1 2

OF/DE

25410

National Library
of Canada

Bibliothèque nationale
du Canada

CANADIAN THESES
ON MICROFICHE

THÈSES CANADIENNES
SUR MICROFICHE

NAME OF AUTHOR/NOM DE L'AUTEUR Nicolas Spyratos

TITLE OF THESIS/TITRE DE LA THÈSE Decomposition Techniques for Computing Reliability
Measures in Large Systems.

UNIVERSITY/UNIVERSITÉ Carleton University

DEGREE FOR WHICH THESIS WAS PRESENTED/
GRADE POUR LEQUEL CETTE THÈSE FUT PRÉSENTÉE Ph.D.

YEAR THIS DEGREE CONFERRED/ANNÉE D'OBTENTION DE CE DEGRÉ 1975

NAME OF SUPERVISOR/NOM DU DIRECTEUR DE THÈSE Professor B.A. Bowen

Permission is hereby granted to the NATIONAL LIBRARY OF
CANADA to microfilm this thesis and to lend or sell copies
of the film.

*L'autorisation est, par la présente, accordée à la BIBLIOTHÈ-
QUE NATIONALE DU CANADA de microfilmer cette thèse et
de prêter ou de vendre des exemplaires du film.*

The author reserves other publication rights, and neither the
thesis nor extensive extracts from it may be printed or other-
wise reproduced without the author's written permission.

*L'auteur se réserve les autres droits de publication; ni la
thèse ni de longs extraits de celle-ci ne doivent être imprimés
ou autrement reproduits sans l'autorisation écrite de l'auteur.*

DATED/DATE June 26, 1975 SIGNED/SIGNÉ [Signature]

PERMANENT ADDRESS/RÉSIDENCE FIXE Nicolas Spyratos

108 Russell Avenue

Ottawa, Ontario K1N 7X1

DECOMPOSITION TECHNIQUES FOR COMPUTING RELIABILITY MEASURES

IN LARGE SYSTEMS

BY

NICOLAS SPYRATOS, DIPL. ING., M.A.Sc.

A thesis submitted to the Faculty of
Graduate Studies in partial fulfilment
of the requirements for the degree of
Doctorate of Philosophy
Department of Systems Engineering
Carleton University
Ottawa, Ontario

June, 1975

© Nicolas Spyros, 1975

THE UNIVERSITY OF CHICAGO
LIBRARY
540 EAST 58TH STREET
CHICAGO, ILL. 60637
TEL: 773-936-5000
FAX: 773-936-5000

The undersigned recommend to the Faculty of Graduate Studies acceptance of this thesis submitted by Nicolas Spyrtos in partial fulfillment of the requirements for the degree of Doctor of Philosophy.

(J.A. Buzacott)

External Examiner

(J. de Mercado)

Co-Supervisor

(Ivan T. Frisch)

External Examiner

(B.A. Bowen)

Co-Supervisor

(J.S. Riordon)

Chairman of Department
of Systems Engineering

ABSTRACT

Enumerating techniques are powerful tools for computing terminal reliability in graphs with non-series-parallel structure. These techniques are based on an enumeration of the paths or the cuts of the graph between a specified pair of nodes. However, the number of these paths and cuts increases much faster than the size of the graph. As a result, the application of enumerating techniques is limited to relatively small graphs.

The thesis presents a new unified method for the calculation of a variety of reliability measures one of which is terminal reliability. The method is based on a decomposition concept and it is presented in the context of terminal reliability. The basic idea is the following: The reliability graph is first decomposed into a number of smaller subgraphs of manageable size; terminal reliability is then obtained from reliability computations in the subgraphs. This makes the method easier to apply to large graphs than other exact or approximate techniques.

A specialized version of the method leads to the development of an algorithm, called "The Decomposition Algorithm". In the context of this algorithm computational complexity is discussed and optimality criteria are presented. An APL implementation showed that the Decomposition Algorithm is a powerful tool for computing terminal reliability in large and complex systems.

It is further shown that the decomposition concept introduced in the thesis applies to any reliability measure which can be viewed as the expectation of a binary random variable defined on the sample space of the reliability graph. This class of measures includes terminal reliability. While principally concerned with the exact computation of the reliability measures in this class, the thesis also examines the possibility of applying the decomposition concept to a larger class. This class includes all reliability measures which can be viewed as expectations of random variables which are not necessarily binary. It is shown that the cumulative distribution functions of these random variables can be readily calculated using the decomposition concept.

ACKNOWLEDGEMENTS

I would like to gratefully acknowledge all of the people whose assistance and support proved invaluable to me in this thesis undertaking.

Special thanks are due to my thesis advisor, Dr. B.A. Bowen, whose timely suggestions were one thing, but whose friendship and guidance sustained me in my work.

Thanks are due also to Dr. J.K. Cavers for many fruitful discussions which took place during the early stages of this work.

To Dr. J. deMercado, I would like to express my gratitude for providing me with the opportunity of summer work with the Department of Communications. During my work there, I benefited a great deal from discussions with Messrs. J. DaSilva, R. Guindon and S. Mahmoud on the practical aspects of computer communications.

The programming assistance of Mr. K. Toth proved also to be a valuable contribution to this work.

The financial support of the National Research Council during my studies and research is also gratefully acknowledged.

TABLE OF CONTENTS

	<u>Page</u>
CHAPTER ONE - INTRODUCTION	1
1.1 Relevance and scope	2
1.2 An outline of the thesis	7
CHAPTER TWO - TERMINAL RELIABILITY	9
2.1 Formulation of the problem	10
2.2 Reliability of series and parallel systems	13
2.3 Enumerating techniques	22
2.4 A conditional probability technique	33
2.5 A transformation technique	38
2.6 Discussion and conclusions	43
CHAPTER THREE - THE 2-PART DECOMPOSITION METHOD	45
3.1 Introduction	46
3.2 Motivation	52
3.3 Mathematical analysis	61
3.4 Applying the decomposition method	71
3.5 A criterion of optimality	78
3.6 Discussion and conclusions	84
CHAPTER FOUR - THE MULTIPART DECOMPOSITION METHOD	87
4.1 The multipart decomposition and its reliability tree	88
4.2 Computing terminal reliability from a multipart decomposition	96
4.3 Computational experience	105
4.4 Discussion and conclusions	118

CHAPTER FIVE - THE GENERALIZED DECOMPOSITION	120
5.1 Introduction	121
5.2 The generalized decomposition method	125
5.3 Applying the generalized decomposition method	132
5.4 Reliability measures as expectations of random variables	140
5.5 Discussion and conclusions	144
CHAPTER SIX - CONCLUDING DISCUSSION	145
6.1 Summary and conclusions	146
6.2 Suggestions for further work	149
APPENDIX - Proofs of theorems	152
REFERENCES	160
INDEX	165

CHAPTER ONE

INTRODUCTION

The problem considered in the thesis and the main results obtained are summarized and discussed. An outline of the thesis is presented.

1.1 RELEVANCE AND SCOPE

In system design one is faced with the problem of interconnecting a number of available components into a configuration which meets specified performance requirements. In the reliability context, the designer is faced with two problems - to design the system for performance and to design it for endurance. Normally, there are several ways to design a system for a specified performance. But the choice is narrowed down when reliability requirements must be met in addition to performance requirements. One specific design will be more reliable than the other possible approaches. Thus, a reliability analysis is necessary before a definite approach is decided upon. Usually, a number of options are available for the reliability of individual components in a certain system configuration, and a cost (money, weight, volume, etc.) is associated with each option. Each possible system configuration is thus associated with a number of reliability allocations on individual components. In order to evaluate the merits and limitations of a specific system configuration one is interested to know

- (1) What is the reliability allocation which provides the required system reliability in minimum overall cost.
- (2) What is the allocation which maximizes system reliability within an imposed overall cost limit.

In order to answer these questions one should, first, select an appropriate measure for system reliability and, second, develop an efficient method for the calculation of this measure in a given system configuration and for a specific

reliability allocation. It is in this context that the research work presented here was carried out.

In its simplest and most general form, reliability is the probability of a system performing its purpose adequately for the period of time intended under the operating conditions encountered. Depending on what is considered adequate performance, a number of reliability measures have been defined. A measure widely used to evaluate system reliability is the so-called terminal reliability. In performing the reliability analysis of a system, its reliability structure is represented by a graph whose nodes and branches represent the various system components. This graph is called the reliability graph of the system. Terminal reliability is the probability that two specific nodes of the reliability graph are joined by a path all the components of which operate satisfactorily.

The calculation of terminal reliability requires the knowledge of good estimates of individual component reliabilities. The measure of a component's reliability is the frequency at which failures occur in time. Reliability distinguishes three characteristic types of failures which may be inherent in the equipment and occur without any fault on the part of the operator. These are: early failures, wear out failures and chance failures. Reliability engineering is concerned with eliminating early failures by appropriate debugging techniques and wear out failures by proper maintenance practices. Its main attention, though, is focused on chance failures and their reduction, because it is the chance failure phenomenon which most undesirably affects equipment reliability in actual operational use - in the period after the

equipment has been debugged and before parts begin to wear out. Thus, the reliability of a well-designed, well-engineered, thoroughly tested and properly maintained component depends on chance failures. This means that the true reliability is never exactly known, that is, the exact numerical value of the probability of adequate performance is not known. However, numerical estimates quite close to this value can be obtained by the use of statistical methods and probability calculations. How close the statistically estimated reliability comes to the true reliability depends on the amount of testing, the completeness of field service reporting all successes and failures and other essential operational data.

When component reliabilities are known, then terminal reliability can be calculated by means of the probability calculus. Under the assumption that each component of the system has two states - one good and one bad - and that component failures are statistically independent, the calculation of terminal reliability is - conceptually - straightforward. The real difficulty encountered here is the complexity of the calculations in large problems. A number of methods exist for the exact or approximate calculation of terminal reliability from component reliabilities, under the assumptions stated above. However, on investigation, one quickly discovers that the state of the art is deficient in handling large systems with the complexity, sophistication and automation inherent in modern technology. In particular:

- (1) Methods of exact calculation are limited by the size and the configuration of the system.
- (2) Methods of approximate calculation, although capable of handling larger systems, produce good bounds only in the high or low

reliability region.

The research work given here helps to solve the difficulties posed by (1) and (2). As such, the thesis is a contribution to reliability analysis and design of large and complex systems.

The thesis presents a new unified method for the calculation of a variety of reliability measures one of which is terminal reliability. The method is based on a decomposition concept and it is presented in the context of terminal reliability. The basic idea is the following: The reliability graph is first decomposed into a number of smaller subgraphs of manageable size; terminal reliability is then obtained from reliability computations in the subgraphs. A detailed presentation of the theory involved is given in sections 3.3 and 5.2. The method is applied, both as a concept and as an analytical tool in chapters 3, 4 and 5; its merits and limitations are summarized in chapter 6.

The thesis is principally concerned with the exact computation of terminal reliability. However, as it is shown in chapter 5, the decomposition concept introduced in the thesis applies to any reliability measure which, like terminal reliability, can be viewed as the expectation of a binary random variable. It is further shown that the same decomposition concept can be used to provide valuable information about reliability measures which belong to a larger class. This class includes all those measures which can be viewed as expectations of random variables which are not necessarily binary. Thus the

decomposition concept introduced in the thesis, extends the capabilities of existing techniques to larger systems and applies to a variety of reliability measures. This is precisely the contribution of the research work given here.

1.2 AN OUTLINE OF THE THESIS

The primary aim of the thesis is to explain, by means of general discussion, analytical development, and examples, new techniques for computing reliability measures in large and complex systems.

Each chapter in the thesis is somewhat self-contained. Generally speaking, the text proceeds in tutorial form from the simple to the complex, defining new terms and concepts as they appear. Theorems are only stated, and interpreted in the context that they are used. Their proofs are given in the appendix.

Chapter 2 discusses the model used in reliability calculations. In the context of this model terminal reliability is defined and known methods for its calculation are briefly reviewed.

Chapter 3 contains a detailed presentation of a 2-part decomposition technique. The technique is presented in the context of terminal reliability. The detailed mathematical analysis is motivated by a simple example. The details of the application of this technique are discussed and a simple rule for reducing computational complexity is derived.

Chapter 4 explains the recursive application of the 2-part decomposition by means of a tree, given the title "The Reliability Tree". This tree provides valuable insight into the structure of the multipart decomposition of a reliability graph. A path-tracing technique is used in conjunction with the reliability tree to produce an algorithm for the exact calculation of

terminal reliability. The computational complexity of this algorithm is discussed and simple rules are derived for reducing this complexity. The results obtained from an APL implementation of the algorithm are discussed. Substantial computational improvements are observed relative to recently published analyses. Considerable improvement is shown to occur as the size of the reliability graph increases.

The decomposition techniques presented in chapters 3 and 4 are viewed as a special case of a generalized decomposition concept, presented in chapter 5, which also discusses the applications of this concept to a variety of reliability measures.

Chapter 6 discusses the merits (and limitations) of the decomposition techniques presented in the thesis and contains suggestions for further work.

CHAPTER TWO

TERMINAL RELIABILITY

A simple model frequently used in reliability calculations is described. Terminal reliability is defined in the context of this model and known methods for its calculation are briefly reviewed.

2.1 FORMULATION OF THE PROBLEM

In performing the reliability analysis of a complex system, it is almost impossible to treat the system in its entirety. The logical approach is to decompose the system into functional entities or components. The subdivision generates a block-diagram description of system operation. Models are then formulated to fit this logical structure, and the calculus of probability is used to compute the system reliability in terms of the component reliabilities. Series and parallel structures often occur, and their reliability can be described very simply. In many cases the structure is of a more complicated nature, and more general techniques are needed.

The formulation of a structural-reliability model can be difficult in a large, sophisticated system and requires much approximation and judgement. This is best done by a system engineer or someone closely associated with him who knows the system operation thoroughly.

It is convenient to represent the structural-reliability model by a graph whose nodes and branches represent components of the system. This graph is called the reliability graph of the system.

Reliability predicts mathematically the system's behaviour under expected operating conditions. More specifically, reliability expresses in numbers the chance of a system operating satisfactorily for a given length of time in an environment for which it was designed. A number of system-reliability measures have been considered depending on the criteria for what is regarded as satisfactory operation of the system. A widely used measure, especially in systems such as computer-communication and power systems, is the so-called terminal reliability.

It is the probability that two specific nodes in the reliability graph, a source node s and a terminal node t , are connected by a path all components of which operate satisfactorily. This probability is denoted by R_{st} .

The decomposition techniques presented in the thesis, are developed in the context of the following problem:

Given a reliability graph with known component reliabilities (estimated for the same period of operation), and two specific nodes, a source node s and a terminal node t , calculate the terminal reliability R_{st} , assuming that:

- (1) Each component has two states, one good and one bad.
- (2) Component failures are statistically independent.

It is assumed that the branches of the reliability graph are not directed, that is, a system component corresponding to a branch can provide connection in both directions (e.g., full duplex communications links in a communication network). It is also assumed that the nodes of the reliability graph are perfectly reliable, that is, the corresponding system components have reliabilities equal to 1. An example of a system where this second assumption is quite reasonable, is a computer-communication network, where computers and software are, in general, much more reliable than their communication channel counterparts. It should be noted that these two additional assumptions are not necessary but have been included only in an effort to simplify the presentation.

It is not true, in general, that each system component has only two states, one good and one bad. Many components may assume three or more states. For example, a resistor may open-circuit, short-circuit, or function properly.

A transistor may exhibit even more than three states if all possible means of failure are included. Not all the failed-component states need represent system failure, and the success or failure of each state must be evaluated in conjunction with the particular circuit and use in question.

The assumption that component failures are statistically independent does not apply to every system. Dependent failures occur in standby systems, interactive systems, and systems which respond to a common environmental change. The models which govern these cases contain time-dependent conditional probabilities.

The assumption that component failures are statistically independent makes possible a combinatorial approach to system reliability. When dependent failures are present, it is generally more convenient to use the Markov model approach.

2.2 RELIABILITY OF SERIES AND PARALLEL SYSTEMS

System reliability calculations are based on two important operations: (1) as precise as possible a measurement of the reliability of the components used in the system environment, and (2) the calculation of the reliability of some complex combination of these components.

When computing terminal reliabilities, in a given reliability graph, component reliabilities are assumed to be known and calculations are then carried out by means of a few comparatively simple probability theorems. The exactness of the results does not hinge on the probability calculations because these are perfectly accurate; rather, it hinges on the exactness of the reliability data of the components.

In terminal reliability calculations we shall use the following basic rules of probability calculus:

1. If A and B are two independent events with probabilities $p(A)$ and $p(B)$, then the probability that both events will occur is the product

$$p(AB) = p(A)p(B) \quad (2.2.1)$$

2. If the two events are mutually exclusive so that when one occurs the other cannot occur, the probability that either A or B will occur is:

$$p(A+B) = p(A)+p(B) \quad (2.2.2)$$

3. If the two events are complementary in addition to being mutually exclusive, that is, if A does not occur B must occur and vice versa, we obtain from (2.2.2)

$$p(A)+p(B) = 1 \quad (2.2.3)$$

Let $r_1, r_2, \dots, r_n$ be the reliabilities of n components, calculated for the same period of operation. Since component survival and failure are two complementary and mutually exclusive events, the unreliabilities of the n components are $q_1 = 1-r_1, q_2 = 1-r_2, \dots, q_n = 1-r_n$. When component failures are assumed statistically independent, the previous equations, translated into the language of reliability, yield the following basic formulas for the calculation of reliability of a combination of $n \geq 2$ components:

1. The probability that both components 1 and 2 will survive is $r_1 r_2$. A repeated application of the argument shows that the probability that all n components will survive is

$$R_s = r_1 r_2 \dots r_n \quad (2.2.4)$$

This equation is often called the product law of reliabilities.

2. The probability that both components 1 and 2 will fail is $q_1 q_2$. A repeated application of the argument shows that the probability that all n components will fail is

$$Q_p = q_1 q_2 \dots q_n \quad (2.2.5)$$

This equation is often called the product law of unreliabilities.

These product laws motivate the following definitions of a series and a parallel system of components:

A series system of components is one in which if any one component fails, the system has failed. Hence, the reliability of a series system of n components is expressed by Equation 2.2.4. According to the definition of the series system, its failure is the complementary event to all n components surviving; thus, the unreliability of a series system of n components is

$$Q_s = 1 - R_s = 1 - r_1 r_2 \dots r_n \quad (2.2.6)$$

A parallel system of components is one which fails only when all its components have failed. Hence, the unreliability of a parallel system of n components is expressed by Equation 2.2.5. According to the definition of the parallel system, its survival is the complementary event to all n components failing; thus, the reliability of a parallel system of n components is

$$R_p = 1 - Q_p = 1 - q_1 q_2 \dots q_n \quad (2.2.7)$$

It follows from Equation 2.2.4, that the reliability of a series system is always worse than the poorest component, and this is generally a disappointment from a reliability standpoint. As an example let us consider a magnetic-core computer memory. The size of the memory is 100×100 or a total of 10^4 cores. The reliability of each memory cell is r . Assuming that all cells are used in

a particular computation and that any cell error is a computational error, the computation reliability is r^{10^4} . If we wish an error rate of 1 in 10^3 computations, we can solve for the required value of r . Letting $q = 1-r$,

$$r^{10^4} = (1-q)^{10^4} = .999$$

Since $q \ll 1$,

$$(1-q)^{10^4} \approx 1 - 10^4 q = .999$$

$$q = 10^{-7}$$

Thus, the failure probability of a memory cell must be four orders of magnitude smaller than the computational error probability.

Related to the parallel system of components is the so-called m -out-of- n system. It is a system of n components which survives, if m out of the n components survive, e.g., a bridge supported by n cables, m of which are necessary to support the maximum load. The parallel system of n components is the special case where $m = 1$. According to its definition, the m -out-of- n system survives if m or more components survive. Thus, if each of the n components is identical, the reliability of the system is

$$R_{m,n} = \sum_{k=m}^n \binom{n}{k} r^k (1-r)^{n-k} \quad (2.2.8)$$

where r is the reliability of any component. If the components all differ, Equation 2.2.8 no longer holds, and one is faced with the explicit enumeration of all possible successful combinations of the n components.

A simple example where all three systems defined above may be considered, is the reliability design of an aircraft electric system. Assume that it is necessary to build a highly reliable power source for the electric system of an aircraft. The maximum electric load to be supplied is assumed to be 10-kw, the average load 7-kw, and the vital functions alone may be run with only 5-kw. Three different generating means are being considered, use of a single 10-kw generator, use of two 5-kw generators with parallel outputs, and use of three 3.5-kw generators with parallel outputs. It is assumed that each generator has a reliability r . Actually, three modes of system operation exist, full rated power, average power, and emergency power. The reliability graphs and the reliabilities corresponding to each mode are given in Table 2.2.9 where R denotes the system reliability. Each of the three systems has advantages and disadvantages with regard to size, weight, cost, and performance, and to arrive at a wise decision, the reliability must also be considered. The five different reliability expressions developed in Table 2.2.9 are compared in Figure 2.2.10. The series configurations are always poorer than a single element. The parallel system of two components is the best. The two-out-of-three system is better than a single element as long as $.5 < r \leq 1.0$ and becomes worse than a single element for $0 \leq r < .5$.

In performing the reliability analysis of a large and complex system it is convenient to replace any subgraph of the reliability graph consisting of n branches in a series or parallel configuration, by a single branch with the

reliability of the subgraph. This operation is usually called series-parallel reduction of the reliability graph and is carried out using the equations 2.2.4 through 2.2.8. When no series-parallel reduction is possible, then the reliability graph is called irreducible. As an example, consider the reliability graph of Figure 2.2.11, where the numbers on the branches are the reliabilities of the corresponding system components and all subgraphs with parallel configuration are of the 1-out-of- n type. Assume the terminal reliability R_{17} is to be calculated. The three parallel branches linking nodes 3 and 4 can be replaced by a single branch whose reliability is calculated - using Equation 2.2.7 - as

$$1 - .10 \times .50 \times .50 = .975$$

This single branch between nodes 3 and 4 is now in series with the branch between nodes 1 and 3, so that a single branch can replace the series-parallel configuration between nodes 1 and 4. The reliability of this branch is calculated - using Equation 2.2.4 - as

$$.40 \times .975 = .39$$

The reliability of the single branch replacing the series-parallel configurations between nodes 2 and 6, is calculated in a similar manner. Thus, repeated series-parallel reduction on the given reliability graph yielded the graph of Figure 2.2.12 which is irreducible with respect to terminal reliability R_{17} .

Note that if terminal reliability R_{16} were to be calculated, the corresponding irreducible graph would be the one shown in Figure 2.2.13.

When series parallel reduction cannot reduce the reliability graph to a single branch, then more general techniques are needed for the calculation of terminal reliability. These techniques - which are briefly reviewed in the following three sections - fall into three groups.

- (1) Enumerating techniques
- (2) Conditional probability techniques
- (3) Transformation techniques.

System	10-kw maximum load mode	7-kw average load mode	5-kw emergency mode
One 10-kw generator g_1	$R = r$	$R = r$	$R = r$
Two 5-kw generators g_1, g_2	$R = r^2$	$R = r^2$	$R = 2r - r^2$
Three 3.5-kw generators g_1, g_2, g_3	$R = r^3$	$R = 3r^2 - 2r^3$	$R = 3r^2 - 2r^3$

Table 2.2.9. Comparison of the reliability of three aircraft electric systems.

Figure 2.2.10. Comparison of several reliability functions for an aircraft electric system.

Figure 2.2.11. A given reliability graph.

Figure 2.2.12. The irreducible reliability graph with respect to R_{17} .

Figure 2.2.13. The irreducible reliability graph with respect to R_{16} .

2.3 ENUMERATING TECHNIQUES

When calculating terminal reliability, each component is assumed to have two states, one good or ON-state and one bad or OFF-state. It is convenient to indicate the ON-state by 1 and the OFF-state by 0. For example, assume that the reliability graph consists of b branches. Then any array of b 0's and 1's will represent a specification of component performances, that is, a state of the reliability graph. Such an array will be referred to as a b -tuple minterm and it will be denoted as x .

The set of all possible minterms of a reliability graph will be denoted as X . It should be realized that the set X is the sample space of the reliability graph, since it consists of all possible minterms (outcomes) of the system. Since component failures are assumed statistically independent, the calculation of the probability $p(x)$ of any minterm (outcome) x , of this sample space, is straightforward: it is the product of the probabilities of occurrence of the individual component states indicated by the minterm x .

The terminal reliability R_{st} was defined as the probability that there is a path in the reliability graph - with all path-components ON - which joins the source node s and the terminal node t . In this context, a state of the reliability graph is favorable if there is an s - t path with all its components ON in this state. A minterm indicating a favorable state is called an applicable minterm (with respect to R_{st}). Thus, R_{st} is the probability of the set of all applicable minterms in X and can be calculated in the following four steps:

- (1) Enumerate all possible minterms of the reliability graph.
- (2) Identify the applicable minterms.
- (3) Calculate the probability of each applicable minterm.
- (4) Sum up the probabilities of all applicable minterms to obtain R_{st} .

The number of possible minterms for a reliability graph consisting of b branches, can be found as follows: each branch has two states and a minterm is formed by considering one state for each of the b branches; hence, the number of possible minterms is

$$\underbrace{2 \times 2 \times \dots \times 2}_{b \text{ times}}$$

that is, their number is 2^b .

It is clear that in order to determine whether a minterm is applicable or not, it is necessary to know all possible paths which can join nodes s and t in the reliability graph. However, knowledge of only the minimal $s-t$ paths is sufficient. A minimal $s-t$ path is a path of the reliability graph which joins nodes s and t and which passes at most once through any given node. Any minterm whose ON components include a minimal $s-t$ path is an applicable minterm.

Another way to identify the applicable minterms in X , is by means of the minimal $s-t$ cuts of the reliability graph. A minimal $s-t$ cut is any minimal set of branches which, when OFF, disrupt all $s-t$ paths. If the OFF components of a minterm include a minimal $s-t$ cut then this minterm is not applicable; otherwise it is an applicable minterm.

As an example consider the reliability graph of Figure 2.3.1 with $b = 5$ branches numbered from 1 to 5. The $2^5 = 32$ possible minterms are listed in Table 2.3.2, numbered from 0 to 31. Suppose that the terminal reliability R_{ad} is to be calculated. In order to identify the minterms which are applicable with respect to R_{ad} , we shall use the minimal s - t paths. Since the reliability graph is small, these paths can be found by inspection and they are:

$$P_1 = \{1, 4\}, \quad P_2 = \{2, 5\}, \quad P_3 = \{2, 3, 5\}, \quad P_4 = \{2, 3, 4\}$$

Any minterm in Table 2.3.2 whose ON components include one of these four paths is an applicable minterm. The applicable minterms of this table are indicated by a 1. In the same table, the probabilities of the applicable minterms have been calculated for the following values of component reliabilities:

$$r_1 = .93, \quad r_2 = .91, \quad r_3 = .93, \quad r_4 = .95, \quad r_5 = .93$$

Note that one could have used the minimal s - t cuts of the reliability graph. These cuts can be obtained by inspection and they are:

$$C_1 = \{1, 2\}, \quad C_2 = \{4, 5\}, \quad C_3 = \{1, 3, 5\}, \quad C_4 = \{2, 3, 4\}$$

Any minterm in Table 2.3.2 whose OFF components include one of these four cuts is not an applicable minterm. The non-applicable minterms of this table are indicated by a 0.

It is interesting to note that the terminal reliability R_{ad} can be viewed as the expectation of a random variable f defined on the set of all minterms of Table 2.3.2 as follows:

$$f(x_i) = 1, \text{ if } x_i \text{ is an applicable minterm}$$

$$= 0, \text{ if } x_i \text{ is not an applicable minterm}$$

where $i = 0, 1, \dots, 31$. Then the expectation of f is calculated as:

$$E(f) = \sum_{i=0}^{31} f(x_i) p(x_i) = R_{ad}$$

This is a very interesting point of view which will be discussed further in Chapter 5.

It should be realized that when the reliability graph is large and complex, explicit enumeration of all possible minterms and then identification of the applicable minterms using the minimal s - t paths or cuts, is out of the question. For, not only does the number of minterms increase exponentially with the number of system components; but also the identification and enumeration of minimal s - t paths or cuts becomes extremely difficult in large and complex graphs. However, a number of efficient techniques are available for carrying out minimal path or cut identification and enumeration. Based on these techniques several methods have been proposed for the exact or the approximate calculation of terminal reliability, without explicit enumeration of all possible minterms.

The exact calculation methods generate the set of applicable minterms by manipulating the set of minimal s - t paths or cuts of the reliability graph. In this context, two powerful techniques have been presented recently. L. Fratta and U.G. Montanari (1973) proposed a method based on path enumeration. Given the set of all minimal s - t paths, the terminal reliability is symbolically computed by transforming a Boolean sum of products into an equivalent form in which all terms are disjoint. On the other hand, E. Hänsler, G.K. McAuliffe and R.S. Wilkov (1974) presented a method based on cut enumeration. A procedure is developed which generates the set of all non-applicable minterms, partitioned into a number of pairwise disjoint subsets, and calculates the probability of each subset. Summing these probabilities one obtains the terminal unreliability. This unreliability is usually denoted as Q_{st} and it is equal to $1-R_{st}$.

The approximate calculation of terminal reliability through path or cut enumeration is based on the bounds obtained by J.D. Esary and F. Proschan (1963) for the terminal reliability in a monotonic system. A system is called monotonic if the following conditions hold:

- (1) The minterm with all its components ON is an applicable minterm.
- (2) The minterm with all its components OFF is not an applicable minterm.
- (3) By turning ON some of the OFF components of an applicable minterm, the minterm remains applicable.

Under these assumptions, suppose that a given reliability graph contains l minimal s - t paths and m minimal s - t cuts. Further, suppose that a_i is the product of the reliabilities of the components in the i -th path and b_j is the product of the unreliabilities of the components in the j -th cut. Then,

$$\prod_{j=1}^m (1 - \alpha_j) \leq R_{st} \leq 1 - \prod_{i=1}^l (1 - \alpha_i) \quad (2.3.3.)$$

The intuitive motivation for these bounds is quite simple. For example, the right-hand bound corresponds to arranging the components of each path to act in series and arranging the resulting path substructures to act in parallel between nodes s and t . Of course the same component may appear in more than one path so that in considering the structure from a physical point of view, it is necessary to suppose some mechanism which causes replications, of the same component, to either all function or all fail. The right-hand bound is the terminal reliability that would result if each linked replication of a component were replaced by an independently performing version of the component, a process which should increase terminal reliability.

The theoretical interest of the bounds is that they offer the means in reliability studies of approximating structures having complex component arrangements with structures having only series and parallel arrangements. The practical interest of the bounds is that they can be useful for structures in which terminal reliability is tedious to evaluate exactly, but whose paths and cuts can be determined by inspection. Such structures are quite numerous. As an example, consider again the irreducible reliability graph of Figure 2.3.1. Using 2.3.3 we can obtain the following bounds for terminal reliability R_{ad} :

$$\begin{aligned} (1 - q_1 q_2)(1 - q_4 q_5)(1 - q_1 q_3 q_5)(1 - q_2 q_3 q_4) &\leq R_{ad} \\ &\leq 1 - [(1 - r_1 r_4)(1 - r_2 r_5)(1 - r_1 r_3 r_5)(1 - r_2 r_3 r_4)] \end{aligned} \quad (2.3.4.)$$

The exact calculation of R_{ad} was carried out in Table 2.3.2 for specific numerical values of the component reliabilities. It was found that $R_{ad} = .989853058$. For the same numerical values, 2.3.4 yields,

$$.9893466 \leq R_{ad} \leq .9993133 \quad (2.3.5.)$$

It should be noted that the lower bound is closer to the exact value of R_{ad} than the upper bound. In general, the lower bound in 2.3.3 offers a good approximation to the exact value of terminal reliability in the high reliability region. The upper bound offers a good approximation in the low reliability region.

It is interesting to note that when component reliabilities are low then the right-hand bound in 2.3.3 can be approximated by an even simpler expression. For, α_i is then very small for all $i = 1, 2, \dots, L$ and thus,

$$\prod_{i=1}^L (1 - \alpha_i) \approx 1 - \sum_{i=1}^L \alpha_i \quad (2.3.6.)$$

Hence, in the low reliability region, a good approximation of R_{st} is:

$$R_{st} \approx \sum_{i=1}^L \alpha_i \quad (2.3.7.)$$

Similarly, when component reliabilities are high, then δ_j is very small for all $j = 1, 2, \dots, m$, and, thus, the left-hand bound in 2.3.3 can be approximated as follows:

$$\prod_{j=1}^m (1 - \delta_j) \approx 1 - \sum_{j=1}^m \delta_j$$

Hence, in the high reliability region, a good approximation of the terminal unreliability Q_{st} is:

$$Q_{st} \approx \sum_{j=1}^m \delta_j \quad (2.3.8.)$$

Applying this approximation to the example of Figure 2.3.1 we obtain

$$Q_{st} = q_1 q_2 + q_4 q_5 + q_1 q_3 q_5 + q_2 q_3 q_4 \quad (2.3.9.)$$

For the (high reliability region) numerical values used in Table 2.3.2 ,

Equation 2.3.9 yields $Q_{st} = .010458$, that is, $R_{st} = .989542$.

This value is very close to the left-hand bound in 2.3.5.

The approximating expressions 2.3.7 and 2.3.8 are very simple and one is only faced with the problem of identifying and enumerating the minimal paths or cuts in a given reliability graph. A method has been developed by S. Okada (1954) and H. Mine (1959) for using the incidence and circuit matrices of a graph to find all paths in order to compute R_{st} . O. Wing and W.H. Kim (1959) studied the properties of the set of all paths between two

specified nodes in an undirected graph. Y. Fu and S.S. Yau (1962) and D.B. Brown (1971) also presented methods to enumerate all paths between a specified pair of nodes. A book by A.T. Berztiss (1971) develops some path-finding algorithms and includes a very comprehensive bibliography. A recent book by E.J. Henley and R.A. Williams (1973) discusses a general method which leads directly to the explicit definition of every path and loop in a graph and gives the result in an easily usable form. L. Fratta and U.G. Montanari (1971) presented a method for determining all minimal paths in a graph by solving a system of linear equations. A procedure given by P.A. Jensen and M. Bellmore (1969) enumerates all minimal s - t cuts in order to compute Q_{st} in an undirected graph. Similar results have been given by A.C. Nelson Jr., J.R. Batts and R.L. Beadles (1970) for directed graphs.

Path and cut enumerating techniques are powerful tools for computing terminal reliability. The exact techniques are time consuming but they apply in the whole reliability region. The approximate techniques are faster but they produce good approximations only in the high or low reliability region. In general, the application of enumerating techniques is limited to relatively small graphs. This is due to the fact that the number of paths and cuts in a graph increases much faster than the size of the graph. As a result, their enumeration in large and complex graphs becomes a difficult task. The aim of the decomposition method presented in this thesis is, precisely, to apply existing enumerating techniques to subgraphs of convenient size, in order to compute the terminal reliability in the original graph.

Minimal a - d paths: $P_1 = \{1, 4\}$, $P_2 = \{2, 5\}$, $P_3 = \{1, 3, 5\}$, $P_4 = \{2, 3, 4\}$.

Minimal a - d cuts: $C_1 = \{1, 2\}$, $C_2 = \{4, 5\}$, $C_3 = \{1, 3, 5\}$, $C_4 = \{2, 3, 4\}$.

Figure 2.3.1. An irreducible reliability graph and its minimal a - d paths and cuts.

i	Minterm x					$p(x)$
	1	2	3	4	5	
0	0	0	0	0	0	0
1	0	0	0	0	1	0
2	0	0	0	1	0	0
3	0	0	0	1	1	0
4	0	0	1	0	0	0
5	0	0	1	0	1	0
6	0	0	1	1	0	0
7	0	0	1	1	1	0
8	0	1	0	0	0	0
9	0	1	0	0	1	1
10	0	1	0	1	0	0
11	0	1	0	1	1	1
12	0	1	1	0	0	0
13	0	1	1	0	1	1
14	0	1	1	1	0	1
15	0	1	1	1	1	1
16	1	0	0	0	0	0
17	1	0	0	0	1	0
18	1	0	0	1	0	1
19	1	0	0	1	1	1
20	1	0	1	0	0	0
21	1	0	1	0	1	1
22	1	0	1	1	0	1
23	1	0	1	1	1	1
24	1	1	0	0	0	0
25	1	1	0	0	1	1
26	1	1	0	1	0	1
27	1	1	0	1	1	1
28	1	1	1	0	0	0
29	1	1	1	0	1	1
30	1	1	1	1	0	1
31	1	1	1	1	1	1

$$\underline{R_{ad} = .9896570580}$$

Table 2.3.2. The calculation of R_{ad} in the graph of Figure 2.3.1. for $r_1=r_3=r_5=.93$, $r_2=.91$, $r_4=.95$

2.4 A CONDITIONAL PROBABILITY TECHNIQUE

The calculation of terminal reliability in irreducible graphs can be simplified using a conditional probability theorem. This theorem states that if B and C are two complementary and mutually exclusive events then the probability of any event A can be written as:

$$p(A) = p(B) \cdot p(A/B) + p(C) \cdot p(A/C) \quad (2.4.1.)$$

That is, the probability of A equals the probability of B , times the probability of A given that the event B has occurred, plus the probability of C , times the probability of A given that the event C has occurred.

As it was explained in the previous section, the set X of all possible minterms of a reliability graph is the sample space of the graph. Consider the λ -th component of the graph and let X_0 denote the set of all those minterms in which the λ -th component is OFF and X_1 the remaining minterms, that is, all those minterms in which the λ -th component is ON. The sets X_0 and X_1 are two complementary and mutually exclusive events in X . Let now A be the set of all applicable minterms in X , with respect to terminal reliability R_{st} . Then $R_{st} = p(A)$ and Equation 2.4.1, translated into the language of reliability, yields:

$$R_{st} = r \cdot [R_{st}(\text{if } r=1)] + (1-r) \cdot [R_{st}(\text{if } r=0)] \quad (2.4.2.)$$

That is, the terminal reliability R_{st} equals to the reliability of a specified component, times the terminal reliability given that the component is ON, plus the unreliability of the said component, times the terminal reliability given that the component is OFF.

Thus, if the reliability graph consists of b branches, Equation 2.4.2 holds for any $\lambda = 1, 2, \dots, b$, that is, it does not depend on the choice of the component. This makes possible the calculation of terminal reliability in irreducible graphs, using only series-parallel reduction. An example, consider the graph of Figure 2.4.3 which is irreducible with respect to terminal reliability R_{ad} (it is the same graph as the one considered in Figure 2.3.1). Let us apply Equation 2.4.2 for component number 3. When $r_3 = 1$ then the given graph takes the form of Figure 2.4.4 and the corresponding terminal reliability is calculated, using series-parallel reduction, as follows:

$$R_{ad}(\text{if } r_3=1) = (1-q_1q_2)(1-q_4q_5) \quad (2.4.6.)$$

When $r_3=0$ then the given graph takes the form of Figure 2.4.5 and the corresponding terminal reliability is calculated as:

$$R_{ad}(\text{if } r_3=0) = 1-(1-r_1r_4)(1-r_2r_5) \quad (2.4.7.)$$

Applying Equation 2.4.2 we obtain

$$R_{ad} = r_3 (1-q_1q_2)(1-q_4q_5) + q_3 [1-(1-r_1r_4)(1-r_2r_5)]$$

K.B. Misra (1970) and K.B. Misra and T.S.M. Rao (1970) developed general algorithms for calculating terminal reliability based on a combination of series-parallel reduction and the iterative use of Equation 2.4.2 for all non-series-parallel branches. These algorithms are very effective for reliability graphs containing few non-series-parallel branches. However, for highly non-series-parallel graphs the required computation time is prohibitively long.

It is interesting to note that equation 2.4.2 can be used for deriving a formula for the sensitivity of terminal reliability with respect to individual component reliability. In a reliability graph with b branches, terminal reliability R_{st} is a function of the reliabilities $r_1, r_2, \dots, r_b$ of the branches. The partial derivative of R_{st} with respect to $r_\lambda, \lambda=1, 2, \dots, b$, is called the sensitivity of R_{st} with respect to the λ -th component. Using Equation 2.4.2 the following expression can be obtained for this sensitivity:

$$\frac{\partial R_{st}}{\partial r} = [R_{st}(\text{if } r = 1)] - [R_{st}(\text{if } r = 0)] \quad (2.4.8.)$$

Thus, the sensitivity of the terminal reliability, in a given graph, with respect to a specified component, can be calculated by computing the terminal reliability in two simpler graphs. These two graphs are obtained from the given graph if the reliability of the specified component is put equal to 1 and then equal to 0. In the example of Figure 2.4.3, the sensitivity of R_{ad} with respect to component number 3 is

$$\frac{\partial R_{ad}}{\partial r_3} = [R_{ad}(\text{if } r_3=1)] - [R_{ad}(\text{if } r_3=0)]$$

and using the results obtained in 2.4.6 and 2.4.7 we obtain

$$\frac{\partial R_{ad}}{\partial r_3} = (1-q_1q_2)(1-q_4q_5) + (1-r_1r_4)(1-r_2r_5) - 1$$

We shall return to the sensitivity of terminal reliability in Chapter 4, where an algorithm for the exact calculation of terminal reliability is presented.

Figure 2.4.3. A reliability graph which is irreducible with respect to R_{ad} .

Figure 2.4.4. The graph of Figure 2.4.3. when $r_b=1$.

Figure 2.4.5. The graph of Figure 2.4.3. when $r_b=0$.

2.5 A TRANSFORMATION TECHNIQUE

When component reliabilities are high, a simple transformation allows calculation of terminal reliability in irreducible graphs to be carried out through series-parallel reduction. The technique was presented by S.K. Banerjee and K. Rajamani (1972) and it is based on delta-to-star conversion formulas, similar to those employed in the analysis of electrical networks.

Let r denote the reliability of a component and q the unreliability of the same component. The following transformation will be used:

$$\sigma = \frac{q}{r} \quad (2.5.1.)$$

The component reliability and unreliability can then be expressed in terms of σ as follows:

$$r = \frac{1}{1+\sigma} \quad (2.5.2.)$$

$$q = \frac{1}{1+\frac{1}{\sigma}}$$

The reliability of a series system of n components is $r_s = r_1 r_2 \dots r_n$ and, using 2.5.2, we obtain:

$$1+\sigma_s = (1+\sigma_1)(1+\sigma_2)\dots(1+\sigma_n)$$

For high component reliabilities σ_i is small, $i = 1, 2, \dots, n$, so that this last equation can be approximated by

$$\sigma_s = \sigma_1 + \sigma_2 + \dots + \sigma_n \quad (2.5.3.)$$

The unreliability of a parallel system of n components is $q_p = q_1 q_2 \dots q_n$ and, using 2.5.2, we obtain

$$\begin{aligned} 1 + \frac{1}{\sigma_p} &= \left(1 + \frac{1}{\sigma_1}\right) \left(1 + \frac{1}{\sigma_2}\right) \dots \left(1 + \frac{1}{\sigma_n}\right) \\ &= \frac{(1 + \sigma_1)(1 + \sigma_2) \dots (1 + \sigma_n)}{\sigma_1 \sigma_2 \dots \sigma_n} \end{aligned}$$

For high component reliabilities σ_p and σ_i are very small, $i = 1, 2, \dots, n$, so that the last equation can be approximated by

$$\begin{aligned} \frac{1}{\sigma_p} &= \frac{1 + (\sigma_1 + \sigma_2 + \dots + \sigma_n)}{\sigma_1 \sigma_2 \dots \sigma_n} \\ &= \frac{1 + \sigma_s}{\sigma_1 \sigma_2 \dots \sigma_n} \end{aligned}$$

This last equation implies that

$$\sigma_p = \frac{\sigma_1 \sigma_2 \dots \sigma_n}{1 + \sigma_s} \quad (2.5.4.)$$

The delta-to-star conversion formulas are obtained using equations 2.5.3 and 2.5.4. In order for the star to be equivalent, in terms of reliability, to the delta structure, the following equations should be true:

$$\left. \begin{aligned} \sigma_1 + \sigma_3 &= \frac{\sigma_{13}(\sigma_{12} + \sigma_{23})}{1 + \sigma_{\Delta}} \\ \sigma_2 + \sigma_1 &= \frac{\sigma_{21}(\sigma_{23} + \sigma_{31})}{1 + \sigma_{\Delta}} \\ \sigma_3 + \sigma_2 &= \frac{\sigma_{32}(\sigma_{31} + \sigma_{12})}{1 + \sigma_{\Delta}} \end{aligned} \right\}$$

Solving these equations for $\sigma_1, \sigma_2, \sigma_3$ we obtain

$$\left. \begin{aligned} \sigma_1 &= \frac{\sigma_{12}\sigma_{13}}{1 + \sigma_{\Delta}} \\ \sigma_2 &= \frac{\sigma_{21}\sigma_{23}}{1 + \sigma_{\Delta}} \\ \sigma_3 &= \frac{\sigma_{31}\sigma_{32}}{1 + \sigma_{\Delta}} \end{aligned} \right\} \quad (2.5.5.)$$

Equations 2.5.5 are the delta-to-star conversion formulas which allow computation of terminal reliability in irreducible graphs to be carried out through series-parallel reduction. As an example, consider the graph of Figure 2.5.6 which is irreducible with respect to the terminal reliability R_{ad} . Component reliabilities and their transforms - as calculated using 2.5.1 - are indicated on the branches of the graph. The given graph is equivalent to the series-parallel graph of Figure 2.5.7, which was obtained by converting the delta structure formed by the nodes b , c , and d into a star, using equations 2.5.5. Using equations 2.5.3 and 2.5.4, the transform σ_{ad} of the terminal reliability R_{ad} , can be readily calculated in this last graph as follows:

$$\sigma_{ad} = \frac{(\sigma_1 + \sigma_b)(\sigma_2 + \sigma_c)}{1 + (\sigma_1 + \sigma_b) + (\sigma_2 + \sigma_c)} + \sigma_d$$

$$= .0293$$

Finally, the terminal reliability R_{ad} is calculated using 2.5.2

$$R_{ad} = \frac{1}{1 + \sigma_{ad}}$$

$$= \frac{1}{1 + .0293}$$

$$= .9716$$

Figure 2.5.6. A reliability graph which is irreducible with respect to R_{ad} .

Figure 2.5.7. The graph obtained from the given graph by transforming the delta bcd into a star.

2.6 DISCUSSION AND CONCLUSIONS

A simple model frequently used in reliability calculations was described and terminal reliability was defined in the context of this model.

When the reliability graph has a series-parallel structure then the calculation of terminal reliability can be carried out extremely fast using series-parallel reduction of the graph. When the graph contains only a few non-series-parallel branches, then a combination of the conditional probability technique and series-parallel reduction is the best approach to the problem.

When the reliability graph is large and irreducible, then enumerating techniques provide powerful tools for calculating terminal reliability. Among these techniques, the ones producing exact solutions to the problem are more sophisticated but they work in the whole reliability region. The approximate techniques are faster and easier to apply but they provide good approximations only in the high or low reliability region. The enumerating techniques are limited by the fact that they are based on the identification and enumeration of all paths or cuts in the reliability graph. The number of these paths and cuts increases much faster than the size of the graph. As a result, the application of enumerating techniques is limited to relatively small graphs.

The transformation technique is another powerful tool which applies to large irreducible graphs but it produces good results only in the high reliability region. Nevertheless, reliability transformation seems to be a promising approach and it should receive more attention in the future.

In pursuing the research presented in this thesis, the aim was to develop a unified method applicable to a variety of reliability measures. In the context of terminal reliability, the method should extend the capability of existing techniques to larger graphs. Also, it should be applicable in the whole reliability region. The strategy was to study the structure of the problem and try to decompose it into a number of subproblems of manageable size. Also, to develop the analytical tools which would produce the solution to the original problem using the solutions obtained in the subproblems. Finally, to formulate optimality criteria which would allow the definition of the subproblems in a way that reduces computational complexity to a minimum. It is in this spirit that the decomposition method is presented in the following chapters.

CHAPTER THREE

THE 2-PART DECOMPOSITION METHOD

The main result of this chapter is a graph-partitioning technique which allows the problem of calculating terminal reliability to be decomposed into two subproblems. The details of the application of this 2-part decomposition are discussed and a simple rule for minimizing computational complexity is derived. A recently published algorithm is shown to be a special case of the 2-part decomposition method. Further, the technique is shown to offer substantial computational savings relative to this algorithm.

3.1 INTRODUCTION

In large irreducible graphs enumerating techniques seem to provide the only means for calculating terminal reliability when component reliabilities range through the whole reliability region. However, the application of these techniques is limited by the fact that the number of paths and cuts increases much faster than the size of the graph.

O. Wing and W.H. Kim (1959) showed that, in an undirected graph with b branches and n nodes, the number of paths between two nodes s and t is 2^{b-n-2} . On the other hand, the number of $s-t$ cuts in this graph can be found as follows: Each cut corresponds to a partition of the set of nodes into two disjoint subsets, one containing node s and the other containing node t . The number of ways in which $n-2$ nodes can be associated with node s (or node t) in order to form such a subset, is

$$\binom{n-2}{0} + \binom{n-2}{1} + \dots + \binom{n-2}{n-1} + \binom{n-2}{n-2}$$

That is, the number of $s-t$ cuts in the graph is on the order of 2^{n-2} . Thus both the number of $s-t$ paths and the number of $s-t$ cuts in a graph grow exponentially as the size of the graph increases. Hence, if existing enumerating techniques are to be applied to large problems, a decomposition of the reliability graph into a number of subgraphs of manageable size, seems to be necessary. The questions which arise in this context are:

- (1) How to decompose the reliability graph.
- (2) How to obtain terminal reliability from reliability calculations in the subgraphs.

The decomposition method proposed in this chapter makes use of graph partitioning through minimal boundaries. A minimal s - t boundary of the reliability graph is a minimal set of nodes whose removal from the graph disrupts all s - t paths. It should be realized that, according to this definition, node s alone is a minimal s - t boundary and so is node t alone.

The mathematical analysis necessary to derive terminal reliability from reliability calculations in the subgraphs, is first developed in the context of a 2-part partition of the reliability graph and it is presented in this chapter. The results obtained are then used to develop the multipart decomposition method presented in the following chapter. Thus, the problem considered in this chapter is to calculate terminal reliability from a 2-part partition of a given reliability graph under the following assumptions:

- (1) Component reliabilities are known.
- (2) Component failures are statistically independent.
- (3) Each component has two states, one good (or ON state) and one bad (or OFF state).

Two additional assumptions are made, namely, that the reliability graph is undirected and that its nodes are perfectly reliable. As it will be explained later in this chapter, these two assumptions can be removed. However, they are included here in an effort to simplify the presentation of the decomposition method.

The mathematical analysis of the decomposition method makes use of indexing through the subsets of a finite set. When the subsets of a finite set A with μ elements are used for indexing purposes then it is convenient to assign a decimal number (indicator) to each subset. A simple way to do this is the following: When forming a specific subset of A , each element of A is assigned a 1 or a 0 depending on whether it is included in the subset or not. If the elements of A are listed in a specific order, then to each subset of A there corresponds a specific binary number with μ digits. The decimal value of this binary number is the decimal indicator assigned to the subset. As an example, consider the set $A = \{a, b, c\}$. Its distinct subsets and the corresponding binary and decimal indicators are shown in Table 3.1.1.

In presenting the decomposition method we shall use certain simple facts from set theory which are summarized here.

- (1) A collection of subsets of a set A is called a partition of A if the subsets in the collection are pairwise disjoint and their union is the set A . As an example consider the set $A = \{a, b, c\}$. The collection $\{\{a\}, \{b, c\}\}$ is a partition of A whereas, the collection $\{\{a\}, \{b\}, \{b, c\}\}$ is not a partition of A .
- (2) The number of distinct subsets of a finite set A with μ elements is 2^μ . This is a consequence of the fact that in forming all subsets of A one has to consider the elements of A choosing 0, 1, 2, ..., $\mu-1$, μ , at a time. Thus, there are

$$\binom{\mu}{0} + \binom{\mu}{1} + \binom{\mu}{2} + \dots + \binom{\mu}{\mu-1} + \binom{\mu}{\mu}$$

distinct subsets of A , that is, their number is 2^r .

The following notation will be used in this and later chapters:

b : The number of branches in the given irreducible reliability graph.

x : A b -tuple minterm indicating the state of the reliability graph.

X : The set of all 2^b minterms x .

N_k : The minimal s - t boundary between part $k-1$ and part k of a graph partition.

n_k : The number of nodes in the set N_k .

$$L_k = 2^{n_k - 1}$$

$S(N_k)$: The collection of the L_k distinct non-empty subsets of the set N_k .

u_k : The decimal indicator of the subsets of the set N_k ; u_k ranges from 0 through to L_k .

$\bar{u}_k$: The decimal indicator of the following subset of N_k : The complement of the subset of N_k with decimal indicator u_k .

b_k : The number of branches in part k of a graph partition.

x^k : The b_k -tuple minterm indicating the state of part k in a graph partition.

X^k : The set of all 2^{b_k} minterms x^k .

$x_{u_{k-1}u_k}^k$: The set of all minterms x^k each with the following property:

The ON components contain paths from at least one node in the subset u_{k-1} of the boundary N_{k-1} , to all nodes in the subset u_k

of the boundary N_k and only to those nodes (i.e., the ON components do not contain a path to any other node in N_k).

$\mathbf{x}^k$: A $2^{b_k} \times L_{k-1}$ matrix called the k -th part matrix; it is defined in section 3.4, and it is used for determining the sets $x_{u_{k-1}u_k}^k$.

$p_{u_{k-1}u_k}^k$: The probability of the set (event) $x_{u_{k-1}u_k}^k$.

ϵ : A symbol denoting membership.

$\cup$: A symbol denoting union of sets.

$\cap$: A symbol denoting intersection of sets.

$\subset$: A symbol denoting set inclusion.

$$A = \{a, b, c\}$$

0 0 0	{ }	0
0 0 1	{ c }	1
0 1 0	{ b }	2
0 1 1	{ b, c }	3
1 0 0	{ a }	4
1 0 1	{ a, c }	5
1 1 0	{ a, b }	6
1 1 1	{ a, b, c }	7

Binary indicator
 Decimal indicator

Table 3.1.1. The distinct subsets of the set $A = \{a, b, c\}$ and the corresponding binary and decimal indicators.

3.2 MOTIVATION

The decomposition technique presented in this chapter applies to the calculation of a variety of reliability measures. However, it is presented in the context of a particular reliability measure, namely, the terminal reliability R_{st} . The reason for doing so is simple: Rather than developing an abstract theory we chose to present the decomposition method in concrete terms. Its general nature will be made clear in chapter 5.

Let us begin with an example of a reliability graph in which the terminal reliability R_{st} is to be calculated. This graph is shown in Figure 3.2.1, where the numbers on the branches are the component reliabilities. Applying series-parallel reduction we obtain the irreducible graph shown in Figure 3.2.2 with seven branches numbered from 1 to 7. Table 3.2.4 is a list of all possible 7-tuple minterms x of this graph. In calculating R_{st} , as described in section 2.3, the next step is to identify the applicable minterms in this table. To do this we need to know all minimal s - t paths (or cuts). Since the graph is small, these paths can be found by inspection. They are:

$$P_1 = \{1, 3, 5, 7\}, \quad P_2 = \{1, 3, 6\}, \quad P_3 = \{1, 4, 5, 6\}, \quad P_4 = \{1, 4, 7\},$$

$$P_5 = \{2, 3, 4, 7\}, \quad P_6 = \{2, 5, 7\}, \quad P_7 = \{2, 6\}.$$

Any minterm in Table 3.2.4 whose ON components contain one of these paths is an applicable minterm. The applicable minterms of the table are indicated by a 1.

Rather than computing the probabilities of the applicable minterms and summing them up to obtain R_{st} , let us make some remarks which are the key to the mathematical analysis of the decomposition method.

First, let us consider a minimal s - t boundary of the irreducible graph, say, the boundary $\{b, c\}$. This boundary defines a partition of the graph into the two parts (subgraphs) 1 and 2 shown in Figure 3.2.3. These two parts have no common branches and are tied through nodes b and c . Corresponding to this partition of the graph each minterm x in Table 3.2.4 is partitioned into two parts x^1 and x^2 ; x^1 is a 4-tuple minterm indicating the state of part 1 and x^2 is a 3-tuple minterm indicating the state of part 2.

Table 3.2.5 is a shorthand notation for the applicable minterms of Table 3.2.4. That is, by concatenating each minterm x^1 of Table 3.2.4 with all minterms x^2 in the corresponding set $A_j^2, j = 1, 2, 3$, we obtain all the applicable minterms x of Table 3.2.4. Further, since some sets A_j^2 are repeated, Table 3.2.5 can be written in the more compact form of Table 3.2.6. In this last table, by concatenating each minterm x^1 in the set A_i^1 with all minterms x^2 in the set $A_i^2, i = 1, 2, 3$, we obtain all the applicable minterms x of Table 3.2.4. Thus, the applicable minterms of Table 3.2.4 can be represented in the very convenient form of Table 3.2.6. In fact, all one has to do now, in order to obtain R_{st} is to compute the sum

$$\sum_{i=1}^3 \left(\sum_{x^1 \in A_i^1} p(x^1) \sum_{x^2 \in A_i^2} p(x^2) \right)$$

The natural question which arises now is: Can one produce Table 3.2.6 without going into the time-consuming procedure of first listing all possible minterms x and then identifying among them the applicable minterms? The answer is yes, provided that one can answer the following questions:

- (1) What is the defining property of the set A_i^1 ? That is, according to which criteria can one select those minterms x^1 which make up the set A_i^1 , $i = 1, 2, 3$?
- (2) What is the defining property of the set A_j^2 ? That is, according to which criteria can one select those minterms x^2 which make up the set A_j^2 , $j = 1, 2, 3$?
- (3) What makes a set A_j^2 "correspond" to a set A_i^1 ?

In answering these questions the key must be the minimal s - t boundary $\{b, c\}$ which defined the minterms x^1 and x^2 . This is, indeed, the case.

Let us consider two sets from Table 3.2.6, say, A_1^1 and A_3^2 . A close examination of the minterms contained in these two sets shows that, with respect to the boundary nodes b and c ,

- (a) The ON components of each minterm $x^1 \in A_1^1$ contain a path joining node s to node c only (i.e., they do not contain an s - b path).
- (b) The ON components of each minterm $x^2 \in A_3^2$ contain a path joining at least one of the boundary nodes b and c to node t .

Thus, the answers to questions (1), (2) and (3) above seem to be the following:

- (1') Given a subset i of the set of boundary nodes, each minterm x^1

whose ON components contain paths joining s to every node in i but to no node outside i , belongs to the set A_i^1 .

(2') Given a subset j of the set of boundary nodes, each minterm x^2 whose ON components contain paths joining at least one node in j to node t , belongs to A_j^2 .

(3') The set A_j^2 corresponds to the set A_j^1 if $i=j$.

These observations constitute a strong motivation for the mathematical analysis presented in the next section.

Figure 3.2.1. A given reliability graph.

Figure 3.2.2. The equivalent irreducible graph.

Figure 3.2.3. A partition of the irreducible graph corresponding to the minimal s - t boundary $\{b, c\}$.

x							
x ¹				x ²			
1	2	3	4	5	6	7	
0	0	0	0	0	0	0	
0	0	0	0	0	0	1	
0	0	0	0	0	1	0	
0	0	0	0	0	1	1	
0	0	0	0	1	0	0	
0	0	0	0	1	0	1	
0	0	0	0	1	1	0	
0	0	0	0	1	1	1	
0	0	0	1	0	0	0	
0	0	0	1	0	0	1	
0	0	0	1	0	1	0	
0	0	0	1	0	1	1	
0	0	0	1	1	0	0	
0	0	0	1	1	0	1	
0	0	0	1	1	1	0	
0	0	0	1	1	1	1	
0	0	1	0	0	0	0	
0	0	1	0	0	0	1	
0	0	1	0	0	1	0	
0	0	1	0	0	1	1	
0	0	1	0	1	0	0	
0	0	1	0	1	0	1	
0	0	1	0	1	1	0	
0	0	1	0	1	1	1	
0	0	1	1	0	0	0	
0	0	1	1	0	0	1	
0	0	1	1	0	1	0	
0	0	1	1	0	1	1	
0	0	1	1	1	0	0	
0	0	1	1	1	0	1	
0	0	1	1	1	1	0	
0	0	1	1	1	1	1	

1	2	3	4	5	6	7	
0	1	0	0	0	0	0	
0	1	0	0	0	0	1	
0	1	0	0	0	1	0	1
0	1	0	0	0	1	1	1
0	1	0	0	1	0	0	
0	1	0	0	1	0	1	1
0	1	0	0	1	1	0	1
0	1	0	0	1	1	1	1
0	1	0	1	0	0	0	
0	1	0	1	0	0	1	
0	1	0	1	0	1	0	1
0	1	0	1	0	1	1	1
0	1	0	1	1	0	0	
0	1	0	1	1	0	1	1
0	1	0	1	1	1	0	1
0	1	0	1	1	1	1	1
0	1	1	0	0	0	0	
0	1	1	0	0	0	1	1
0	1	1	0	0	1	0	1
0	1	1	0	0	1	1	1
0	1	1	0	1	0	0	
0	1	1	0	1	0	1	1
0	1	1	0	1	1	0	1
0	1	1	0	1	1	1	1
0	1	1	1	0	0	0	
0	1	1	1	0	0	1	1
0	1	1	1	0	1	0	1
0	1	1	1	0	1	1	1
0	1	1	1	1	0	0	
0	1	1	1	1	0	1	1
0	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1

(Continued)

1	3	4	5	6	7	
1	0	0	0	0	0	0
1	0	0	0	0	0	1
1	0	0	0	0	1	0
1	0	0	0	0	1	1
1	0	0	0	1	0	0
1	0	0	0	1	0	1
1	0	0	0	1	1	0
1	0	0	0	1	1	1
1	0	0	1	0	0	0
1	0	0	1	0	0	1
1	0	0	1	0	1	0
1	0	0	1	0	1	1
1	0	0	1	1	0	0
1	0	0	1	1	0	1
1	0	0	1	1	1	0
1	0	0	1	1	1	1
1	0	1	0	0	0	0
1	0	1	0	0	0	1
1	0	1	0	0	1	0
1	0	1	0	0	1	1
1	0	1	0	1	0	0
1	0	1	0	1	0	1
1	0	1	0	1	1	0
1	0	1	0	1	1	1
1	0	1	1	0	0	0
1	0	1	1	0	0	1
1	0	1	1	0	1	0
1	0	1	1	0	1	1
1	0	1	1	1	0	0
1	0	1	1	1	0	1
1	0	1	1	1	1	0
1	0	1	1	1	1	1

1	2	3	4	5	6	7	
1	1	0	0	0	0	0	
1	1	0	0	0	0	1	
1	1	0	0	0	1	0	1
1	1	0	0	0	1	1	1
1	1	0	0	1	0	0	
1	1	0	0	1	0	1	1
1	1	0	0	1	1	0	1
1	1	0	0	1	1	1	1
1	1	0	1	0	0	0	
1	1	0	1	0	0	1	1
1	1	0	1	0	1	0	1
1	1	0	1	0	1	1	1
1	1	0	1	1	0	0	
1	1	0	1	1	0	1	1
1	1	0	1	1	1	0	1
1	1	0	1	1	1	1	1
1	1	1	0	0	0	0	
1	1	1	0	0	0	1	
1	1	1	0	0	1	0	1
1	1	1	0	0	1	1	1
1	1	1	0	1	0	0	
1	1	1	0	1	0	1	1
1	1	1	0	1	1	0	1
1	1	1	0	1	1	1	1
1	1	1	1	0	0	0	
1	1	1	1	0	0	1	1
1	1	1	1	0	1	0	1
1	1	1	1	0	1	1	1
1	1	1	1	1	0	0	
1	1	1	1	1	0	1	1
1	1	1	1	1	1	0	1
1	1	1	1	1	1	1	1

Table 3.2.4. The set of all minterms x ; the applicable minterms are indicated by a 1.

x^1				x^2		
1	2	3	4	5	6	7
0	1	0	0	0	1	0
				0	1	1
				1	0	1
				1	1	0
				1	1	1
0	1	0	1	0	1	0
				0	1	1
				1	0	1
				1	1	0
				1	1	1
0	1	1	0	0	1	0
				0	1	1
				1	0	1
				1	1	0
				1	1	1
0	1	1	1	0	0	1
				0	1	0
				0	1	1
				1	0	1
				1	1	0
				1	1	1
1	0	0	1	0	0	1
				0	1	1
				1	0	1
				1	1	0
				1	1	1
1	0	1	0	0	1	0
				0	1	1
				1	0	1
				1	1	0
				1	1	1

A_1^2
 A_1^2
 A_1^2
 A_3^2
 A_2^2
 A_1^2

1	2	3	4	5	6	7
1	0	1	1	0	0	1
				0	1	0
				0	1	1
				1	0	1
				1	1	0
				1	1	1
1	1	0	0	0	1	0
				0	1	1
				1	0	1
				1	1	0
				1	1	1
1	1	0	1	0	0	1
				0	1	0
				0	1	1
				1	0	1
				1	1	0
				1	1	1
1	1	1	0	0	1	0
				0	1	1
				0	1	1
				1	0	1
				1	1	0
				1	1	1

A_3^2
 A_1^2
 A_3^2
 A_1^2
 A_3^2

Table 3.2.5. A shorthand notation for the applicable minterms of Table 3.2.4

x						
x^1				x^2		
1	2	3	4	5	6	7
A_1^1 <div> <div>0 1 0 0</div> <div>0 1 0 1</div> <div>0 1 1 0</div> <div>1 0 1 0</div> <div>1 1 0 0</div> <div>1 1 1 0</div> </div>				A_1^2 <div> <div>0 1 0</div> <div>0 1 1</div> <div>1 0 1</div> <div>1 1 0</div> <div>1 1 1</div> </div>		
A_2^1 <div> <div>1 0 0 1</div> </div>				A_2^2 <div> <div>0 0 1</div> <div>0 1 1</div> <div>1 0 1</div> <div>1 1 0</div> <div>1 1 1</div> </div>		
A_3^1 <div> <div>0 1 1 1</div> <div>1 0 1 1</div> <div>1 1 0 1</div> <div>1 1 1 1</div> </div>				A_3^2 <div> <div>0 0 1</div> <div>0 1 0</div> <div>0 1 1</div> <div>1 0 1</div> <div>1 1 0</div> <div>1 1 1</div> </div>		

Table 3.2.6. A shorthand notation of Table 3.2.5.

3.3 MATHEMATICAL ANALYSIS

The decomposition method is based on an initial partition of the reliability graph into two parts (subgraphs) 1 and 2. The set of boundary nodes is then used to define a partition of the set X^1 of all minterms of part 1. This partition defines, in turn, a partition of the set X of all minterms of the original graph, into a number of pairwise disjoint subsets. Within each of those subsets the applicable minterms of the original graph are identified by examining only the set X^2 of all minterms of part 2. We now proceed to outline the mathematical analysis of the decomposition method, leaving formal proofs for the appendix.

Any minimal s - t boundary defines a partition (decomposition) of the graph into two parts tied together through the boundary nodes. A branch connecting two boundary nodes is called a boundary branch. A boundary branch can be included in either of the two parts of this decomposition, but in only one part, so that the two parts (subgraphs) are disjoint. Two subgraphs of a graph are called disjoint if they have no branches in common. Depending on the arrangement of the boundary branches, a minimal s - t boundary can generate a number of graph partitions. As an example, consider the reliability graph shown in Figure 3.3.12, and suppose that the minimal s - t boundary $\{b, c, d\}$ is used for the partition of the graph. There are two boundary branches corresponding to this boundary, hence four different ways to arrange these branches with respect to a partition of the graph into two parts. The corresponding four different 2-part partitions of the given graph are shown in Figure 3.3.13.

It should be realized that when partitioning the graph, using a minimal s - t boundary, nodes s and t are separated. That is, node s is contained in the one part and node t in the other. The part containing node s will be called part 1 and the part containing node t will be called part 2. Further, let N_1 denote the boundary used to partition the graph into the two parts 1 and 2. Node s alone is a minimal s - t boundary and so is node t alone. Let N_0 and N_2 denote these two boundaries, respectively, that is, let $N_0 = \{s\}$ and $N_2 = \{t\}$. Then part k of the graph partition is contained between the boundaries N_{k-1} and N_k , $k=1, 2$. Naturally, the boundary N_1 is assumed to be different than either N_0 or N_2 .

Let b denote the number of branches in the given reliability graph and b_1, b_2 the number of branches in the parts of a 2-part partition of the graph. It should be clear that $b_1 + b_2 = b$. Corresponding to this partition each b -tuple minterm x of the original graph is partitioned into two parts x^1 and x^2 . Part x^1 is a b_1 -tuple minterm indicating the state of part 1 and part x^2 is a b_2 -tuple minterm indicating the state of part 2. The s^{b_1} minterms x^1 when combined (concatenated) with the s^{b_2} minterms x^2 produce the $s^{b_1} s^{b_2} = s^b$ minterms x , of the original graph.

A partition of the set x^1 is now defined using the collection of all subsets of the boundary N_1 . For convenience we shall indicate each subset by a decimal number (indicator) u_k , as it was explained in section 3.1. The complement of the subset u_k will be denoted as $\bar{u}_k$. Supposing that the boundary N_1 consists of r_k nodes, then the number of distinct subsets of the set N_1 is 2^{r_k} . The decimal indicators of these subsets are $0, 1, \dots, 2^{r_k} - 1$.

For convenience, let $s^{k-1} = \ell_k$. Thus, the decimal indicator u_k ranges from 0 through to ℓ_k . Note that the subset number 1 of the boundary N_0 consists of node s alone.

Definition 3.3.1. For each subset u_j of the boundary N_j define $X_{ju_j}^1$ to be the set of minterms x^1 each with the following property: the ON components contain paths from node s to every node in the set u_j only (i.e., the ON components contain no path to any node in the set $\bar{u}_j$).

It should be clear that every minterm x^1 belongs to some set $X_{ju_j}^1$ for an appropriate choice of the set u_j . Therefore, the union of all sets $X_{ju_j}^1$ contains all the minterms x^1 , that is, it is equal to the set X^1 . Moreover, the sets $X_{ju_j}^1$ can be shown to be pairwise disjoint. Hence, we obtain the following lemma:

Lemma 3.3.2. The collection $\{X_{ju_j}^1 : u_j = 0, 1, \dots, \ell_j\}$ is a partition of the set X^1 .

The minterms x^1 in the set $X_{ju_j}^1$ can be combined (concatenated) with all minterms x^2 in the set X^2 , to produce a set X_{u_j} of minterms x . Using lemma 3.3.2, we can show that as u_j ranges from 0 through to ℓ_j , the corresponding sets X_{u_j} form a partition of the set X of minterms x of the original graph. Hence, we obtain the following theorem:

Theorem 3.3.3. The collection $\{X_{u_j} : u_j = 0, 1, \dots, \ell_j\}$ is a partition of the set X .

Let now $\bar{P}_{u_j}$ denote the probability of the set of applicable minterms contained in the set X_{u_j} . In view of the above theorem, the following expression can be written for the terminal reliability R_{st} .

$$R_{st} = \sum_{u_j=0}^{L_j} P_{u_j} \quad (3.3.4.)$$

It now remains to identify the applicable minterms contained in X_{u_j} and then calculate P_{u_j} . The following theorem permits the applicable minterms in X_{u_j} to be readily identified examining only the minterms of the set X^s .

Theorem 3.3.5. Let $x \in X_{u_j}$ and $\mu_j \leq 8$. Then, x is an applicable minterm with respect to R_{st} , if and only if, part x^s of x has the following property: the ON components contain a path from at least one node in the set u_j to node t .

Let us then group together the minterms x^s which have the property described in the above theorem. Since the subset number j of the boundary N_j consists of node t alone, we shall denote this set as $X_{u_j, j}^s$.

Definition 3.3.6. For each subset u_j of the boundary N_j , define $X_{u_j, j}^s$ to be the set of minterms x^s each with the following property: the ON components contain a path from at least one node in the set u_j to node t .

In view of Theorem 3.3.5, the applicable minterms of the set X_{u_j} can be obtained by concatenating each minterm x^i in the set $X_{u_j, j}^i$, with all the minterms x^s in the set $X_{u_j, j}^s$. Hence, the probability of the set of applicable minterms x contained in X_{u_j} is:

$$P_{u_i} = \sum_{x^1 \in X_{1u_i}^1} p(x^1) \sum_{x^2 \in X_{u_i1}^2} p(x^2) \quad (3.3.7.a)$$

Let $P_{1u_i}^1$ denote the probability of the set $X_{1u_i}^1$ and $P_{u_i1}^2$ denote the probability of the set $X_{u_i1}^2$. Then,

$$P_{1u_i}^1 = \sum_{x^1 \in X_{1u_i}^1} p(x^1) \quad (3.3.8.a)$$

$$P_{u_i1}^2 = \sum_{x^2 \in X_{u_i1}^2} p(x^2) \quad (3.3.8.b)$$

Thus equation 3.3.7.a can be written in the following compact form:

$$P_{u_i} = P_{1u_i}^1 P_{u_i1}^2 \quad (3.3.7.b)$$

It follows immediately from theorem 3.3.5 that there are no applicable minterms in the set X_0 , that is, $P_0 = 0$. Hence, combining equation 3.3.4 with equations 3.3.7.a and 3.3.7.b we obtain the following expressions for R_{st} , respectively.

$$R_{st} = \sum_{u_i=1}^L \left(\sum_{x^1 \in X_{1u_i}^1} p(x^1) \sum_{x^2 \in X_{u_i1}^2} p(x^2) \right) \quad (3.3.9)$$

$$R_{st} = \sum_{u_1=1}^{l_1} P_{1u_1}^1 P_{u_1 t}^2 \quad (3.3.10.a)$$

Let us consider now part 2 of the graph partition and suppose that the nodes in the set u_1 are combined and treated as one node. It follows from definition 3.3.6 that the set $X_{u_1 t}^2$ is precisely the set of applicable minterms with respect to terminal reliability $R_{u_1 t}^2$ calculated in part 2 of the graph partition. Hence, $P_{u_1 t}^2 = R_{u_1 t}^2$ and equation 3.3.10.a can be written as:

$$R_{st} = \sum_{u_1=1}^{l_1} P_{1u_1}^1 R_{u_1 t}^2 \quad (3.3.10.b)$$

Equation 3.3.10 was obtained as a consequence of Theorem 3.3.5. However, this theorem is true only when the boundary N_1 contains 1 or 2 nodes. What happens when there are more than two nodes in this boundary? It is still true that by concatenating every minterm in the set $X_{1u_1}^1$ with every minterm in the set $X_{u_1 t}^2$ we obtain applicable minterms of the set $X_{u_1 t}$. The question, however, is: do we still obtain all the applicable minterms of the set $X_{u_1 t}$? Let us consider an example. The graph of Figure 3.3.14 was partitioned using the boundary $\{a, c, e\}$ which contains 3 nodes. Figure 3.3.15 shows a possible state of the graph. The OFF branches of this state are indicated by broken lines. Since the ON branches contain a path joining s and t , this state of the graph corresponds to an applicable minterm x . It should be clear that although part x^1 of this minterm belongs to the set $X_{1u_1}^1$, part x^2 does not belong to

the set x_{41}^2 . Hence, when putting together the parts x_1^1 of the set x_{14}^1 , with the parts x_2^2 of the set x_{41}^2 , we shall not obtain all the applicable minterms of the set x_4 . This means that when $\mu_1 \geq 3$, then equation 3.3.10 may or may not yield an exact value for the terminal reliability R_{st} , depending on the subgraph configuration of the partition. It would be interesting to know what subgraph configurations correspond to an exact solution of the problem. Let us observe that what made possible the existence of the applicable minterm x considered above, was the presence - in both subgraphs - of nodes which did not belong to a boundary (nodes b and d in Figure 3.3.15). The possibility of such a minterm exists even in the case where only part 2 contains non-boundary nodes, provided that the boundary branches are included in part 1 of the partition. Thus, when $\mu_1 \geq 3$, equation 3.3.10 still yields an exact solution to the problem, provided that the partition of the graph is such that:

- (1) At most one part contains non-boundary nodes.
- (2) If this part is part 2, then the boundary branches - if any - are included in part 2.

For partitions with $\mu_1 \geq 3$ and for which the above conditions are not satisfied, equation 3.3.10 yields a value which is less than or at most equal to the exact value of the terminal reliability. How close to the exact value this approximation is, depends on the values of the component reliabilities and the size and configuration of the graph. Computational experience showed that this approximation is very close to the exact value in systems whose graphs are sparse. Results obtained from such graphs will be discussed in the next chapter.

In summary then, equation 3.3.10 yields the exact value of terminal reliability when there are 1 or 2 nodes in the partitioning boundary. In the case of 3 or more boundary nodes equation 3.3.10 still yields the exact value, provided that the graph partition satisfies conditions (1) and (2) above; otherwise, only an approximate value is obtained. Equation 3.3.10 is the fundamental equation of the 2-part decomposition method and constitutes the basis of the multipart decomposition method to be presented in the next chapter.

An important remark is in order at this point of the presentation. Let us consider a specific 2-part partition of the graph, namely, the partition corresponding to the following minimal s - t boundary: the set of all nodes directly connected to node s . Then each set $X_{u_1}^1$ contains only one minterm x^1 , namely, the minterm whose ON components correspond to the branches connecting node s to the nodes in the set u_1 and whose OFF components correspond to the branches connecting node s to the nodes in the set $\bar{u}_1$. Let r_{1i} denote the reliability of the branch connecting s to node i in the boundary N_1 . Then equation 3.3.10.b takes the form

$$R_{st} = \sum_{u_1=1}^{l_1} \left[\left(\prod_{i \in u_1} r_{1i} \prod_{j \in \bar{u}_1} (1-r_{1j}) \right) R_{u_1, t} \right] \quad (3.3.11.)$$

This equation is the counterpart of the equation derived by E. Hnsler (1972) in the context of terminal unreliability Q_{st} . Thus, a powerful algorithm for computing terminal unreliability, is shown to be readily obtained as a special case of a 2-part partition of the reliability graph.

Figure 3.3.12. A given irreducible reliability graph.

Figure 3.3.13. The four different 2-part partitions of the given graph generated by the minimal s - t boundary $\{b, c, d\}$.

Figure 3.3.14. A 2-part partition of a reliability graph with $\mu_1 = 3$.

Figure 3.3.15. A state of the reliability graph corresponding to an applicable minterm $x \in X_4$ which cannot be obtained using x_{14}^1 and x_{41}^2 .

3.4 APPLYING THE DECOMPOSITION METHOD

The purpose of this section is to illustrate the concepts introduced in the previous section. We shall do so by using the fundamental equation 3.3.10 to calculate the terminal reliability R_{st} in the irreducible graph of Figure 3.4.1. It should be noted that this is the same graph as the one used in section 3.2 to motivate the mathematical analysis of the decomposition method.

Let $S(N_k)$ denote the collection of all distinct non-empty subsets of the set N_k . If there are ℓ_k nodes in the boundary N_k , then there are $2^{\ell_k} - 1 = \ell_k$ subsets in $S(N_k)$ with decimal indicators $1, 2, \dots, \ell_k$. The 2-part partition shown in Figure 3.4.2 was obtained using the minimal s - t boundary $N_1 = \{b, c\}$. It should be clear that $S(N_1) = \{\{c\}, \{b\}, \{b, c\}\}$ with corresponding decimal indicators 1, 2 and 3 that is, $u_1 = 1, 2, 3$.

In order to compute the probabilities $P_{1u_1}^1$ and $P_{u_1 1}^2$, $u_1 = 1, 2, 3$, we need first to know the sets $X_{1u_1}^1$ and $X_{u_1 1}^2$, $u_1 = 1, 2, 3$, and then apply equation 3.3.8. Definitions 3.3.1 and 3.3.6 suggest a unified method to form the sets $X_{u_{k-1}u_k}^k$, corresponding to part k , $k = 1, 2$. This is done by constructing a $2^{b_k} \times \ell_{k-1}$ matrix M^k called the k -th part matrix. This matrix is defined as follows:

- (1) The rows are indexed by the 2^{b_k} minterms x^k of the set X^k and the columns by the ℓ_{k-1} non-empty subsets of the set N_{k-1} , that is, by the elements of $S(N_{k-1})$.

- (2) If a column heading is a single node, then the i -th entry of this column is the binary indicator of that set of nodes in N_k which are connected to the column heading through the ON components of the i -th minterm x^k .
- (3) If the heading of a column j consists of more than one nodes, then the i -th entry of this column is obtained by OR-ing the i -th entries of all columns headed by a single node, and whose headings make up the heading of column j .

It should be clear that in constructing the part matrix M^k , the time-consuming step is determining the columns which are headed by a single node. For, each entry of such a column requires a path-tracing search through the ON components of the corresponding minterm.

Table 3.4.4 contains the part matrices corresponding to the 2-part decomposition of Figure 3.4.2. Since part 1 contains $b_1=4$ branches and $S(N_0)$ contains $\ell_0=1$ element, the part matrix M^1 is a $2^{b_1} \times \ell_0 = 16 \times 1$ matrix. Similarly, the part matrix M^2 is a $2^{b_2} \times \ell_1 = 8 \times 3$ matrix. The entries of the third column of M^2 were obtained by OR-ing the corresponding entries of the first and second column, following rule (3) above.

It is worth noting that when constructing the part matrix M^k , no path-tracing search is needed for the first and last minterm. For, all the components of the first minterm x^k are OFF, hence, no node of the boundary N_{k-1} is connected to any node of the boundary N_k . Similarly, all the components of the last minterm x^k are ON, hence, every node of the boundary N_{k-1} is connected to all nodes of the boundary N_k .

The calculations involved in evaluating the terminal reliability R_{st} , using the 2-part partition of Figure 3.4.2, are summarized in Table 3.4.5. From this table we obtain

$$P_{11}^1 = .1014, \quad P_{12}^1 = .0972, \quad P_{13}^1 = .7506$$

$$P_{11}^2 = .852, \quad P_{21}^2 = .862, \quad P_{31}^2 = .88$$

Applying equation 3.3.10.a, we obtain the terminal reliability R_{st} as follows:

$$R_{st} = .1014 \times .852 + .0972 \times .862 + .7506 \times .88$$

$$= .8307072$$

If we apply equation 3.3.10.b, computation time is considerably reduced. For, as it was explained in the previous section, the probability $P_{u_1}^2$, $u_1=1,2,3$, can be viewed as a terminal reliability in part 2, if all nodes in the set u_1 are combined and treated as one node. Then, $P_{u_1}^2$ is simply the terminal reliability R_{u_1t} and can be calculated from part 2 using any of the existing techniques. Thus, the calculations contained in part 2 of both Table 3.4.4 and 3.4.5 can be replaced by the calculations contained in Table 3.4.6. It should be noted that since all graphs appearing in this table are simple series-parallel graphs, the calculations involved were carried out by inspection.

Figure 3.4.1. A given irreducible reliability graph.

Figure 3.4.2. A 2-part partition of the given graph using the minimal s - t boundary $\{b, c\}$.

Figure 3.4.3. A 2-part partition of the given graph using the minimal s - t boundary $\{a, c\}$.

PART 1					PART 2				
N^1					N^2				
$N^1 \backslash S(N^1)$					$N^2 \backslash S(N^2)$				
0 0 0 0	0 0	0 0 0 0	0 0	0 0	0 0 0	0 0	0 0	0	
0 0 0 1	0 0	0 0 0 1	0 0	0 0	0 0 1	0 0	0 1	1	
0 0 1 0	0 0	0 0 1 0	0 0	0 0	0 1 0	0 1	0 0	1	
0 0 1 1	0 0	0 0 1 1	0 0	0 0	0 1 1	0 1	0 1	1	
0 1 0 0	0 1	1 0 0 0	0 1	0 0	1 0 0	0 0	0 0	0	
0 1 0 1	0 1	1 0 0 1	0 1	0 0	1 0 1	0 1	0 1	1	
0 1 1 0	0 1	1 0 1 0	0 1	0 0	1 1 0	0 1	0 1	1	
0 1 1 1	0 1	1 0 1 1	0 1	0 0	1 1 1	0 1	0 1	1	
1 0 0 0	0 0								
1 0 0 1	0 0								
1 0 1 0	0 1								
1 0 1 1	0 1								
1 1 0 0	0 1								
1 1 0 1	0 1								
1 1 1 0	0 1								
1 1 1 1	0 1								

Table 3.4.4. The part matrices corresponding to the partition of Figure 3.4.2.

u_j	PART 1			PART 2	
	x_{juj}^1	p_{juj}^1	x_{uj1}^2	p_{uj1}^2	
1	0 1 0 0	.1, .6, .3, .1 = .0012	0 1 0	.1, .6, .3, .1 = .012	
	0 1 0 1	.1, .6, .3, .3 = .0162	0 1 1	.1, .6, .7 = .042	
	0 1 1 0	.1, .6, .7, .1 = .0042	1 0 1	.9, .4, .7 = .252	
	1 0 1 0	.9, .4, .7, .1 = .0252	1 1 0	.3, .6, .3 = .162	
	1 1 0 0	.9, .6, .3, .1 = .0162	1 1 1	.9, .6, .7 = .378	
2	1 1 1 0	.9, .6, .7, .1 = .0378			.252
		.1014			
	1 0 0 1	.9, .4, .3, .3 = .0972	0 0 1	.1, .4, .7 = .028	
			0 1 1	.1, .6, .7 = .042	
			1 0 1	.9, .4, .7 = .252	
3			1 1 0	.9, .6, .3 = .162	
			1 1 1	.9, .6, .7 = .378	
					.862
	0 1 1 1	.1, .6, .7, .9 = .0378	0 0 1	.1, .4, .7 = .028	
	1 0 1 1	.9, .4, .7, .9 = .2268	0 1 0	.1, .6, .3 = .012	
4	1 1 0 1	.9, .6, .3, .9 = .1458	0 1 1	.1, .6, .7 = .042	
	1 1 1 1	.9, .6, .7, .9 = .3402	1 0 1	.9, .4, .7 = .252	
		.7506	1 1 0	.9, .6, .3 = .162	
			1 1 1	.9, .6, .7 = .378	
					.820

Table 3.4.5. The calculations involved in computing R_{gt} , using the partition of

Figure 3.4.2, when applying equation 3.3.10.8.

u_j	Part 2, in the form used for computing $R_{u_j t}$	$R_{u_j t} = P_{u_j t}^2$
1		$1 - .4 \times (1 - .9 \times .7) = .852$
2		$1 - .5 \times (1 - .9 \times .6) = .862$
3		$1 - .4 \times .5 = .80$

Table 3.4.6. The calculations involved with part 2 of the partition in Figure 3.4.2., when applying the equation 3.3.10.b.

3.5 A CRITERION OF OPTIMALITY

In applying the 2-part decomposition method in the previous section, the aim was to illustrate the concepts involved rather than carrying out the calculations in a systematic way. The aim of this section is to develop simple rules which will allow 2-part decomposition to be applied in an efficient manner, that is, with minimum computational effort.

Let us consider again the reliability graph of Figure 3.4.1. The main steps followed for the calculation of the terminal reliability R_{st} can be summarized as follows:

- (1) Decomposition of the graph into two parts using a minimal s - t boundary.
- (2) Construction of the part matrices X^1 and X^2 .
- (3) Formation of the sets $X^1_{j\mu_1}$ and $X^2_{\mu_2 j}$, $\mu_1 = 1, 2, 3$, and calculation of their probabilities, $P^1_{j\mu_1}$ and $P^2_{\mu_2 j}$.
- (4) Calculation of the terminal reliability R_{st} applying equation 3.3.10.

Of these four steps the most time consuming is the second, that is, the construction of the part matrices X^1 and X^2 . As it was explained in the previous section, each column of these matrices headed by a single node requires a number of path-tracing searches in order to determine the entries. The total number of path-tracing searches required for the construction of the part matrices X^1 and X^2 will be referred to as the complexity of the 2-part decomposition method and it will be denoted as C_g . In order to evaluate

complexity let us recall that the matrix N^1 contains only one column, namely, the column headed by node s . The number of entries in this column is 2^{b_1} - the number of minterms of part 1. On the other hand, the matrix N^2 contains as many columns headed by a single node, as there are nodes in the boundary N_1 - the minimal s - t boundary used to realize the 2-part partition of the graph. The number of entries in each such column is 2^{b_2} - the number of minterms of part 2. Let μ_1 denote the number of nodes in the boundary N_1 . Then the total number of path-tracing searches required for the construction of the part matrix N^2 is equal to $\mu_1 2^{b_2}$. Thus, the complexity of the 2-part decomposition can be expressed as

$$C_g = 2^{b_1} + \mu_1 2^{b_2} \quad (3.5.1.)$$

The complexity C_g as defined above, provides a good measure of the computational effort required when applying the 2-part decomposition method. However, it should be realized, from equation 3.5.1., that this complexity depends on the particular graph partition used when applying the method. It is natural then to want to use this graph partition which makes the number C_g as small as possible. The following theorem provides a simple rule for choosing the 2-part partition which minimizes the complexity C_g .

Theorem 3.5.2. Given a positive real number b , let b_1, b_2 be two non-negative real numbers such that $b_1 + b_2 = b$. Then for any given positive real number μ_1 , the quantity $C_g = 2^{b_1} + \mu_1 2^{b_2}$ becomes minimum when $b_1 = \frac{1}{2}(b + \log_2 \mu_1)$ and $b_2 = \frac{1}{2}(b - \log_2 \mu_1)$.

In view of the above theorem, let us classify the available 2-part partitions of a graph into classes. Each class contains the partitions with the same number of boundary nodes μ_1 . Within each class, the partition presenting the lower complexity is the one in which the numbers of branches are: $b_1 = \frac{1}{2}(b + \log_2 \mu_1)$ and $b_2 = \frac{1}{2}(b - \log_2 \mu_1)$. Since it may happen that the numbers $\frac{1}{2}(b + \log_2 \mu_1)$ and $\frac{1}{2}(b - \log_2 \mu_1)$ are not integers, one should take care and replace these numbers with the nearest integers whose sum is the number of branches b . In the example of Figure 3.4.1. there are four 2-part partitions of the graph, generated by the boundaries $\{a, c\}$ and $\{b, c\}$. Since $\mu_1 = 2$, for both boundaries, all four partitions belong to the same class. The partition which presents the lower complexity should have:

$$b_1 = \frac{1}{2}(b + \log_2 \mu_1) = \frac{1}{2}(7 + \log_2 2) = 4$$

$$b_2 = \frac{1}{2}(b - \log_2 \mu_1) = \frac{1}{2}(7 - \log_2 2) = 3$$

Such a partition exists and it is shown in Figure 3.4.2. It is, precisely, the partition used for the calculation of R_{st} in the previous section. Applying equation 3.5.1, the complexity of this partition is calculated as:

$$C_2 = 2^{b_1 + \mu_1} 2^{b_2} = 2^{4+2} \cdot 2^3 = 32$$

The partition shown in Figure 3.4.3 with $b_1 = 2$ and $b_2 = 5$, presents the highest complexity among the four partitions. Its complexity is:

$$C_2 = 2^{b_1 + \mu_1} 2^{b_2} = 2^{2+2} \cdot 2^5 = 68$$

It is interesting to note that this highest complexity partition is precisely the one used when applying the algorithm proposed by E. Hänsler (1972) to the graph of Figure 3.4.1 ; this algorithm was discussed in section 3.3. This is a very important remark. Hänsler's algorithm, makes use of a specific partition of the reliability graph. As this example indicates, it may very well happen that this specific partition is the one of highest complexity. On the other hand, the 2-part decomposition method allows relative freedom in the choice of the graph partition to be used for the calculations. This means that, following the criteria derived from Theorem 3.5.2 , one can choose the partition of lowest complexity, thus minimizing the required computational effort. The freedom of choice offered by the decomposition method results in substantial computational savings, as it will be shown in the next chapter.

It should be realized that, when classifying the available 2-part partitions of a reliability graph according to the number of boundary nodes μ_1 , it is not necessarily true that complexity increases as the number μ_1 increases. As an example consider the graph shown in Figure 3.5.3. It should be easy to check that the lowest complexity obtained from a partition with $\mu_1 = 2$ is equal to 1032. On the other hand, the lowest complexity obtained from a partition with $\mu_1 = 3$ is equal to 822.

It is interesting to note that, in the context of the decomposition method, the minimum value μ_1 can take is assumed to be 2. For, if there is an s - t boundary consisting of only one node, say, node z , which is different than either node s or node t , then there is no need to follow any specific calculation scheme. The calculation of terminal reliability in this case is

straightforward: The reliability graph is separated by node s into two graphs G_1 and G_2 ; graph G_1 contains node s and graph G_2 contains node t . The terminal reliability R_{st} is then expressed as:

$$R_{st} = P_{ss} P_{st}$$

where P_{ss} is calculated in graph G_1 and P_{st} in graph G_2 .

Figure 3.5.3. A reliability graph in which the complexity decreases as the number of boundary nodes k_1 increases.

3.6 DISCUSSION AND CONCLUSIONS

A graph partitioning technique was used to decompose the problem of calculating terminal reliability into two subproblems. The results obtained from reliability calculations in the subproblems were then combined to derive the terminal reliability of the original problem.

The 2-part partition of the original graph is obtained using a minimal s - t boundary. The reliability calculations in the two parts (subgraphs) are carried out in a systematic way, using the part matrices M^1 and M^2 . The results of these calculations are then combined in equation 3.3.10.a to obtain the terminal reliability R_{st} . The computational complexity was shown to depend on the graph partition used for the calculations. If the rule obtained from Theorem 3.5.2 is followed for the selection of the graph partition, then this complexity is minimized. In certain cases, partitions of higher complexity may be considered, if part 2 of the graph has a series-parallel structure. The terminal reliability is then obtained from equation 3.3.10.b.

A decomposition algorithm proposed by E. Hänsler (1972) was shown to be a special case of the 2-part decomposition method presented in this chapter. The algorithm makes use of a specific partition of the graph and can be obtained from equation 3.3.10.b. The use of a fixed partition makes the efficiency of the algorithm heavily dependent on the configuration of the reliability graph. Thus, as it was shown in section 3.5, it may very well happen that this fixed partition is the one of highest complexity. The 2-part decomposition method, on the other hand, allows for the choice of the partition

which minimizes computational complexity. As a result, substantial computational savings can be expected relative to Hansler's algorithm.

In an effort to simplify the presentation, it was assumed, throughout this chapter, that the branches of the reliability graph are undirected and that the nodes are perfectly reliable. It should be clear that in order to apply the decomposition method to directed graphs all one has to do is to replace "minimal s - t path" and "cut" with "minimal directed s - t path" and "semitcut", throughout. On the other hand, if failing nodes are to be included, then these nodes can be treated in the same manner as the branches, except for the nodes in the boundary N_1 . These nodes will be treated as perfectly reliable nodes in part 1 and they will be considered as operating with their given reliabilities in part 2 of the graph partition. Another way is to replace every failing node with two perfectly reliable nodes connected with a failing branch whose reliability is the reliability of the failing node.

The 2-part decomposition method seems to provide a powerful tool for computing terminal reliability in relatively small graphs. It is interesting to note that substantial computational savings result when the terminal reliability is calculated for more than one pair of nodes. It should be clear that when computing the terminal reliability between node s and any node of part 2, the part matrix M_1^1 is always the same. Similarly, when computing the terminal reliability between node t and any node of part 1, the part matrix M_2^2 is always the same.

As the size of the reliability graph increases, the two parts of the decomposition can be quite large and a partition of the graph into more than two parts may be necessary. This multipart decomposition of the graph can be generated using equation 3.3.10.b in a recursive manner. The details of the method are presented in the following chapter.

CHAPTER FOUR

THE MULTIPART DECOMPOSITION METHOD

The 2-part decomposition method is applied recursively to generate the multipart decomposition of a large system. The computational structure of this decomposition is studied by means of the so-called reliability tree. This tree provides the basis for the development of a powerful algorithm for the computation of terminal reliability in large systems. The computational complexity of the algorithm is discussed and results from computational experience are presented.

4.1 THE MULTIPART DECOMPOSITION AND ITS RELIABILITY TREE

The aim of the method presented in the previous chapter was to decompose the problem of calculating terminal reliability into two subproblems of manageable size. A graph partitioning technique was proposed for decomposing the reliability graph into two subgraphs. Terminal reliability was then obtained from reliability calculations in the subgraphs. However, as the size of the reliability graph increases, the subgraphs of the decomposition can be quite large, that is, the corresponding subproblems may not be of manageable size. In this case, a decomposition of the reliability graph into more than two subgraphs seems to be the natural approach. The questions which arise in this context are:

- (1) How to decompose the reliability graph?
- (2) How to obtain terminal reliability from reliability calculations in the subgraphs.
- (3) How to recognize whether or not a multipart decomposition of the graph will produce the exact value of terminal reliability.

The answers to all three questions are based on the results obtained from the 2-part decomposition method.

We shall answer these questions using a specific reliability graph, namely, the graph shown in Figure 4.1.3 ; the answers will then be generalized in the following section. Suppose that the terminal reliability R_{18} is to be calculated, using a 2-part partition of the graph. This partition is obtained

2

OF/DE

2

from the minimal 1-8 boundary $N_1 = \{3, 4\}$ which decomposes the graph into two parts, I and II. According to equation 3.3.10.b, the terminal reliability can be expressed as:

$$R_{18} = \sum_{u_1=1}^3 P_{1u_1}^I R_{u_1 8}^{II} \quad (4.1.1.)$$

Thus, the calculation of the terminal reliability R_{18} involves the calculation of the terminal reliabilities $R_{u_1 8}^{II}$, $u_1 = 1, 2, 3$, in subgraph II of the decomposition. Let us consider calculating these terminal reliabilities using a 2-part partition of subgraph II. This partition is obtained using the boundary $N_2 = \{5, 6, 7\}$ which is a minimal 3-8 boundary as well as a minimal 4-8 boundary, hence, a minimal 1-8 boundary. This boundary decomposes part II of the graph into two parts, 2 and 3, as shown in Figure 4.1.3. As a result, the original reliability graph is decomposed into three parts, 1, 2 and 3 (part 1 is identical to part I of the original 2-part partition). Thus, using the minimal 1-8 boundaries N_1 and N_2 , we obtain a 3-part partition of the original graph, and if we let $N_0 = \{1\}$ and $N_3 = \{8\}$, then part k of this partition is contained between the boundaries N_{k-1} and N_k , $k = 1, 2, 3$.

The next task is to obtain terminal reliability R_{18} from reliability calculations in the subgraphs of this 3-part partition. According to equation 3.3.10.a, the terminal reliability $R_{u_1 8}^{II}$ can be written as:

$$R_{u_1 8}^{II} = \sum_{u_2=1}^2 P_{u_1 u_2}^2 P_{u_2 8}^3$$

Hence, equation 4.1.1 can be rewritten as:

$$R_{18} = \sum_{u_1=1}^3 P_{1u_1}^1 \sum_{u_2=1}^7 P_{u_1 u_2}^2 P_{u_2 1}^3 \quad (4.1.2)$$

The problem of calculating the terminal reliability R_{18} now becomes one of organizing the computational data obtained from the subgraphs of the decomposition. This problem can be handled conveniently by means of a tree representation of the data structure. This tree contains three parts, that is, it contains as many parts as the graph partition, and it is defined as shown in Figure 4.1.4. The unique non-empty subset of boundary N_0 is represented as node 1 and it is connected by a branch to each of the three non-empty subsets of boundary N_1 , represented as nodes 1, 2 and 3. As a result, we obtain a special kind of a tree called a rooted tree; node 1 of boundary N_0 is the root of this tree; nodes 1, 2 and 3 of boundary N_1 are the leaves. The three branches of this rooted tree represent the sets x_{11}^1 , x_{12}^1 and x_{13}^1 , respectively. That is, every branch represents the set of minterms x^1 each with the following property: the ON components connect at least one node in the root to each and every node in the leaf. Next, the leaves of part 1 serve as the roots of the rooted trees which make up part 2. For instance, node 1 of boundary N_1 is now the root of a rooted tree whose leaves are nodes 1 through 7 of boundary N_2 . These leaves represent the seven non-empty subsets of boundary N_2 ; the branches of this tree represent the sets $x_{u_1 u_2}^2$, where $u_1 = 1$ and $u_2 = 1, 2, \dots, 7$. If we repeat the process for $u_1 = 2, 3$, then we obtain the three rooted trees which make up part 2. In a similar manner one can obtain the 21 rooted trees which

make up part 3. Observe that each of these last trees has only one branch.

It should be clear that the procedure described above generates a tree which is characteristic of the 3-part partition used in our example. This tree is called the reliability tree of the partition and it represents the relative structure of the data obtained from the subgraphs. In order to calculate the terminal reliability R_{18} using the reliability tree, each branch of this tree is assigned a probability and each node is assigned a weight. The probability assigned to a branch is precisely the probability of the set of minterms represented by the branch. On the other hand, every node of the reliability tree is a root of some rooted tree except for the nodes of the boundary N_3 . Each node of the reliability tree which is not a root is assigned a weight equal to 1. Each node which is a root of a rooted tree is assigned a weight equal to the sum of the probabilities of the branches of this rooted tree, each weighted with the weight of the corresponding leaf. It should be easy to verify that the weight of node 1 of the boundary N_0 of the reliability tree, is the terminal reliability R_{18} as expressed by equation 4.1.2.

Thus, the reliability tree provides an easy, as well as systematic, way for calculating terminal reliability from the subgraphs of a graph partition. Further, the reliability tree provides valuable insight to the structure of the decomposition method itself. Let us interpret the qualitative information contained in the various parts of the reliability tree of Figure 4.1.4. Part 1 of the reliability tree implies that from the minterms x of the given graph, only those whose part x^1 connects to some node of the boundary N_1 of the reliability tree may be applicable minterms. The three branches of part 1,

represent a partition of these minterms into three pairwise disjoint subsets. Let us follow those minterms x whose part x^1 connects to node 1 of the boundary N_1 . Part 2 of the reliability tree implies that of these minterms only those whose part x^2 connects to some node of the boundary N_2 of the reliability tree may be applicable minterms. The seven branches of the corresponding rooted tree together with branch (1,1) of part 1, define a partition of these minterms x into seven pairwise disjoint subsets. Let us follow those minterms x which connect up to node 1 of boundary N_2 . Part 3 of the reliability tree implies that of these minterms only those whose part x^3 connects to node 1 of the boundary N_3 of the reliability tree (i.e., to node 8 of the graph) are applicable.

Thus, the reliability tree of a graph partition indicates how the minterms representing the states of the subgraphs in the partition should be put together in order to generate applicable minterms of the original graph. In the process, these applicable minterms are classified into pairwise disjoint sets (i.e., mutually exclusive events) so that event probabilities can be added in order to obtain terminal reliability. The question remains, however, whether the value obtained for the terminal reliability is an exact or an approximate one. To answer this question we shall make use of the decomposition rules derived in section 3.3.

The 3-part partition of the graph shown in Figure 4.1.3 was obtained using two 2-part partitions: The original graph was first decomposed into two parts using boundary N_1 ; then, part II of this decomposition was further decomposed into two parts using boundary N_2 . Since the boundary N_1 contains two nodes, the associated decomposition corresponds to exact calculation.

On the other hand, boundary N_2 contains three nodes; however, since the second part of the associated 2-part partition contains no non-boundary nodes, this partition also corresponds to exact calculation. Thus the value of the terminal reliability R_{18} , obtained from the 3-part partition of Figure 4.1.3, is an exact value.

In conclusion, the questions posed in this section were answered using a specific reliability graph. In the following section the answers are generalized and the reliability tree is used to derive an algorithm for calculating terminal reliability.

Figure 4.1.3. A 3-part decomposition of a reliability graph using the minimal 1-8 boundaries 3, 4 and 5, 6, 7.

Figure 4.1.4. The reliability tree corresponding to the 3-part partition of Figure 4.1.3.

4.2 COMPUTING TERMINAL RELIABILITY FROM A MULTIPART DECOMPOSITION

In this section we shall consider a decomposition of the reliability graph into m -parts and try to develop an algorithm for computing terminal reliability using the reliability tree associated with this m -part partition. First, let us describe a procedure for generating an m -part partition of a given graph.

An m -part partition of a given reliability graph is obtained using $m-1$ minimal s - t boundaries $N_1, N_2, \dots, N_{m-1}$, defined as follows:

- (1) Select a minimal s - t boundary N_1 ; this boundary decomposes the given graph into two parts (subgraphs), S_1 and S_2 .
- (2) In subgraph S_k , select a boundary N_k which is a minimal s_k - t boundary for every $i = 1, 2, \dots, k-1$ (n_{k-1} is the number of nodes in the boundary N_{k-1}). This boundary decomposes subgraph S_k into two parts (subgraphs), k and S_{k+1} .
- (3) Repeat step (2) until subgraph S_m is obtained; call this subgraph "part m " and end the procedure.

It should be clear that the boundaries $N_1, N_2, \dots, N_{m-1}$, used to obtain the m -part partition of the given graph, are all minimal s - t boundaries of a special kind: no two of them have common nodes. Such boundaries will be called non-intersecting boundaries. Thus, a simple way to obtain an m -part partition of a given graph is the following:

Select $m+1$ non-intersecting minimal s - t boundaries $N_0, N_1, \dots, N_m$ (numbered sequentially); where $N_0 = \{s\}$ and $N_m = \{t\}$. Then, part k of

the m -part partition is the subgraph contained between the boundaries N_{k-1} and N_k , $k = 1, 2, \dots, m$.

The next task is to define the reliability tree associated with an m -part partition. This tree consists of m parts $1, 2, \dots, m$, that is, it consists of as many parts as there are subgraphs in the partition. Part k of the tree is contained between the boundaries $k-1$ and k and it consists of a number of rooted trees. The reliability tree is obtained as follows: The unique non-empty subset of boundary N_0 is represented as node 1 and it is connected by a branch to each of the ℓ_1 non-empty subsets of boundary N_1 , represented as nodes $1, 2, \dots, \ell_1$. As a result, we obtain a unique rooted tree for part 1. Next, the leaves of part 1, that is, the nodes $1, 2, \dots, \ell_1$ of boundary N_1 , serve as the roots of the rooted trees which make up part 2 of the reliability tree. Thus, node u_1 of boundary N_1 is now the root of a rooted tree whose leaves are nodes $1, 2, \dots, \ell_2$ of boundary N_2 . As u_1 can be any of the nodes $1, 2, \dots, \ell_1$, part 2 of the reliability tree consists of ℓ_1 rooted trees. Similarly, part 3 consists of $\ell_1 \ell_2$ rooted trees, each having, as leaves, the nodes $1, 2, \dots, \ell_3$ of boundary N_3 . In general, part k consists of $\ell_1 \ell_2 \dots \ell_{k-1}$ rooted trees, each having, as leaves, the nodes $1, 2, \dots, \ell_k$ of boundary N_k , $k = 1, 2, \dots, m$. Each branch of the reliability tree belongs to some part, say, part k , of the tree. Such a branch represents the set of minterms x^k each with the following property: the ON components connect at least one node in the root to each and every node in the leaf. It should be clear that this set of minterms can be obtained from the part matrix N^k . The branch-probability is defined as the probability of the set of minterms represented by the branch.

Thus, the reliability tree contains all the information necessary to calculate the terminal reliability R_{st} . This information consists of m pieces obtained from the subgraphs of the decomposition. The reliability tree indicates, precisely, how these m pieces of information should be combined in order to produce the value of the terminal reliability R_{st} . Let us try then to develop a calculation scheme based on the structure of the reliability tree. Equation 3.3.10.b is the key to the answer. A recursive application of this equation suggests the following steps for calculating the terminal reliability R_{st} from the reliability tree of a graph partition:

- (1) To every node of the reliability tree which is not a root, assign a weight W equal to 1.
- (2) To every node of the reliability tree which is a root, assign a weight equal to the sum of the branch probabilities each weighted with the weight of the corresponding leaf.
- (3) The weight of node 1 of boundary N_0 is the terminal reliability R_{st} .

This calculation scheme can be formalized as an algorithm for computing terminal reliability. This algorithm, called the Decomposition Algorithm, is described in Figure 4.2.5.

What is the computational complexity of this algorithm? The most time consuming operation when constructing the reliability tree is calculating the branch probabilities. The necessary information for this calculation is obtained from the part matrices of the graph partition. Thus, in much the same way as it was done for a 2-part partition, the computational complexity

of an m -part partition is defined as the total number of path-tracing searches required for the construction of the part matrices N^k , $k = 1, 2, \dots, m$. This complexity will be denoted as C_m . Let μ_k be the number of nodes in the boundary N_k and b_k the number of branches in part k of an m -part partition, $k = 1, 2, \dots, m$. For convenience, let $\mu_0 = 1$. Then, the number of path-tracing searches required for the construction of the part matrix N^k is $\mu_{k-1} 2^{b_k}$, $k = 1, 2, \dots, m$. Thus, the complexity C_m can be expressed as

$$C_m = \sum_{k=1}^m \mu_{k-1} 2^{b_k} \quad (4.2.1.)$$

The complexity C_m provides a good measure of the computational effort required when computing terminal reliability from an m -part partition of the graph. It is natural to keep this complexity as low as possible.

It should be clear from equation 4.2.1 that the complexity C_m depends on the specific m -part partition used in the calculation of terminal reliability. The question is: Among the available m -part partitions of the graph what is the one corresponding to minimum complexity? In section 3.3.5 a simple answer was obtained for $m = 2$. When $m > 2$ the situation becomes more complicated. The following theorem, however, provides a basis for answering this question:

Theorem 4.2.2. Let m, N be integers greater than or equal to 2 and let b be a positive real number. Let $\mu_1, \mu_2, \dots, \mu_{m-1}$ be integers and $b_1, b_2, \dots, b_m$ positive reals, such that:

$$\mu_1 \mu_2 \dots \mu_{m-1} = N \text{ and } \mu_k \geq 2, \quad k = 1, 2, \dots, m-1$$

$$b_1 + b_2 + \dots + b_m = b$$

Then the quantity:

$$C_m = \sum_{k=1}^m r_{k-1} 2^{b_k}$$

becomes minimum when:

$$b_k = \frac{1}{m}(b + \log_2 M) - \log_2 r_{k-1}$$

In view of the above theorem let us classify all m -part partitions of a given graph into classes according to the product M . Then, the minimum-complexity partition, in each class, is the one in which the number of branches in part k equals the integer best approximating the number: $\frac{1}{m}(b + \log_2 M) - \log_2 r_{k-1}$. For example, in the graph of Figure 4.1.3 there are two 3-part partitions with $M = 6$. The partition of minimum complexity should have:

$$b_1 = \frac{1}{3}(13 + \log_2 6) - \log_2 1 \approx 5$$

$$b_2 = \frac{1}{3}(13 + \log_2 6) - \log_2 2 \approx 4$$

$$b_3 = \frac{1}{3}(13 + \log_2 6) - \log_2 3 \approx 4$$

This partition is the one obtained using the boundaries $N_1 = \{3, 4\}$ and $N_2 = \{5, 6, 7\}$, as shown in Figure 4.1.3.

It is interesting to note that, according to theorem 4.2.2, the 2-part partition of minimum complexity is obtained when:

$$b_1 = \frac{1}{2}(b + \log_2 r_1)$$

$$b_2 = \frac{1}{2}(b - \log_2 r_1)$$

This is precisely the result obtained for 2-part partitions in section 3.5.

When the partition of the reliability graph is carried out using boundaries each having two nodes, the corresponding value of terminal reliability is always an exact value. According to theorem 4.2.2, such an m -part partition is of minimum complexity when the number of branches in part k equals the integer best approximating the number: $\frac{1}{m}(b+m-1) - \log_2 r_{k-1}$, $k = 1, 2, \dots, m$.

An interesting case arises when component reliabilities can be assumed equal. In this case, the terminal reliability R_{st} is a function of the common component reliability. Let us denote this reliability by x . Then, the function $R_{st}(x)$ — usually called the reliability function — describes the behaviour of the system between nodes s and t in terms of the common component reliability. It should be clear that the function $R_{st}(x)$ is a polynomial in x whose degree is, at most, equal to the number of branches in the reliability graph. The coefficients of this polynomial can be computed from an m -part

partition of the graph using the associated reliability tree. The necessary information is obtained from the branches of this tree using the binomial theorem.

As discussed earlier in this section, the branch joining nodes u_{k-1} and u_k of the reliability tree represents the set of minterms $x_{u_{k-1}u_k}^k$. Let us consider a minterm x^k from this set and let h be the number of ON components in this minterm. Then, the probability of x^k can be expressed as

$$p(x^k) = r^h(1-r)^{b_k-h} \quad (4.2.3.)$$

Let $c(i)$ be the coefficient of r^i in the above equation, $i = 0, 1, \dots, b_k$.

Applying the binomial theorem, we obtain the following expression for $c(i)$:

$$c(i) = \begin{cases} \binom{b_k-h}{i-h} (-1)^{i-h}, & \text{if } h \leq i \leq b_k \\ 0, & \text{if } 0 \leq i < h \end{cases} \quad (4.2.4.)$$

Using this last equation, one can obtain the coefficient of r^i corresponding to the probability of the set $x_{u_{k-1}u_k}^k$, $i = 0, 1, \dots, b_k$. Thus, each branch of the reliability tree can be assigned the set of coefficients obtained for the branch probability. Once this is done for every branch, the reliability tree can be used to formulate an algorithm — similar to the one described in Figure 4.2.5 — for computing the coefficients of the reliability function.

In summary then, computing terminal reliability from an m -part partition of a graph is done using the decomposition algorithm described in Figure 4.2.5. A similar algorithm can be used for the computation of the coefficients in the reliability function. In both cases, computational time is minimized if the graph partition is chosen according to the rules derived from theorem 4.2.2.

Figure 4.2.5. The decomposition algorithm for computing terminal reliability from an m -part partition of a reliability graph.

4.3 COMPUTATIONAL EXPERIENCE

In the previous section a decomposition algorithm was developed for the calculation of terminal reliability from an m -part partition of the reliability graph. The purpose of this section is to explain, by means of examples, the application of this algorithm and to discuss its computational efficiency. Two recently proposed algorithms will be used for evaluating the decomposition algorithm. The first, proposed by L. Fratta and U.G. Montanari (1973), is based on path enumeration; the second, proposed by E. Hänsler, G.K. McAuliffe and R.S. Wilkov (1974), is based on cut enumeration. We shall refer to these two algorithms as FM and HMW, respectively.

The Decomposition Algorithm (DA) was implemented by a computer program given the title DECOMP01. The functions in this program are written in APL (A Programming Language) and reside on the Carleton University Sigma-9 time-shared computer. The program has been written so as to allow the user to enter conversationally the data of the reliability graph and execute the algorithm.

A number of reliability graphs were chosen, at random, from the current literature and DECOMP01 was run in order to compute terminal reliabilities for different pairs of nodes in each graph. Partitions of various complexities were used in the application of the algorithm. These partitions were all of minimum complexity chosen so as to yield exact values of terminal reliabilities. The results of computational experience with the decomposition algorithm are summarized in the diagram of Figure 4.3.1. This diagram shows how the CPU time T increases as the computational complexity C increases. It should be noted

that the CPU time corresponding to each point of the diagram represents the average time taken for the computation of terminal reliability in a number of graph partitions of, approximately, the same complexity C . The available results of computational experience with the FM and HMW algorithm are also indicated in the diagram for comparison purposes.

It should be clear from the diagram that the decomposition algorithm offers substantial computational savings relative to the FM and HMW algorithms and that these savings improve as computational complexity increases. This result was expected in view of the relative structures of the three algorithms. The FM algorithm is based on a manipulation of the whole set of minimal s - t paths in the reliability graph. Similarly, the HMW algorithm generates mutually exclusive events from the set of minimal s - t cuts in the reliability graph. The decomposition algorithm, on the other hand, is based on manipulations of the sets of paths in the subgraphs of an appropriate partition of the reliability graph. Thus, depending on the size and the configuration of the reliability graph, the number of paths (or cuts) requiring examination by the decomposition algorithm can be drastically smaller. This should result in computational savings and these savings should improve as the size of the graph increases.

Let us consider some specific examples. First, let us consider the example of the, relatively small, graph shown in Figure 4.3.2. This graph has been used by E. Hänsler, G.K. McAuliffe and R.S. Wilkov (1974) as an example for the application of their method for computing terminal reliability. Application of the HMW algorithm required 18 sec. of CPU time to produce

the exact value of terminal reliability R_{96} from an APL implementation on an IBM 360/91 computer. Using DECOMP01, this reliability was obtained in only 4 sec, from the minimum-complexity partition. That is, a reduction, in CPU time, of about 80%, is observed, relative to the HMW algorithm.

In larger graphs it is not always possible to find a partition of satisfactory complexity which yields the exact value of terminal reliability. Fortunately, this difficulty does not arise very often in practice, due, mainly, to the sparsity that many real systems exhibit. Let us consider as an example the ARPA (Advanced Research Project Agency) computer network, presently in operation in U.S.A. A configuration of this network is shown in Figure 4.3.3. Suppose that the terminal reliability between UCLA and CMU is to be computed. This is the terminal reliability R_{78} in the equivalent irreducible graph obtained after all possible series-parallel reductions have been performed.

In this graph, no minimal 7-8 boundary contains more than two nodes; hence, any partition of the graph yields the exact value of terminal reliability R_{78} . Application of the FM algorithm required 112 sec of CPU time to produce the exact value of this reliability from a FORTRAN IV implementation on an IBM 360/67 computer. Using DECOMP01 this result was obtained in only 11 sec of CPU time, from the minimum-complexity partition. That is, a reduction, in CPU time, of about 90%, is observed, relative to the HMW algorithm.

Graphs of the ARPA type are usually called sparse. The term sparse graph is used, in the current literature, to indicate graphs in which less than 4 branches emanate, on the average, from each node. In such graphs it is usually possible to find partitions which yield the exact value of terminal

reliability. Computational experience showed that sparse graphs with up to 20 branches, present complexities less than 2^9 ; that is, it is usually possible to partition such graphs into two or more parts (subgraphs) each having less than 7 branches. According to the diagram of Figure 4.3.1, exact values of terminal reliabilities in this class of graphs should be obtained in less than 80 sec. of CPU time. On the other hand, it is interesting to note that a number of sensitivity computations were carried out in sparse graphs with up to 20 branches, following the technique described in 2.4; CPU time never exceeded 105 sec. Hence, it is reasonable to conclude that for systems like the ARPA network, reliability calculations can be carried out very rapidly using the decomposition algorithm. Thus, reliability analysis of such systems can take place with throughput analysis in an iterative design procedure.

In the existing techniques for the exact or approximate calculation of terminal reliability, computation time grows exponentially with the size of the graph. As a result, these techniques are applicable to irreducible graphs with less than 20 branches. Thus, no computational results are available to compare the efficiency of the decomposition algorithm when applied to larger graphs. An example of the application of the algorithm to a large graph is given in Figure 4.3.4. This is a sparse graph with 27 branches. A 4-part partition of the graph was used to obtain the terminal reliability R_{st} ; its complexity is

$$2^7 + 2 \cdot 2^6 + 2 \cdot 2^7 + 2 \cdot 2^7 = 3 \cdot 2^8$$

Since no boundary contains more than two nodes, this partition yields the exact value of terminal reliability. Using DECOMP01 this value was obtained in 109 sec. of CPU time. It is interesting to note that a 3-part partition of the same graph was used to obtain the terminal reliability R_{st} . This partition is shown in Figure 4.3.4 ; its complexity is:

$$2^{10} + 3 \cdot 2^{10} + 2 \cdot 2^7 = 2^{12} + 2^8$$

The execution of the program was stopped after 4 min of CPU time. At that point, the program was calculating parameters in subgraph 2 of the partition. It should be also noted that this 3-part partition would not yield an exact value of R_{st} . This is a striking example of how important the choice of partition is for the application of the decomposition algorithm.

In theory, the decomposition algorithm applies to any reliability graph and the accuracy of results depends on the graph partition used for the application of the algorithm. However, computational experience showed that, in terms of computation time, the algorithm is applicable only to graph partitions with complexity less than 2^{12} . For sparse graphs, this complexity corresponds to about 40 branches. It is interesting to note at this point that in sparse graphs it is, sometimes, possible to modify the graph so as to obtain lower complexities. There are two ways to do this; the first consists in adding zero-reliability branches at appropriate places in the graph, so as to "force" a desirable graph partition; the second is to apply the conditional probability technique, described in 2.4, for an appropriate branch of the reliability graph. We shall illustrate these techniques using examples.

The graph shown in Figure 4.3.5 presents a minimum complexity of $2^7 + 2^5 + 2^3$, with respect to terminal reliability R_{18} . This complexity corresponds to the 3-part partition obtained using the boundaries $\{3,4\}$ and $\{5,7\}$, as indicated in the figure. Adding a zero-reliability branch between nodes 6 and 8 "forces" the set $\{5,6,7\}$ to become a minimal 1-8 boundary. Using this boundary we can obtain a new 3-part partition of the graph with complexity equal to: $2^5 + 2 \cdot 2^4 + 2 \cdot 2^5 = 2^7$. It should be observed that both partitions yield exact values of the terminal reliability R_{18} .

In the graph shown in Figure 4.3.6, the exact value of the terminal reliability R_{8t} can be obtained with a minimum complexity of:

$$2^3 + 3 \cdot 2^7 + 2 \cdot 2^3 = 418$$

If we apply the conditional probability technique, described in 2.4, for branch number 11, then two simpler graphs are obtained. After series-parallel reductions, the terminal reliability R_{8t} can be obtained from these two graphs with a minimum (total) complexity of:

$$(2^5 + 2 \cdot 2^3 + 2 \cdot 2^3) + (2^7 + 3 \cdot 2^5) = 288$$

In conclusion, computational experience showed that the decomposition algorithm extends the capabilities of the existing techniques to larger graphs. The computational times observed in sparse graphs of the ARPA type, seem to justify reliability analysis to be considered as a part of iterative design

procedures. As the size of the reliability graph grows the decomposition algorithm becomes limited in two ways: In terms of accuracy of results it becomes limited to sparse graphs; in terms of computational complexity — hence, in terms of computational time — it becomes limited to graph partitions with complexity less than 2^{12} . These limitations are, partly, overcome by the method presented in the following chapter.

Figure 4.3.1. The results of computational experience with the Decomposition Algorithm (DA).

Figure 4.3.2. A reliability graph and its irreducible equivalent with respect to R_{96} .

Figure 4.3.3. The ARPANET computer network and its irreducible equivalent with respect to R_{78} .

Figure 4.3.4. A 27-branch "sparse graph" and its partitions into 4 and 3 parts, using minimal s - t boundaries.

Figure 4.3.5. Reducing computational complexity by adding a zero-reliability branch.

Figure 4.3.6. Applying the conditional probability technique to reduce computational complexity.

4.5 DISCUSSION AND CONCLUSIONS

The aims of this chapter were twofold; the first to develop an algorithm for computing terminal reliability from the subgraphs of a multipart decomposition of the reliability graph; the second, to evaluate this algorithm and compare computational results to those obtained in recently published analyses.

A simple method was presented for obtaining an m -part partition of a given graph using non-intersecting minimal boundaries. Then, the computational data from the subgraphs of this partition were organized into a tree structure, called the reliability tree. Based on this tree, an algorithm was formulated for computing terminal reliability. This algorithm was given the name, the decomposition algorithm. Its computational complexity as well as the accuracy of results were shown to depend on the graph partition used for the calculations.

Some simple rules were derived for selecting partitions of minimum complexity. The decomposition algorithm was then applied to a number of reliability graphs from the current literature. Computational experience showed that, in terms of computation time, the decomposition algorithm is applicable to graph partitions with complexity less than 2^{12} . However, as partitions of satisfactory complexity do not necessarily yield exact values of terminal reliability, the question was to define the class of graphs for which the algorithm applies satisfactorily both, in terms of computation time and accuracy of results. Computational experience showed that this class includes all sparse graphs with up to 40 branches. Thus, the decomposition algorithm extends the capabilities of existing techniques which are limited to

sparse graphs with less than 20 branches.

The interest in the class of sparse graphs lies in the fact that many real systems do exhibit sparsity in the sense defined in this chapter. Especially systems as large as the ARPA network can be handled extremely well by the decomposition algorithm. In fact, the computational times obtained were so small as to justify reliability analysis to take a place in iterative design procedures in such systems.

However powerful the decomposition algorithm may seem to be, it is, nevertheless, limited to sparse graphs which present a complexity less than 2^{12} . These limitations are, partly, overcome by a generalized decomposition method presented in the following chapter.

CHAPTER FIVE

THE GENERALIZED DECOMPOSITION

The concept of a minimal s - t boundary is extended and a probability theorem is used to derive a generalized decomposition method. This method allows greater flexibility in the decomposition of a reliability graph and it applies to the calculation of a variety of reliability measures. The decomposition method developed in the previous two chapters is shown to be a special case of the generalized decomposition.

5.1 INTRODUCTION

The decomposition algorithm developed in the previous chapter was shown to extend the capabilities of existing techniques to larger graphs. The computational savings observed in the case of sparse graphs of the ARPA type seem to justify reliability analysis to take a place with throughput analysis in an iterative design procedure. Computational experience showed that even in larger graphs, which exhibit some sparsity, the decomposition algorithm always produces the exact value of terminal reliability. However, the accuracy of the results obtained from the decomposition algorithm decreases as the size of the graph and the number of nodes in the minimal boundaries increase. The method presented in this chapter, called the generalized decomposition method, overcomes this difficulty in the sense that it produces always the exact value of terminal reliability.

The basic idea of the generalized decomposition method is still the same: the reliability graph is decomposed into a number of subgraphs and terminal reliability is obtained from reliability calculations in the subgraphs. However, the means for obtaining this decomposition are now different. Instead of using a minimal s - t boundary to create two subgraphs of a special kind, any subgraph of the given reliability graph can be used to create a 2-part partition. Thus, the concept of a minimal s - t boundary is extended to that of a minimal subgraph-boundary defined as follows: Given any subgraph S of the reliability graph, the minimal S -boundary is the minimal set of nodes in S whose removal from the graph disrupts all

paths joining node s or node t to any node of S . This boundary will be denoted as B . It should be clear that if either node s or node t is a node of S then this node should be included in B .

In developing the generalized decomposition method we shall need a convenient notation for the partitions of the set B . Let us recall that a partition of a set is a collection of non-empty subsets which are pairwise disjoint and make up the whole set. As it was explained in 3.1, each subset of a set can be represented by either a binary or a decimal indicator. Thus, let us agree to represent a partition of B by the set of decimal indicators of the sets in the partition. As an example, let $B = \{s, y, z\}$. There are five distinct partitions of this set. These partitions along with their decimal indicators are listed in Table 5.1.1. An easy way to order these partitions is the following: number 1 is assigned to the partition containing only one member; among the partitions with two members, number 2 is assigned to the one containing the smallest decimal indicator, number 3 to the one containing the second smallest decimal indicator, etc. The ordering in Table 5.1.1 was done in this manner.

Most of the notation introduced in 3.1 will be also used in this chapter. However, some additional notation will be necessary. This notation is listed below.

S_1 : A subgraph of the reliability graph.

S_2 : The subgraph "remaining" after removing S_1 from the reliability graph.

b_k : The number of branches in subgraph S_k , $k=1,2$.

x^k : The set of minterms of subgraph S_k , $k=1,2$.

B_1 : The minimal S_1 -boundary.

v_1 : The number of nodes in B_1 .

d_1 : The number of possible partitions of the set B_1 .

M_1 : A $2^{b_1} \times v_1$ matrix called the subgraph matrix; it is defined in section 5.3.

h_1 : The number of partitions of B_1 obtained from M_1 .

x_i^1 : The set of minterms x^1 inducing the partition number i , $i=1,2,\dots,h_1$.

P_i^1 : The probability of the set x_i^1 , $i = 1,2,\dots,h_1$.

Partition of $B = a, b, c$	Decimal Indicator	Order
$\{s, y, z\}$	$\{7\}$	1
$\{z\}, \{s, y\}$	$\{1, 6\}$	2
$\{y\}, \{s, z\}$	$\{2, 5\}$	3
$\{y, z\}, \{s\}$	$\{3, 4\}$	4
$\{z\}, \{y\}, \{s\}$	$\{1, 2, 4\}$	5

Table 5.1.1. The partitions of the set $B = \{s, y, z\}$, their decimal indicators and their ordering.

5.2 THE GENERALIZED DECOMPOSITION METHOD

The generalized decomposition method will be presented in the context of terminal reliability. In a later section the method will be applied to the calculation of a variety of reliability measures.

The following probability theorem motivated the development of the generalized decomposition method:

Theorem 5.2.1. Let X be a probability space and let $\{X_1, X_2, \dots, X_n\}$ be a partition of X into n mutually exclusive events with $p(X_j) \neq 0$, $j=1, 2, \dots, n$. Then the probability of any event A can be expressed as:

$$p(A) = \sum_{j=1}^n p(A/X_j) p(X_j) \quad (5.2.2.)$$

Suppose that the terminal reliability is to be calculated for a pair of nodes s and t in a given reliability graph with b branches. Let X denote the set of the 2^b b -tuple minterms x of this graph. Then, as explained in 2.3, the set X can be viewed as a probability space with the probability $p(x)$, of any minterm x , defined as the product of the probabilities of occurrence of the individual component states indicated by the minterm x . Let A denote the set (event) of minterms x which are applicable with respect to the terminal reliability R_{st} . Then the probability $p(A)$ is, precisely, the terminal reliability R_{st} . We shall try now to partition the set X into a number of pairwise disjoint sets (mutually exclusive events) of non-zero

probability, in order to apply equation 5.2.2. for the calculation of R_{st} . Naturally, this partition should be done in such a way so as to result in computational savings.

Consider any subgraph S_1 of the given reliability graph and let S_2 denote the remaining part (subgraph) of the graph. Let b_k be the number of branches in subgraph S_k and let X^k be the set of the 2^{b_k} b_k -tuple minterms of subgraph S_k , $k=1,2$. It should be clear that $b_1+b_2=b$ and that the 2^{b_1} b_1 -tuple minterms x^1 , each combined (concatenated) with each of the 2^{b_2} b_2 -tuple minterms x^2 , produce the $2^{b_1} \cdot 2^{b_2} = 2^b$ minterms x of the set X . Thus, one can derive a partition of X by combining each member of a partition of X^1 with the set X^2 . The process is shown, schematically, in Figure 5.2.5.

Let B_1 be the minimal S_1 -boundary. Then every minterm x^1 defines a partition of the set B_1 as follows: all nodes of B_1 which are connected together through the ON components of x^1 belong to the same member of the partition. Let h_1 denote the number of distinct partitions of the set B_1 obtained, in this manner, from the minterms of the set X^1 . It should be clear that each minterm x^1 defines one and only one of the h_1 partitions. Hence, if we group together all minterms x^1 which define the same partition of the set B_1 , we obtain a partition of the set X^1 into h_1 pairwise disjoint subsets $X_1^1, X_2^1, \dots, X_{h_1}^1$. Next, combining each of the sets X_i^1 , in this partition, with the set X^2 , we obtain a partition of the set X into h_1 pairwise disjoint sets $X_1, X_2, \dots, X_{h_1}$. It should be observed that, because of the way each X_i was formed, we have:

$$p(X_i) = p(X_i^1), \quad i=1,2,\dots,h_1 \quad (5.2.3.)$$

On the other hand, since every X_i^1 contains at least one minterm its probability is different than zero.

We have, thus, obtained a partition of the set X of all minterms x (probability space) into h_1 pairwise disjoint sets (mutually exclusive events) $X_1, X_2, \dots, X_{h_1}$, each with non-zero probability. Since A denotes the set (event) of minterms x which are applicable with respect to R_{st} , we have: $p(A) = R_{st}$. Similarly, $p(A/X_i)$ is equal to the terminal reliability between nodes s and t , given that the event X_i has occurred. We shall denote this terminal reliability as R_{st}^i . Hence, equation 5.2.2, translated into the language of reliability, yields:

$$R_{st} = \sum_{i=1}^{h_1} R_{st}^i p(X_i)$$

For convenience, let $p(X_i^1) = P_i^1$. Then this last equation combined with equation 5.2.3 yields:

$$R_{st} = \sum_{i=1}^{h_1} P_i^1 R_{st}^i \quad (5.2.4.)$$

The terminal reliability R_{st}^i is calculated in the original reliability graph given that the event X_i has occurred, that is, given that the nodes in each

member of the i -th partition of B_1 are connected together in subgraph S_1 .

Thus, terminal reliability R_{st}^i , $i=1,2,\dots,h_1$, is calculated as follows:

- (1) In each member of the i -th partition of the set B_1 , combine all nodes and treat them as a single node.
- (2) Compute the terminal reliability between nodes s and t in subgraph S_2 as modified by step (1); the result of this calculation is the terminal reliability R_{st}^i .

Equation 5.2.4 is the fundamental equation of the generalized decomposition method. The conditional probability technique described in 2.4 as well as the 2-part decomposition method discussed in 3.3, can be derived as special cases from this equation. The process is outlined below.

Let us consider a branch joining nodes y and z of the reliability graph. This branch along with nodes y and z can be considered as subgraph S_1 of the reliability graph. Since this subgraph consists of only one branch, there are only two minterms for S_1 , that is, $X^1 = \{1, 0\}$. The minimal S_1 -boundary is: $B_1 = \{y, z\}$. The first minterm in X^1 defines the partition $\{\{y, z\}\}$ of B_1 with decimal indicator $\{3\}$; the second minterm in X^1 defines the partition $\{\{y\}, \{z\}\}$ of B_1 with decimal indicator $\{2, 1\}$. Thus, $X_1^1 = \{1\}$ and $X_2^1 = \{0\}$ with $p(X_1^1) = r$ and $p(X_2^1) = 1-r$. Applying equation 5.2.4. we obtain

$$R_{st} = R_{st}^1 r + R_{st}^2 (1-r)$$

which is, precisely, the fundamental equation of the conditional probability technique (equation 2.4.2).

Let us now consider a 2-part partition of the reliability graph using a minimal s - t boundary N_1 . To simplify matters, suppose that this boundary contains only two nodes y and z , that is, $N_1 = \{y, z\}$. As explained in 3.3, part 1 of the partition is the subgraph containing node s . This part can be considered as subgraph S_1 in the application of the generalized decomposition method. The minimal S_1 -boundary is: $B_1 = \{s, y, z\}$, that is, it consists of the set N_1 together with node s . The five possible partitions of the set B_1 are listed in Table 5.1.1, along with their decimal indicators. It should be realized that R_{st}^i is zero for all those partitions in which s alone is a member of the partition. This happens because the subgraph S_1 was chosen in a very special manner, namely, using a minimal s - t boundary. Thus, of the five partitions listed in Table 5.1.1 only the three first need be considered. Applying equation 5.2.4 we obtain:

$$R_{st} = \sum_{i=1}^3 R_{st}^i p(X_i^1)$$

It should be easy to verify that this is, precisely, the fundamental equation obtained in the 2-part decomposition method for $\mu_1 = 2$ (equation 3.3.10.b).

In conclusion, the method presented in this section allows the decomposition of the reliability graph in a very general manner. Moreover, it produces always the exact value of terminal reliability. The conditional

probability technique described in 2.4, as well as, the 2-part decomposition method presented in 3.3 can be derived as special cases of the generalized decomposition method. The details of the application of the method will be discussed in the following section.

Figure 5.2.5. Partitioning the sample space X through a partition of the set X^1 .

5.3 APPLYING THE GENERALIZED DECOMPOSITION METHOD

The purpose of this section is to illustrate the concepts introduced in the previous section. We shall do so by using the fundamental equation 5.2.4 to calculate the terminal reliability R_{st} in the graph of Figure 5.3.2. It should be noted that this is the same graph as the one used in section 3.4 to illustrate the 2-part decomposition method.

If we choose subgraph S_1 to be the graph defined by nodes a, b and c and branches 3, 4 and 5, then, as shown in Figure 5.3.2, the remaining part of the graph (subgraph S_2) is quite simple. In order to compute the probabilities P_i^1 we need first to know the sets x_i^1 . This is done by constructing a matrix M_1 , called the subgraph-matrix. Let b_1 be the number of branches in the subgraph S_1 and v_1 the number of nodes in the minimal S_1 -boundary B_1 ; in our example, we have: $b_1 = 3$, $B_1 = \{a, b, c\}$ and $v_1 = 3$. The subgraph-matrix M_1 is a $2^{b_1} \times v_1$ matrix defined as follows:

- (1) The rows are indexed by the $2^{b_1} b_1$ -tuple minterms x^1 of the set x^1 and the columns by the v_1 nodes in the set B_1 .
- (2) The ℓ -th entry of a column is the decimal indicator of that set of nodes in B_1 which are connected to the column heading through the ON components of the ℓ -th minterm x^1 . By convention, the column heading is connected to itself.

The matrix M_1 for our example, is the one shown in Figure 5.3.3. The partition of B_1 induced by the ℓ -th minterm x^1 is obtained as follows: its decimal indicator is the set of decimal indicators appearing in the

l -th row of the matrix M_1 . The partitions of $B_1 = \{a, b, c\}$ generated by the matrix M_1 of our example are shown in Figure 5.3.3. Grouping together the row-headings in M_1 which correspond to the same partition of B_1 , we obtain the sets X_i^1 . These sets are shown in Table 5.3.4, which includes the calculations involved in evaluating R_{st} from equation 5.2.4. It is interesting to note that all reliability calculations in subgraph S_2 were carried out using series-parallel arguments.

The main steps followed in the calculation of R_{st} can be summarized as follows:

- (1) Selection of a subgraph S_1 .
- (2) Construction of the subgraph matrix M_1 .
- (3) Formation of the sets X_i^1 and calculation of their probabilities $P_i^1, i = 1, 2, 3, 4, 5$.
- (4) Calculation of the terminal reliability R_{st}^i in subgraph S_2 , as modified according to the partition of B_1 corresponding to the set $X_i^1, i = 1, 2, 3, 4, 5$.

It should be clear that steps (2) and (4) are the most time consuming. For, each column of the matrix M_1 requires 2^{b_1} path-tracing searches in order to determine the entries, that is, a total of $v_1 2^{b_1}$ path-tracing searches are required for the construction of M_1 . On the other hand, the calculation of R_{st}^i requires 2^{b_2} path-tracing searches, $i = 1, 2, \dots, h_1$. Thus, a total of $h_1 2^{b_2}$ path-tracing searches are required in subgraph S_2 . The total number of path-tracing searches required for the calculation of R_{st} will be referred to

as the complexity of the generalized decomposition method and it will be denoted as K_2 . Thus,

$$K_2 = v_1 2^{b_1} + h_1 2^{b_2} \quad (5.3.1.)$$

^oThe number v_1 is under our control and can be kept small. The number h_1 , however, varies widely depending on the size and configuration of subgraph S_1 . It can be as big as the number d_1 of all possible partitions of the set B_1 ; this was the case in the example of Figure 5.3.2. Unfortunately, the number d_1 grows very rapidly with v_1 , as shown in Table 5.3.5., an indication that the complexity K_2 may also increase rapidly with v_1 . Is there any way to control the growth of this complexity? The answer must lie in the choice of subgraph S_1 . For instance, when this subgraph consists of 4 branches then h_1 can be, at most, equal to $2^4 = 16$; on the other hand, if these 4 branches connect $v_1 = 5$ nodes, then the number of possible partitions of these nodes is 52. As the number v_1 increases, the number d_1 increases much more rapidly than 2^{v_1} . Suppose then that subgraph S_1 is chosen so that $b_1 \approx v_1$. According to equation 5.3.1., the complexity K_2 becomes bounded by the number

$$v_1 2^{b_1} + 2^{v_1} 2^{b_2}$$

That is, the complexity of the generalized decomposition method increases exponentially with the number v_1 . In view of the fact that it is not always possible to choose subgraph S_1 so that $b_1 \approx v_1$, an exponential increase of the complexity seems to be the best one can obtain with the generalized

decomposition method.

In conclusion, the generalized decomposition method presents a computational complexity which increases exponentially with the number of boundary nodes. Thus it compares unfavorably with the 2-part decomposition method whose complexity increases only linearly with the number of boundary nodes. However, the generalized decomposition method presents the advantage that it always yields the exact value of terminal reliability. Moreover, it can be applied in the calculation of a variety of reliability measures, as it is explained in the following section.

Figure 5.3.2. The decomposition of a given reliability graph into subgraphs S_1 and S_2 .

M_1			Induced partitions of the set B_1	
3 4 5	a b c	Decimal indic.	Order	
0 0 0	4 2 1	{ 1, 2, 4 }	5	
0 0 1	4 3 3	{ 3, 4 }	4	
0 1 0	6 6 1	{ 1, 6 }	2	
0 1 1	7 7 7	{ 7 }	1	
1 0 0	5 2 5	{ 2, 5 }	3	
1 0 1	7 7 7	{ 7 }	1	
1 1 0	7 7 7	{ 7 }	1	
1 1 1	7 7 7	{ 7 }	1	

Figure 5.3.3. The subgraph matrix M_1 and the induced partitions of the set B_1 .

Subgraph S_1		Subgraph S_2		Computing R_{st}
i	X_i^1	P_i^1	The modified S_2	
1	0 1 1	$.3 \times .9 \times .9 = .243$		$P_i^1 R_{st}^1$ $.7907328$
	1 0 1	$.7 \times .1 \times .9 = .063$		
	1 1 0	$.7 \times .9 \times .1 = .063$		
	1 1 1	$.7 \times .9 \times .9 = .567$		
		$.936$		
2	0 1 0	$.3 \times .9 \times .1 = .027$		$.0206064$
3	1 0 0	$.7 \times .1 \times .1 = .007$		$.0040320$
4	0 0 1	$.3 \times .1 \times .9 = .027$		$.142560$
5	0 0 0	$.3 \times .1 \times .1 = .003$		$.0010800$

$$R_{st} = .8307072$$

Table 5.3.4. The calculations involved in evaluating R_{st} in the graph of Figure 5.3.2.

v_1	2^{v_1}	d_1
2	4	2
3	8	5
4	16	15
5	32	52
6	64	203
7	128	877
8	256	4140
9	512	21147
10	1024	115975

Table 5.3.5. The number d_1 of all possible partitions of a set with v_1 elements.

Note: The number d_1 was computed using the formulas provided in C. Berge (1971).

5.4 RELIABILITY MEASURES AS EXPECTATIONS OF RANDOM VARIABLES

The generalized decomposition method was developed in the context of terminal reliability. As explained in 2.3, on the other hand, terminal reliability can be viewed as the expectation of a random variable. In this section, we shall use this point of view in order to show that the generalized decomposition method applies to the computation of a variety of reliability measures.

Let us recall that the set X of all minterms of the reliability graph is a probability space. The probability of each minterm ("point") x in this space, is the product of the probabilities of occurrence of the individual component states indicated by the minterm. Referring to two specific nodes s and t of the reliability graph one can define a random variable f on the space X as follows:

$$f(x) = 1, \text{ if } x \text{ is an applicable minterm}$$

$$= 0, \text{ otherwise}$$

It should be clear that the expectation of this random variable is, precisely, the terminal reliability R_{st} . That is,

$$E(f) = \sum_{x \in X} p(x)f(x) = R_{st} \quad (5.4.1.)$$

Using the generalized decomposition method, this terminal reliability was expressed as:

$$R_{st} = \sum_{i=1}^{h_1} p(X_i^1) R_{st}^i \quad (5.4.2.)$$

where R_{st}^i is calculated in the subgraph S_2 , as modified according to the partition of B_1 induced by X_i^1 . On the set of minterms of this modified subgraph S_2 , one can define a random variable g^i as follows:

$$g^i(x^2) = 1, \text{ if } x^2 \text{ is an applicable minterm} \\ = 0, \text{ otherwise}$$

Then, the terminal reliability R_{st}^i is equal to the expectation of g^i and, using equations 5.4.1. and 5.4.2., we obtain:

$$E(f) = \sum_{i=1}^{h_1} p(X_i^1) E(g^i) \quad (5.4.3.)$$

Let us study closer this last equation. Referring back to the definition of f , the expectation $E(f)$ is, precisely, the probability of the set of applicable minterms in X . Similarly, the expectation $E(g^i)$ is the probability of the set of applicable minterms x^2 of the (modified) subgraph S_2 . Let us define a minterm to be applicable whenever it possesses a specific property.

An example of such a property is: the ON components join together two specified nodes s and t of the reliability graph. In this case, the expectation $E(f)$ was called terminal reliability and it was denoted by R_{st} . This process can be extended to other properties of minterms as well. In each case, the expectation $E(f)$ represents the corresponding reliability measure, i.e., the probability of the set of applicable minterms, and it can be computed using equation 5.4.3. We shall illustrate the process using an example.

Consider the following property of minterms: the ON components join together all the nodes of the reliability graph which belong to a specified set V . The corresponding reliability measure is the probability of having all nodes in V joined together. This probability is, precisely, the expectation $E(f)$ and it can be computed from equation 5.4.3. It is interesting to note that if V contains only two nodes then the corresponding measure is a terminal reliability. Similarly, if the set V contains all nodes, then the corresponding reliability measure is the probability that all nodes of the reliability graph are joined together.

The random variable f , as defined above, indicates whether or not a minterm x possesses a specific property. Considering various properties of minterms, one can generate from f a number of reliability measures. These measures are represented by $E(f)$ and they can be computed from equation 5.4.3. Let us now consider a random variable y which assigns a quantitative characteristic to each minterm x , not necessarily equal to 0 or 1. As an example, consider

the case where $g(x)$ denotes the number of nodes, in the reliability graph, which are joined together by the ON components of the minterm x . The expectation $E(g)$ is then the expected number of nodes which are connected together in the reliability graph. This is a reliability measure which has received much attention in the literature, especially in the context of computer-communication networks. Methods for its calculation have been presented by R. Van Slyke and H. Frank (1972). We shall show here that equation 5.4.3 can be used for the calculation of the cumulative distribution function Y of the random variable g .

Let n be the number of nodes in the reliability graph and let λ be an integer such that $2 \leq \lambda \leq n$. Then $Y(\lambda)$ denotes the probability that λ or fewer nodes of the reliability graph are joined together. Consider now the following property of minterms: the ON components join together λ or fewer nodes of the reliability graph. Then the corresponding reliability measure obtained from f is, precisely, $Y(\lambda)$ and can be computed using equation 5.3.4. Thus, one can calculate the distribution function Y by applying equation 5.4.3 for different values of λ .

In conclusion, the generalized decomposition method can be applied to the calculation of a variety of reliability measures. What made this possible, was the fact that terminal reliability can be viewed as the expectation of a random variable. Since many reliability measures can be viewed in the same manner, the generalized decomposition method should prove a powerful tool for their calculation.

5.4 DISCUSSION AND CONCLUSIONS

The aim of this chapter was to generalize the conceptual basis of the decomposition algorithm. Based on a probability theorem, the generalized method was developed. This method allows complete freedom in the decomposition of the reliability graph. Its complexity, however, increases exponentially with the number of boundary nodes.

If the decomposition of the graph is done in a special manner, namely, using minimal s - t boundaries, then the complexity is reduced drastically. This was done in the 2-part decomposition method where the complexity was shown to increase only linearly with the number of boundary nodes. This resulted in the significant computational savings observed in the application of the decomposition algorithm.

In terms of complexity, the generalized decomposition compares unfavorably with its special version. However, it presents two advantages; first, it always yields the exact value of terminal reliability; second, it can be extended to the calculation of a variety of reliability measures.

CHAPTER SIX

CONCLUDING DISCUSSION

The purpose of this chapter is: to summarize and relate the salient features of the foregoing chapters, to comment on the significance of the results to reliability engineering and to make suggestions for further work.

6.1 SUMMARY AND CONCLUSIONS

This thesis is concerned with reliability computations for systems consisting of components which exhibit independent discrete operating modes, by means of an innovative interpretation of a mathematical model. The resulting decomposition concept leads to powerful and efficient computational procedures.

As was emphasized at the beginning of the thesis, proposals for reliability measures must be considered in conjunction with the pragmatics of their computation. In this context, the general aim of the thesis was to create a new approach which would yield tractable methods for computing system reliability. Each topic covered contributes to this theme; chapter 1 establishes the reliability design context by introducing the reliability allocation problem; chapter 2 discusses an important reliability measure, namely, terminal reliability, and reviews known methods for its computation; chapter 3 applies a 2-part decomposition technique to the computation of terminal reliability in small systems, and discusses ways for reducing computational complexity; chapter 4 explains the recursive application of the 2-part decomposition in large systems, and discusses the results of computational experience; the techniques presented in chapters 3 and 4 are viewed as a special case of a generalized decomposition method presented in chapter 5 which also discusses the application of the method to a variety of reliability measures.

The formulation and development throughout the thesis are kept as general as possible; the results can be applied to evaluate the reliability of any

system which can be modeled by a graph in which:

- (1) Each component (node or branch) has two states, one good and one bad.
- (2) Component failures are statistically independent.

The principle behind the original derivations here is basically simple: the reliability graph is decomposed into a number of smaller subgraphs and the reliability measure is obtained from reliability calculations in the subgraphs. The mathematical basis is provided by a general probability theorem concerning the decomposition of a probability space into a number of mutually exclusive events. This theorem, translated into the language of reliability, induces a decomposition of the reliability problem into a number of simpler subproblems.

It has been shown that the decomposition techniques presented in the thesis lead to results consistent with those obtained by other methods. Further, these techniques present lower computational complexities and, thus, allow reliability computations to be extended to larger systems. In particular, the decomposition technique discussed in chapters 3 and 4 presents a complexity which grows only linearly with the number of boundary nodes. As a result, its computer implementation showed substantial computational savings relative to recently published analyses. The criteria for the application of the technique to certain highly connected graphs are sometimes impossible to satisfy. However, the sparsity exhibited by practical systems does not often limit the use of the technique. The generalized decomposition technique developed in chapter 5, is applicable to any graph. The computational complexity of the resulting algorithm, however,

grows exponentially rather than linearly with the number of boundary nodes.

By extending reliability computations to larger systems, the thesis provides an important tool in designing and analyzing a large class of systems in a reliability context. The examples used in evaluating computation times were taken from the computer communication environment. Reliability considerations in this field are of compelling current importance and algorithms and computation results are readily available for comparisons. The results obtained from the decomposition algorithm suggest that reliability analysis can take a place with throughput analysis in iterative design procedures, in systems much larger than the ARPA network.

6.2 SUGGESTIONS FOR FURTHER WORK

The techniques presented in this thesis are capable of handling larger graphs than one usually encounters in modern applications. A significant problem in utilizing the computer program is the input and modification of data during the iterative process of evaluating and altering the configuration of a system. This leads to the possibility of a graphics terminal which would allow light-pen interaction with a display, suitably programed to permit the designer to manipulate the parameters during the design procedure.

The decomposition algorithm developed in chapter 4 presents a complexity which grows only linearly with the number of boundary nodes. As a result, this algorithm was shown to be applicable to sparse irreducible graphs with up to forty branches. On the other hand, the generalized decomposition technique developed in chapter 5 applies to any graph rather than to only sparse graphs. However, its complexity grows exponentially with the number of boundary nodes and this makes the technique applicable to relatively smaller graphs.

As practical systems grow in size and complexity of configuration so do the computation times in the application of the decomposition techniques. And, in view of the ever growing pace of sophistication and automation in modern technology, it is reasonable to expect that practical systems will outgrow the capabilities of these techniques in the not too distant future. The question is: is there any room for improvement or adjustment of the decomposition techniques to these growing needs? There seem to be two different directions where one may turn for an answer.

The first is an algebraic rather than graph-theoretical approach. Recent developments in new algebras for describing and manipulating large binary arrays, recently presented by E. Ulug and B.A. Bowen (1974), may be applicable -- with suitable modifications -- to the computation of reliability measures. The close relation which exists between algebraic and graph-theoretical concepts, in the reliability context, are discussed in a recent paper by J.B. Fussell, G.J. Powers and R.G. Bennetts (1974), in connection with fault trees.

Second, let us recall that the conceptual basis of these techniques is the decomposition concept presented -- in its general form -- in chapter 5: the reliability graph is decomposed into a number of smaller graphs and system reliability is obtained from reliability calculations in the smaller graphs. The interesting feature of the procedure is that one is quite free to choose the configuration of these smaller graphs. This, in turn, may allow a pattern recognition approach to the reliability problem in conjunction with the decomposition techniques presented in this thesis. A pattern recognition approach has already been suggested by B.V. Koen and A. Carnino (1974) for the reduction of logic trees.

In order to gain calculational advantage, once a specific configuration has been evaluated, it can be stored and reused as needed. This implies the recognition of known patterns (i.e., standard configurations) among nodes. There are, however, three basic criteria for a pattern recognition approach to be interesting:

- (1) It must be more economical to identify a specific pattern than it is to recalculate it each time it is needed.
- (2) Rarely occurring structures must not be stored since they can quickly render retrieval impractical unless an efficient enough algorithm is used.
- (3) The largest possible patterns in each problem should be identified in each case in order to reduce the size of the problem as quickly as possible.

A dynamic pattern library may further increase the efficiency of a pattern recognition approach. That is, if a supplementary calculation evaluates a new pattern, that new pattern ought to be stored for later use.

Thus, "learning theory" combined with pattern recognition offers the potential of maintaining the capability of analyzing and predicting the performance of larger and more complex systems.

APPENDIX: PROOFS OF THEOREMS

Lemma 3.3.2. The collection $\{x_{1u_1}^1 : u_1 = 0, 1, \dots, \ell_1\}$ is a partition of the set x^1 .

Proof: According to definition 3.3.1., two different members in the collection correspond to two different values of the index u_1 , say, i and j . Now, the ON components of every $x_{1i}^1 \in x_{1i}^1$ indicate connection to every node in the set i only. Hence, x_{1i}^1 and x_{1j}^1 have no common minterms, that is, $x_{1i}^1 \cap x_{1j}^1 = \emptyset$. On the other hand, every minterm of x^1 belongs to some set $x_{1u_1}^1$, for an appropriate choice of u_1 . Thus, $x_{10}^1 \cup x_{11}^1 \cup \dots \cup x_{1\ell_1}^1 = x^1$. Hence, the collection above is a partition of x^1 .

Lemma 3.3.3. The collection $\{x_{u_1} : u_1 = 0, 1, \dots, \ell_1\}$ is a partition of the set x .

Proof: Two different members of the collection correspond to two different values of the index u_1 , say, i and j . According to lemma 3.3.2., $x_{1i}^1 \cap x_{1j}^1 = \emptyset$ and this implies that $x_i \cap x_j = \emptyset$ (because of the way x_{u_1} is formed from $x_{u_1}^1$). On the other hand, every minterm of x belongs to some set x_{u_1} , for an appropriate choice of u_1 . Thus, $x_0 \cup x_1 \cup \dots \cup x_{\ell_1} = x$. Hence, the collection above is a partition of x .

Theorem 3.3.5. Let $x \in x_{u_1}$ and $\mu_1 \leq 2$. Then, x is an applicable minterm with respect to R_{st} , if and only if, part x^2 of x has the following property:

the ON components contain a path from at least one node in the set u_1 to node t .

Proof: Since the case $\mu_1 = 1$ is trivial, we shall only prove the theorem for $\mu_1 = 2$.

A. Let x be an applicable minterm. Then the ON components of x contain a path joining node s to node t . On the other hand, part x^1 of x belongs to the set $x_{u_1}^1$, that is, the ON components of x^1 contain a path joining s to

every node in u_1 and to no node in $\bar{u}_1$. If u_1 contains one node then it is this node that must be connected to node t through the ON components of part x^2 of x . For if not, then the node in $\bar{u}_1$ is connected to the node in u_1 through the ON components of x^1 ; this is a contradiction, since x^1 belongs to $x_{u_1}^1$. If u_1 contains both boundary nodes then the result follows immediately.

B. Let part x^2 of x have the following property: the ON components contain a path from at least one node in the set u_1 to node t . Part x^1 of x , on the other hand, belongs to the set $x_{u_1}^1$ and, thus, its ON components contain paths joining node s to every node in u_1 . Hence, the ON components of x contain a path joining node s to node t , that is x is an applicable minterm.

Theorem 3.5.2. Given a positive real number b , let b_1, b_2 be two positive real numbers such that $b_1 + b_2 = b$. Then for every given positive real number μ_1 , the quantity: $C_2 = 2^{b_1 + \mu_1 2^{b_2}}$ becomes minimum when: $b_1 = \frac{1}{2}(b + \log_2 \mu_1)$ and $b_2 = \frac{1}{2}(b - \log_2 \mu_1)$.

Proof: For convenience, let: $y = 2^{b_1}$ and $z = \mu_1 2^{b_2}$. Then,

$$yz = 2^{b_1} \cdot \mu_1 2^{b_2} = \mu_1 2^{b_1 + b_2} = \mu_1 2^b$$

On the other hand we have:

$$C_2 = 2^{b_1 + \mu_1 2^{b_2}} = y + z = (y - z)^2 + 2y$$

$$= (y - z)^2 + \mu_1 2^b$$

Since $\mu_1 2^b > 0$, the quantity C_2 becomes minimum when $y = z$, that is, when: $2^{b_1} = \mu_1 2^{b_2}$, or, equivalently, when $b_1 = b_2 + \log_2 \mu_1$. Since $b_1 + b_2 = b$, we obtain:

$$b_1 = \frac{1}{2}(b + \log_2 \mu_1), \quad b_2 = \frac{1}{2}(b - \log_2 \mu_1)$$

Theorem 4.2.2. Let m, M be integers greater than or equal to 2 and let b be a positive real number. Let $\mu_1, \mu_2, \dots, \mu_{m-1}$ be integers and $b_1, b_2, \dots, b_m$ positive reals, such that:

$$\mu_1 \mu_2 \dots \mu_{m-1} = M, \text{ and } \mu_k \geq 2, \quad k=1, 2, \dots, m-1$$

$$b_1 + b_2 + \dots + b_m = b$$

Then the quantity:

$$C_m = \sum_{k=1}^m \mu_{k-1} 2^{b_k}$$

becomes minimum, when:

$$b_k = \frac{1}{m}(b + \log_2 M) - \log_2 \mu_{k-1}$$

Proof: For convenience, let $z_k = \mu_{k-1} 2^{b_k}$, $k = 1, 2, \dots, m$. Then,

$$\begin{aligned}
 z_1 z_2 \dots z_m &= 2^{b_1} (\mu_1 2^{b_2}) \dots (\mu_{m-1} 2^{b_m}) \\
 &= (\mu_1 \mu_2 \dots \mu_{m-1}) 2^{b_1 + b_2 + \dots + b_m} \\
 &= M 2^b
 \end{aligned}$$

On the other hand, we have:

$$\begin{aligned}
 C_m &= \sum_{k=1}^m \mu_{k-1} 2^{b_k} \\
 &= \sum_{k=1}^m z_k
 \end{aligned}$$

Thus, we wish to minimize the function $z_1 + z_2 + \dots + z_m$ over all non-negative z_i satisfying the condition $z_1 z_2 \dots z_m = M 2^b$. Since the minimum clearly is not assumed at a boundary point, we can utilize the Lagrange-multiplier approach to determine the local minima. For the function

$$f(z_1, z_2, \dots, z_m) = z_1 z_2 \dots z_m - \lambda (z_1 + z_2 + \dots + z_m)$$

the variational equations

$$\frac{\partial f}{\partial z_i} = \frac{z_1 z_2 \dots z_m}{z_i} - \lambda = 0, \quad i = 1, 2, \dots, m$$

yield the result that $z_1 = z_2 = \dots = z_m$. From this we readily see that

$$z_i = (M2^b)^{1/m}, i = 1, 2, \dots, m$$

is the unique minimizing point.

Hence, the quantity C_m becomes minimum, when

$$\mu_{k-1} 2^{b_k} = (M2^b)^{1/m}$$

or

$$b_k = \frac{1}{m} (b + \log_2 M) - \log_2 \mu_{k-1}$$

Q.E.D.

Theorem 5.2.1 Let X be a probability space and let $\{X_1, X_2, \dots, X_n\}$ be a partition of X into n mutually exclusive events with $p(X_j) \neq 0$, $j=1, 2, \dots, n$. Then, the probability of any event A can be expressed as:

$$p(A) = \sum_{j=1}^n p(A/X_j) p(X_j)$$

Proof: Since the collection $\{X_1, X_2, \dots, X_n\}$ is a partition of X , we have:

$$X = X_1 \cup X_2 \cup \dots \cup X_n$$

and

$$X_l \cap X_m = \emptyset, \text{ for } l \neq m.$$

Hence, using set manipulations, we obtain:

$$\begin{aligned} A &= A \cap X \\ &= A \cap (X_1 \cup X_2 \cup \dots \cup X_n) \\ &= (A \cap X_1) \cup (A \cap X_2) \cup \dots \cup (A \cap X_n) \end{aligned}$$

and

$$\begin{aligned} (A \cap X_l) \cap (A \cap X_m) &= A \cap (X_l \cap X_m) \\ &= \emptyset, \text{ for } l \neq m \end{aligned}$$

Thus, the collection $\{A \cap X_1, A \cap X_2, \dots, A \cap X_n\}$ is a collection of mutually exclusive events. This allows the probability of A to be written as:

$$p(A) = \sum_{j=1}^n p(A \cap X_j)$$

On the other hand, if $p(X_j) \neq 0$, then the conditional probability $p(A/X_j)$ is defined as:

$$p(A/X_j) = p(A \cap X_j) / p(X_j)$$

That is,

$$p(A \cap X_j) = p(A/X_j)p(X_j)$$

Hence,

$$\begin{aligned} p(A) &= \sum_{j=1}^n p(A \cap X_j) \\ &= \sum_{j=1}^n p(A/X_j)p(X_j), \quad \text{Q.E.D.} \end{aligned}$$

REFERENCES

- Banerjee S.K., K. Rajamani, "Parametric representation of probability in two dimensions - a new approach in system reliability evaluation", IEEE Trans. on Rel., Vol. R-21, Feb. 1972, pp. 56-60.
- Barlow R.E., Proschan F., "Mathematical theory of reliability", John Wiley and Sons, Inc., New York 1965.
- Bazovsky I., "Reliability theory and practice", Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1961.
- Berge C., "Principles of combinatorics", Academic Press, New York and London, 1971.
- Bertelè U., F. Brioschi, "Nonserial dynamic programming", Academic Press, New York and London, 1972.
- Bertziss A.T., "Data structures, theory and practice", Academic Press, New York, 1971.
- Brown D.B., "A computerized algorithm for determining the reliability of redundant configurations", IEEE Trans. on Rel., Vol. R-20, Aug. 1971, pp. 121-124.
- Cavers J.K., "Cutset manipulations for communication network reliability estimation", to appear in the IEEE Trans. on Commun. (1975).
- Christiaanse W.R., "A technique for the analysis of repairable redundant systems", IEEE Trans. on Rel., Vol. R-19, May 1970, pp. 53-60.

deMercado J., N. Spyrtos, B.A. Bowen, "A method for the exact calculation of network reliability", to appear in the IEEE Trans. on Rel. (1975).

Esary J.D., F. Proschan, "Coherent structures of non-identical components", Technometrics, Vol. 5, No. 2, 1963, pp. 191-209.

Frank H., W. Chou, "Topological optimization of computer networks", Proceedings of the IEEE, special issue on computer communications, Nov. 1972, pp. 1385-1397.

Fratta L., U. Montanari, "All simple paths in a graph by solving a system of linear equations", Consiglio Nazionale delle Ricerche, Istituto di Elaborazione dell' Informazione, Nota Interna B71-11, Oct. 1971.

Fratta L., U.G. Montanari, "A Boolean algebra method for computing the terminal reliability in a communication network", IEEE Trans. on Circuit Theory, Vol. C7-20, No. 3, May 1973, pp. 203-211.

Fu Y., S.S. Yau, "A note on the reliability of communication networks", SIAM Journal 10, 1962, pp. 469-474.

Fussell J.B., G.J. Powers, R.G. Bennetts, "Fault trees - A state of the art discussion", IEEE Trans. on Rel., Vol. R-23, No. 1, April 1974, pp. 51-55.

Hinsler E., "A fast recursive algorithm to calculate the reliability of a communication network", IEEE Trans. on Commun., Vol. COM-20, June 1972, pp. 637-640.

- Hänsler E., G.K. McAuliffe, R.S. Wilkov, "Exact calculation of computer network reliability", *Networks*, 4, 1974 by John Wiley and Sons, Inc., pp. 95-112.
- Henley E.J., R.A. Williams, "Graph theory in modern engineering", Academic Press, New York and London, 1973.
- Jensen P.A., M. Bellmore, "An algorithm to determine the reliability of a complex system", *IEEE Trans. on Rel.*, Vol. R-18, Nov. 1969, pp. 169-174.
- Kershenbaum A., R.M. Van Slyke, "Recursive analysis of network reliability", *Networks*, 3, 1973, by John Wiley and Sons, Inc., pp. 81-94.
- Ketelle J.D. Jr., "Least cost allocation of reliability investment", *Operations Research*, Vol. 10, No. 2, 1962, pp. 249-265.
- Kim Y.H., K.E. Case, P.M. Ghare, "A method for computing complex system reliability", *IEEE Trans. on Rel.*, Vol. R-21, No. 4, Nov. 1972, pp. 215-219.
- Koen B.V., A. Carnino, "Reliability calculations with a list processing technique", *IEEE Trans. on Rel.*, Vol. R-23, No. 1, April 1974, pp. 43-50.
- Krishnamurthy E.V., G. Komissar, "Computer-aided reliability analysis of complicated networks", *IEEE Trans. on Rel.*, Vol. R-21, No. 2, May 1972, pp. 86-89.

Lloyd D.K., M. Lipow, "Reliability: Management, methods and mathematics",
Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1962.

Maestrini R., R. Gentilucci, "Fiabilite des systemes et equipements de
telecommunications", Proceedings of a NATO Conference held at
Turin, Italy, June 24-July 4, 1969 (Editor: D. Grouchka).

Mine H., "Reliability of physical systems", IRE Trans. on Circuit Theory,
Vol. CT-6, May 1959, pp. 138-151.

Misra K.B., "An algorithm for the reliability evaluation of redundant networks",
IEEE Trans. on Rel., Vol. R-19, 1970, pp. 146-151.

Misra K.B., T.S.M. Rao, "Reliability analysis of redundant networks", IEEE
Trans. on Rel., Vol. R-19, 1970, pp. 19-24.

Nelson A.C. Jr., J.R. Batts, R.L. Beadles, "A computer program for approximating
system reliability", IEEE Trans. on Rel., Vol. R-19, May 1970,
pp. 61-65.

Okada S., "Topology applied to switching circuits", Proceedings of the
symposium on information networks, Polytechnic Institute of
Brooklyn, New York, April 1954, pp. 267-290.

Shooman M.L., "Probabilistic reliability: An engineering approach", McGraw-
Hill Book Co., New York, 1968.

Ulug M.E., B.A. Bowen, "A unified theory of the algebraic topological methods for the synthesis of combinatorial logic", IEEE Trans. on Computers, Vol. C-23, No. 3, March 1974, pp. 255-267.

Van Slyke R., Frank H., "Network reliability analysis: Part I", Networks, 1, 1972, by John Wiley and Sons, Inc., pp. 279-290.

Wilkov R.S., "Analysis and design of reliable computer networks", IEEE Trans. on Commun., Vol. COM-20, June 1972, pp. 660-678.

INDEX

	<u>Page</u>
Applicable minterm	22
Boundary branch	61
Branch probability	97
Complexity	78,98,134
DECOMP01	105
Decomposition Algorithm	98
Disjoint subgraphs	61
Irreducible reliability graph	18
Leaf	90
Minimal s - t boundary	47
S -boundary	121
s - t cut	23
s - t path	23
Minterm	22
Monotonic system	26
m -out-of- n system	16
Non-intersecting boundaries	96

Parallel system	15
Partition of a set	48
Part matrix	71
Product law of reliabilities	14
unreliabilities	15
Reliability function	101
graph	10
tree	91,97
Root	90
Rooted tree	90
Series-parallel reduction	18
Series system	15
Sensitivity	35
Sparse graph	107
Subgraph matrix	132
Terminal reliability	10

END

27 07 76

FIN