25, 1976 Apr, pp 2-15.

[42] M. O. Locks, "The maximum error in system reliability calculations by using a subset of the minimal states," IEEE Trans. Reliability, vol R-20, 1971 Nov, pp 231-234.

[43] M. O. Locks, Monte Carlo Bayesian System Reliability and MTBF-Confidence Assessment, Air Force Fight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, AFFDL-TR-75-144, 1975.

[44] M. L. Locks, "System reliability analysis: a tutorial," Microelectronics and Reliability, vol 18, 1978, pp 335-345.

[45] M. P. Marcus, Switching Circuits for Engineers, Prentice Hall, New Jersey, 1962.

[46] M. Messinger, M. L. Shooman, "Reliability approximations for complex structures," Proc. IEEE Ann. Symp. on Reliability, New York, 1967, pp 292-304.

[47] K. B. Misra, T. S. M. Rao, "Reliability analysis of redundant networks using flow graphs," IEEE Trans. Reliability, vol R-19, 1970 Feb, pp 19-24.

[48] K. B. Misra, "An algorithm for the reliability evaluation of redundant networks," IEEE Trans. Reliability, vol R-19, 1970 Nov, pp 146-151.

[49] F. Moskowitz, "The analysis of redundancy networks," AIEE Trans. Commun. Electron., vol 77, 1958 Nov, pp 627-632.

[50] H. Nakazawa, "Bayesian decomposition method for computing the reliability of an oriented network," IEEE Trans. Reliability, vol R-25, 1976 June, pp 77-80.

[51] H. Nakazawa, "A decomposition method for computing system reliability by a Boolean expression," IEEE Trans. Reliability, vol R-26, no 4, 1977 Oct, pp 250-252.

[52] A. C. Nelson, Jr., J. R. Batts, R. L. Beadles, "A computer program for approximating system reliability," IEEE Trans. Reliability, vol R-19, 1970 May, pp 61-65.

[53] A. Papoulis, Probability, Random Variables and Stochastic Processes, McGraw-Hill, New York, 1965.

[54] A. F. Premo, Jr., "The use of Boolean algebra and a truth table in the formulation of a mathematical model of success," IEEE Trans. Reliability, 1963 Sep, pp 45-49.

[55] S. Rai, K. K. Aggarwal, "An efficient method for reliability evaluation of a general network," IEEE Trans. Reliability, vol R-27, 1978 Aug, pp 206-211.

[56] M. Ramamoorty, Balgopal, "Block diagram approach to power system reliability," IEEE Trans. PAS, vol PAS-89, 1970 May/June, pp 802-811.

[57] A. Rosenthal, "Approaches to comparing cut-set enumeration algorithms," IEEE Trans. Reliability, vol R-28, 1979 Apr, pp 62-66.

[58] W. G. Schneeweiss, "Calculating the probability of Boolean expression being 1," IEEE Trans. Reliability, vol R-26, no 1, 1977 Apr, pp 16-22.

[59] S. Seshu and M. B. Reed, Linear Graphs and Electrical Networks, Addison Wesley, New York, 1961.

[60] J. Sharma, "Algorithm for reliability evaluation of a reducible network," IEEE Trans. Reliability, vol R-25, 1976 Dec, pp 337-339.

[61] A. W. Shogan, "A recursive algorithm for bounding network reliability," IEEE Trans. Reliability, vol R-26, no 5, 1977 Dec.

[62] M. L. Shooman, Probabilistic Reliability: An Engineering Approach, McGraw-Hill, New York, 1968.

[63] C. Singh, R. Billington, System Reliability Modelling and Evaluation, Hutchinson & Co., 1977.

[64] S. S. Tung, "Reliability of a tree network," IEEE Trans. Reliability, vol R-25, 1976 Dec, pp 333-336.

[65] S. S. Yau, Y. S. Tang, "An efficient algorithm for generating complete test sets for combinational logic circuits," IEEE Trans. Computer, vol C-20, 1971 Nov, pp 1245-1251.

[66] G. E. Whitehouse, "GERT, a useful technique for analyzing reliability problems," Technometrics, vol 12, no 1, 1970 Feb.

[67] R. S. Wilkov, "Analysis and design of reliable computer networks," IEEE Trans. Communications, vol com-20, 1972 Jun, pp 660-678.

[68] F. A. Tillman, C. L. Hwang, W. Kuo, "Optimization techniques for system reliability with redundancy - a review," IEEE Trans. on Reliability, vol R-26, 1977, pp 148-155.

[69] C. H. Lie, C. L. Hwang, F. A. Tillman, "Availability of maintained systems: a state-of-the-art survey," AIEE Trans., vol 9, 1977, pp 247-259.

[70] F. A. Tillman, C. L. Hwang, W. Kuo, "System effectiveness models: an annotated bibliography," IEEE Trans. of Reliability (in press, 1980).

[71] R. E. Barlow, F. Proschan, Statistical Theory of Reliability and Life Testing Probability Models, New York: Holt, Rinehart and Winston, Inc., 1975.

[72] T. B. Boffey, R. J. M. Waters, "Calculation of system reliability by algebraic manipulation of probability expressions," IEEE Trans. Reliability, vol R-28, 1979 Dec, pp 358-363.

[73] J. deMercado, N. Spyratos, B. A. Bowen, "A method for calculation of network reliability," IEEE Trans. Reliability, vol R-25, 1976 Jun, pp 71-76.

[74] J. M. Hammersley, D. C. Handscomb, Monte Carlo Method, London: Methuen and Co. Ltd., 1967.

[75] Y. Hatoyama, "Reliability analysis of 3-state systems," IEEE Trans. Reliability, vol R-28, 1979 Dec, pp 386-393.

[76] S. J. Kamat, M. W. Riley, "Determination of reliability using event-based Monte Carlo simulation," IEEE Trans. Reliability, vol R-24, 1975 Apr, pp 73-75.

[77] S. J. Kamat, W. E. Franzmeier, "Determination of reliability using event-based Monte Carlo simulation - Part II," IEEE Trans. Reliability, vol R-25, 1976 Oct, pp 254-255.

[78] H. Kumamoto, K. Tanaka, K. Inoue, "Efficient evaluation of system reliability by Monte Carlo method," IEEE Trans. Reliability, vol R-26, 1977 Dec, pp 311-315.

[79] M. O. Locks, "Evaluating the KTI Monte Carlo method for system reliability calculations," IEEE Trans. Reliability, vol R-28, 1979 Dec, pp 368-372.

[80] M. O. Locks, "Inverting and minimizing Boolean functions, minimal paths and minimal cuts: noncoherent system analysis," IEEE Trans. Reliability, vol R-28, 1979 Dec, pp 373-375.

[81] M. Mazumdar, "Importance sampling in reliability estimation," in Barlow, Fussell, Singpurwalla (Eds), Reliability and Fault Tree Analysis. Philadelphia: Society for Industrial and Applied Mathematics, 1975, pp 153-163.

[82] H. Aakazawa, "Equivalence of a nonoriented line and a pair of oriented lines in a network," IEEE Trans. Reliability, vol. R-28, 1979 Dec, pp 364-367.

[83] K. Nakashima, Y. Hattori, "An efficient bottom-up algorithm for enumerating minimal cut sets of fault trees," IEEE Trans. Reliability, vol R-28, 1979 Dec, pp 353-357.

[84] F. A. Tillman, C. L. Hwang, L. T. Fan, K. C. Lai, "Optimal reliability of a complex system," IEEE Trans. Reliability, vol R-19, 1970 Aug, pp 95-100.

[85] Bernardo, J. J. and J. M. Blin, "A Programming Model of Consumer Choice among Multi-Attributed Brand," Journal of Consumer Research, Vol. 4, No. 2, pp. 111-118, 1977.

[86] Churchman, C. W. and R. L. Ackoff, "An Approximate Measure of Value," Journal of the Operations Research Society of America, Vol. 2, No. 2, pp. 172-187, 1954.

[87] Fishburn, P. C., "Independence in Utility Theory with Whole Product Sets," Operations Research, Vol. 13, No. 1, pp. 28-45, 1965.

[88] Hwang, C. L., A. S. M. Masud, in collaboration with S. R. Paidy and K. Yoon, Multiple Objective Decision Making-Methods and Applications, A State-of-the Art Survey, Springer-Verlag, Berlin /Heidelberg/New York, 1979.

[89] Hwang, C. L., and K. Yoon, Methods and Applications of Multiple Attribute Decision Making, a research monograph under preparation, 1980.

[90] MacCrimmon, K. R., "Improving the System Design and Evaluation Process by the Use of Trade-off Information: An Application to Northeast Corridor Transportation Planning," RAND Memorandum, RM-5877-DOT, 1969.

[91] MacCrimmon, K. R., "An Overview of Multiple Objective Decision Making," in J. L. Cochrane and M. Zeleny (eds.), Multiple Criteria Decision Making, pp. 18-44, University of South Carolina Press, Columbia, S. C., 1973.

[92] Nijkamp, P., "A Multicriteria Analysis for Project Evaluation: Economic-Ecological Evaluation of a Land Reclamation Project," Papers of the Regional Science Association, Vol. 35, pp. 87-111, 1974.

[93] Nijkamp, P. and A. van Delft, Multi-Criteria Analysis and Regional Decision-Making, Martinus Nijhoff Social Sciences Division, Leiden, the Netherlands, 1977.

[94] Pardee, F. S. et al., "Measurement and Evaluation of Transportation System Effectiveness," RAND Memorandum, RM-5869 DOT, 1969.

[95] Pennington, R. H., Introducing Computer Methods and Numerical Analysis, The McMillan Company, New York, (1965).

[96] Roy, B., "Problems and Methods with Multiple Objective Functions," Mathematical Programming, Vol. 1, No. 2, pp. 239-266, 1971.

[97] Roy, B., "How Outranking Relation Helps Multiple Criteria Decision Making," in Cochrane, J. L. and M. Zeleny (eds.), Multiple Criteria Decision Making, University of South Carolina press, Columbia, South Carolina, pp. 179-201, 1973.

APPENDIX A1

The appendix contains:

1. The computer program which provides bounds for approximating large system reliability
2. The outputs of the program for approximating system reliability of the reliability graph given in Fig. 4.1.1(b), which consist of the success paths, the cutsets, and a series of upper and lower reliability bounds.

To become familiar with the input procedure, refer to Section 4.1.

```
C     ***   BOUNDS FOR RELIABILITY    ***

C      *    ANALYSIS OF PATHS AND CUTS  *

      DIMENSION IP(100,20),IC(20,20)
      DIMENSION IUN(20),IB(20),BOUND(20),PROB(20),IO(20)
    1 READ (5,1500,END=999)N
C     ZERO ARRAYS
      DO 402 I=1,20
      IUN(I)= 0
      IB(I) = 0
      BOUND(I) = 0.
      PROB(I) = 0.
      DO 401 J=1,20
      IC(J,I) = 0
  401 IP(J,I) = 0
      DO 402 J=21,100
  402 IP(J,I) = 0
      READ 1510, (PROB(I),I=1,N), EPSLON
      PRINT 1000,N,(I,PROB(I),I=1,N)
      CALL PATH(N,NP,IP)
      PRINT 1010,NP
      DO 450 I=1,NP
      DO 400 J=1,20
  400 IUN(J) = 0
      DO 430 J=1,20
      K = IP(I,J)
      IF( K )430,430,410
  410 IF(K-25)420,430,430
  420 IUN(K) = 1
  430 CONTINUE
      DO 440 J=1,20
  440 IP(I,J) = IUN(J)
  450 CONTINUE
      DO 4 I=1,NP
      K=0
      DO 3 J=1,N
      IF(IP(I,J))3,3,2
    2 K=K+1
      IO(K)=J
    3 CONTINUE
      PRINT 1020,I,(IO(J),J=1,K)
    4 CONTINUE
      DO 10 I=1,20
      DO 10 J=1,20
   10 IC(I,J) = 0

C
C  ***   DETERMINE SYSTEM CUTS
C  ***   CHECK FOR SINGLE ELEMENT CUTS
C

      K=1
      DO 30 J=1,N
      DO 20 I=1,NP
      IF(IP(I,J))30,30,20
   20 CONTINUE
      IC(K,J) = 1
```

```
      K = K + 1
   30 CONTINUE
C    ***   CHECK FOR DOUBLE ELEMENT CUTS
C
      N1=N-1
      IF(N1)571,571,31
   31 DO 90 I=1,N1
      I1=I+1
      DO 90 J=I1,N
      IDUM=0
      DO 40 L=1,NP
      ITRICK=MINO(IP(L,I)+IP(L,J),1)
      IDUM=IDUM+ITRICK
   40 CONTINUE
      IF(IDUM-NP) 90,50,90
   50 IC(K,I)=1
      IC(K,J)=1
      K1=K-1
      IF(K1)71,71,51
   51 DO 70 L=1,K1
      DO 60 M=1,N
      IDUM=MINO(IC(K,M)*IC(L,M),1)
      IF(IDUM)60,60,80
   60 CONTINUE
   70 CONTINUE
   71 K=K+1
      GO TO 90
   80 IC(K,I)=0
      IC(K,J)=0
   90 CONTINUE
C    ***   CHECK FOR TRIPLE ELEMENT CUTS THAT ARE MINIMAL
C
      N2=N-2
      IF(N2)571,571,91
   91 DO 180 I=1,N2
      I1=I+1
      DO 180 J=I1,N1
      I2=J+1
      DO 180 L=I2,N
      IDUM=0
      DO 120 M=1,NP
      ITRICK=MINO((IP(M,I)+IP(M,J))+IP(M,L),1)
      IDUM=IDUM+ITRICK
  120 CONTINUE
      IF(IDUM-NP)180,130,180
  130 DO 135 II=1,N
  135 IC(K,II) = 0
      IC(K,I)=1
      IC(K,J)=1
      IC(K,L)=1
      K1=K-1
      IF(K1)171,171,131
  131 DO 170 M=1,K1
      IDUM=0
      JDUM=0
      DO 140 IJ=1,N
```

```
      ITRICK=MINO(IC(M,IJ)*IC(K,IJ),1)
      IDUM=IDUM+ITRICK
      JDUM=JDUM+IC(M,IJ)
  140 CONTINUE
      IF(IDUM-JDUM)170,150,170
  150 IC(K,I)=0
      IC(K,J)=0
      IC(K,L)=0
      GO TO 180
  170 CONTINUE
  171 K = K+1
  180 CONTINUE

C *** CHECK FOR FOUR ELEMENT CUTS
C

      N3=N-3
      IF(N3)571,571,181
  181 DO 510 I=1,N3
      I1=I+1
      DO 510 J=I1,N2
      I2=J+1
      DO 510 KK=I2,N1
      I3=KK+1
      DO 510 L=I3,N
      IDUM=0
      DO 460 M=1,NP
      ITRICK=MINO((IP(M,I)+IP(M,J))+(IP(M,KK)+IP(M,L)),1)
      IDUM=IDUM+ITRICK
  460 CONTINUE
      IF(IDUM-NP)510,470,510
  470 IC(K,I)=1
      IC(K,J)=1
      IC(K,L)=1
      IC(K,KK)=1
      K1=K-1
      IF(K1)501,501,471
  471 DO 500 M=1,K1
      IDUM=0
      JDUM=0
      DO 480 IJ=1,N
      ITRICK=MINO(IC(M,IJ)*IC(K,IJ),1)
      IDUM=IDUM+ITRICK
      JDUM=JDUM+IC(M,IJ)
  480 CONTINUE
      IF(IDUM-JDUM)500,490,500
  490 IC(K,I)=0
      IC(K,KK)=0
      IC(K,J)=0
      IC(K,L)=0
      GO TO 510
  500 CONTINUE
  501 K=K+1
  510 CONTINUE

C *** CHECK FOR FIVE ELEMENT CUTS
C

      N4=N-4
      IF(N4)571,571,511
```

```
  511 DO 570 I=1,N4
      I1=I+1
      DO 570 J=I1,N3
      I2=J+1
      DO 570 KK=I2,N2
      I3=KK+1
      DO 570 L=I3,N1
      I4=L+1
      DO 570 MM=I4,N
      IDUM=0
      DO 520 M=1,NP
      ITRICK=MINO(((IP(M,I)+IP(M,J))+IP(M,KK))+(IP(M,L)+
     1IP(M,MM)),1)
      IDUM=IDUM+ITRICK
  520 CONTINUE
      IF(IDUM-NP)570,530,570
  530 IC(K,I)=1
      IC(K,J)=1
      IC(K,L)=1
      IC(K,KK)=1
      IC(K,MM)=1
      K1=K-1
      IF(K1)561,561,531
  531 DO 560 M=1,K1
      IDUM=0
      JDUM=0
      DO 540 IJ=1,N
      ITRICK=MINO(IC(M,IJ)*IC(K,IJ),1)
      IDUM=IDUM+ITRICK
      JDUM=JDUM+IC(M,IJ)
  540 CONTINUE
      IF(IDUM-JDUM)560,550,560
  550 IC(K,I)=0
      IC(K,J)=0
      IC(K,L)=0
      IC(K,KK)=0
      IC(K,MM)=0
      GO TO 570
  560 CONTINUE
  561 K=K+1
  570 CONTINUE

C
C  ***  ALL MINIMAL CUTS FIFTH ORDER OR LESS HAVE BEEN
C                       DETERMINED  ***
C

  571 NC = K - 1
      PRINT 1030 ,NC
      DO 195 I=1,NC
      K=0
      DO 190 J=1,N
      IF(IC(I,J))190,190,185
  185 K=K+1
      IO(K)=J
  190 CONTINUE
      PRINT 1020,I,(IO(J),J=1,K)
  195 CONTINUE
      SYSBD = 1.0
      DO 370 I=1,NC
```

```
      IM1 = I-1
      IF(IM1)201,201,196
  196 DO 200 K=1,IM1
  200 IB(K) = K
  201 NM = INCOE(NC,I)
      IJ = IM1
      BOUND(I) = 0.0
      IF(NM)321,321,202
  202 DO 320 J=1,NM
      K = 1
      IF(IJ-NC)250,210,210
  210 IK = I-K
      IF(IB(IK)-(NC-K))230,220,220
  220 K = K+1
      GO TO 210
  230 IB(IK) = IB(IK) + 1
      IF(I-IK)241,231,231
  231 DO 240 JI=IK,I
  240 IB(JI+1) = IB(JI) + 1
  241 IJ = IB(I-1) + 1
      GO TO 260
  250 IJ = IJ + 1
  260 DO 270 JI= 1,N
  270 IUN(JI) = IC(IJ,JI)
      IF(IM1)291,291,271
  271 DO 290 JI=1,IM1
      N1 = IB(JI)
      DO 280 IK=1,N
  280 IUN(IK) =MINO(IUN(IK)+IC(N1,IK),1)
  290 CONTINUE
  291 PR = 1.0

C    ***  CALCULATE EVENT PROBABILITY
C

      DO 310 IK=1,N
      IF(IUN(IK))310,310,300
  300 PR = PR*(1.0-PROB(IK))
  310 CONTINUE
      BOUND(I) = BOUND(I) + PR
  320 CONTINUE


C    ***   PRINT BOUNDS (UPPER OR LOWER) AND TEST FOR
C                        CCNVERGENCE  ***
C

  321 II = MOD(I,2) + 1
      SYSBD = SYSBD + BCUND(I)*(-1.0)**(II-1)
      GO TO (340,330),II
  330 PRINT 2010,SYSBD,BOUND(I)
      GO TO 350
  340 PRINT 2020,SYSBD,BCUNC(I)
  350 IF(I-1)370,370,360
  360 IF(ABS(BOUND(I)-BOUNC(I-1))-EPSLON)1,1,370
  370 CONTINUE
      PRINT 2030,SYSBD
  380 GO TO 1
 1000 FORMAT(57H-  B O U N D S   F O R   S Y S T E M   R E L I
```

```
     1A B I L I T Y//7X,16HCIRCUIT CONTAINS,I3,9H ELEMENTS//
     211X,7HELEMENT,15X,11HPROBABILITY/11X,7HNUMBER,16X,11HOF
     3  SUCCESS//(I15,F26.4))
1010 FORMAT(///7X,27HTIE SETS OR SUCCESS PATHS (,I3,2H )//
     112X,4HPATH,5X,15HELEMENT NUMBERS/)
1020 FORMAT(I15,8X,16I5)
1030 FORMAT(///7X, 9HCUT SETS(,I3,2H ))
2010 FORMAT(16HOLOWER BOUND IS ,E11.5,5X,10HLAST TERM ,E11.5)
2020 FORMAT(16HOUPPER BOUND IS ,E11.5,5X,10HLAST TERM ,E11.5)
2030 FORMAT(20HOSYSTEM RELIABILITY ,E12.5)
1500 FORMAT(10I5)
1510 FORMAT(8E10.4)
 999 STOP
     END
     FUNCTION INCOE(N,M)
     AN=N
     AM=M
     ALN=ALOG(AN)
     ALM=ALOG(AM)
     NM=M-1
     IF(NM)15,15,5
   5 DO 10 K=1,NM
     AK=K
     ALN=ALN+ALOG(AN-AK)
     ALM=ALM+ALOG(AM-AK)
  10 CONTINUE
  15 TERM = EXP(ALN-ALM)
     INCOE = TERM+0.1
     RETURN
     END


     SUBROUTINE PATH (NEQP,IMAX,L)
     DIMENSION ITABLE(25,9),IPRED(9),L(100,20)
     DO 100 I=1,100
     DO 100 J=1,20
 100 L(I,J) = 0
     JB = NEQP + 1
     DO 4 I=1,JB
     READ 2,IACTIV,IPRED
     DO 4 J=1,9
   4 ITABLE(IACTIV,J)=IPRED(J)
     J = 1
     IMAX = 1
     L(1,1) = 25
   6 J = J + 1
     IC = 0
     ICOUNT = 0
     DO 12 I=1,IMAX
     IF(L(I,J-1))7,7,8
   7 IC = IC + 1
     GO TO 12
   8 K = L(I,J-1)
     M = 1
     L(I,J) = ITABLE(K,M)
   9 M = M + 1
     IF(ITABLE(K,M))10,12,10
```

```
10 ICOUNT = ICOUNT + 1
   KBC = IMAX + ICOUNT
   DO 11 KK=1,J
11 L(KBC,KK) = L(I,KK)
   L(KBC,J) = ITABLE(K,M)
   GO TO 9
12 CONTINUE
   IF( IC-IMAX )13,14,14
13 IMAX = IMAX + ICOUNT
   GO TO 6
14 RETURN
 2 FORMAT(10I5)
   END
```

```
B O U N D S   F O R   S Y S T E M   R E L I A B I L I T Y

     CIRCUIT CONTAINS  5 ELEMENTS

          ELEMENT                    PROBABILITY
          NUMBER                      OF  SUCCESS

              1                        0.9300
              2                        0.8600
              3                        0.9200
              4                        0.9500
              5                        0.9800


     TIE SETS OR SUCCESS PATHS (   3 )

            PATH       ELEMENT NUMBERS

              1             2     5
              2             1     3     5
              3             1     4     5


     CUT SETS(   3 )
              1             5
              2             1     2
              3             2     3     4

LOWER BOUND IS 0.96964E 00      LAST TERM 0.30360E-01

UPPER BOUND IS 0.96989E 00      LAST TERM 0.24640E-03

LOWER BOUND IS 0.96989E 00      LAST TERM 0.78400E-06

SYSTEM RELIABILITY  0.96989E 00
```

APPENDIX A2

COMPUTER OUTPUT FOR LINEAR ASSIGNMENT METHOD

EXECUTOR. MPS/360 V2-M11

SECTION 2 - COLUMNS

	NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT
	20	P11	LL	.	.20000	.	NONE
	21	P12	LL	.	.	.	NONE
	22	P13	LL	.	.	.	NONE
	23	P14	LL	.	.	.	NONE
	24	P15	LL	.	.10000	.	NONE
	25	P16	LL	.	.10000	.	NONE
	26	P17	LL	.	.10000	.	NONE
	27	P18	BS	.	.25000	.	NONE
	28	P19	BS	1.00000	.25000	.	NONE
	29	P21	LL	.	.	.	NONE
	30	P22	LL	.	.10000	.	NONE
	31	P23	LL	.	.17500	.	NONE
	32	P24	LL	.	.07500	.	NONE
	33	P25	LL	.	.17500	.	NONE
	34	P26	LL	.	.17500	.	NONE
	35	P27	LL	.	.	.	NONE
	36	P28	BS	1.00000	.15000	.	NONE
A	37	P29	LL	.	.15000	.	NONE
	38	P31	LL	.	.	.	NONE
	39	P32	LL	.	.10000	.	NONE
	40	P33	LL	.	.17500	.	NONE
	41	P34	LL	.	.07500	.	NONE
	42	P35	LL	.	.17500	.	NONE
	43	P36	BS	1.00000	.17500	.	NONE
	44	P37	BS	.	.16700	.	NONE
A	45	P38	LL	.	.06700	.	NONE
A	46	P39	LL	.	.06700	.	NONE
	47	P41	BS	.	.15000	.	NONE
A	48	P42	LL	.	.15000	.	NONE
	49	P43	LL	.	.	.	NONE
	50	P44	LL	.	.10000	.	NONE
	51	P45	BS	1.00000	.20000	.	NONE
	52	P46	LL	.	.10000	.	NONE
	53	P47	BS	.	.16700	.	NONE
A	54	P48	LL	.	.06700	.	NONE
A	55	P49	LL	.	.06700	.	NONE
	56	P51	LL	.	.	.	NONE
	57	P52	LL	.	.	.	NONE
	58	P53	LL	.	.	.	NONE
	59	P54	LL	.	.	.	NONE
	60	P55	LL	.	.10000	.	NONE
	61	P56	LL	.	.16700	.	NONE
	62	P57	BS	1.00000	.23300	.	NONE
	63	P58	BS	.	.23300	.	NONE
	64	P59	LL	.	.16700	.	NONE
	65	P61	BS	1.00000	.16700	.	NONE
A	66	P62	LL	.	.16700	.	NONE
A	67	P63	LL	.	.24200	.	NONE
	68	P64	LL	.	.17500	.	NONE

EXECUTOR. MPS/360 V2-M11

NUMBER	.COLUMN.	AT	...ACTIVITY...	..INPUT COST..	..LOWER LIMIT.	..UPPER LIMIT
69	P65	LL	.	.17500	.	NONE
70	P66	LL	.	.07500	.	NONE
71	P67	LL	.	.	.	NONE
72	P68	LL	.	.	.	NONE
73	P69	LL	.	.	.	NONE
74	P71	BS	.	.31700	.	NONE
75	P72	BS	1.00000	.31700	.	NONE
76	P73	LL	.	.16700	.	NONE
77	P74	LL	.	.	.	NONE
78	P75	LL	.	.	.	NONE
79	P76	LL	.	.06700	.	NONE
80	P77	LL	.	.06700	.	NONE
81	P78	LL	.	.06700	.	NONE
82	P79	LL	.	.	.	NONE
83	P81	BS	.	.16700	.	NONE
84	P82	LL	.	.16700	.	NONE
85	P83	BS	1.00000	.24200	.	NONE
86	P84	LL	.	.07500	.	NONE
87	P85	LL	.	.07500	.	NONE
88	P86	LL	.	.14200	.	NONE
89	P87	LL	.	.06700	.	NONE
90	P88	LL	.	.06700	.	NONE
91	P89	LL	.	.	.	NONE
92	P91	LL	.	.	.	NONE
93	P92	LL	.	.	.	NONE
94	P93	LL	.	.	.	NONE
95	P94	BS	1.00000	.50000	.	NONE
96	P95	LL	.	.	.	NONE
97	P96	LL	.	.	.	NONE
98	P97	LL	.	.10000	.	NONE
99	P98	LL	.	.10000	.	NONE
100	P99	BS	.	.30000	.	NONE

SYSTEMS RELIABILITY EVALUATION OF THE COMPLEX AND LARGE SYSTEMS

by

MYOUNG HO LEE

B.S., Seoul National University, Seoul, Korea, 1973
M.B.A., Seoul National University, Seoul, Korea, 1976

AN ABSTRACT OF A MASTER'S THESIS

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1980

The traditional way of calculating the reliability of a system is to break the system into series and parallel subsystems, calculate the subsystem reliabilities, and then combine these through series or parallel formulations to yield the total system reliability. This method is not only unsuitable for computer programming, but also it cannot treat non-series-parallel systems such as those which have bridge type connections. A complex system may be defined as a system which cannot be reduced to a series-parallel system.

The purpose of this thesis is to present a critical review and classification of small to large complex system reliability problems which have been analyzed with various evaluation techniques; to illustrate the theoretical concepts and the practical formulae required to evaluate systems reliability in the analysis and design of system networks; to investigate the computational procedures of each technique and provide an insight into its strengths and weakness; and to use multiple attribute decision making (MADM) methods for determining a suitable system reliability evaluation technique depending upon the size and configuration of the system.

This study analyzes and describes the general techniques related to the evaluation of complex and large system reliability. System models dealt here are small complex, moderate complex and large systems.

A state-of-the-art review of the literature related to system reliability evaluation techniques for the complex and large systems is presented in Chapter 2. Chapter 3 describes the system reliability evaluation techniques, which can evaluate the reliability of small complex and moderate complex system configurations. In Chapter 4, the evaluation methods for the large systems reliability are introduced and a general computer program together with detailed computer diagram is supplemented.

In the above chapters, literature published on system reliability evaluation techniques is classified and critically reviewed. In general, the basic problem is to decide what kind of evaluation technique should be employed depending upon the size and configuration of the system. Chapter 5 demonstrates the decision making process through the applications of the multiple attribute decision making (MADM) methods in the selection of a suitable system reliability evaluation technique for the corresponding system configuration.

All the evaluation techniques employed in the papers surveyed have limited success in solving some large/complex system reliability evaluation problems. Few techniques have been completely effective when applied to large system reliability problems.