University of Alberta

Multi-state System Reliability Analysis and Evaluation

By

Jinsheng Huang ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirement for the degree of Doctor of Philosophy

Department of Mechanical Engineering

Edmonton, Alberta

Fall 2001

0-612-69835-1

University of Alberta

Library Release Form

Name of Author: Jinsheng Huang

Title of Thesis: Multi-state System Reliability Analysis and Evaluation

Degree: Doctor of Philosophy

Year This Degree Granted: 2001

Jinsheng Huang
No.6307, 5325-26 AV, SW
Calgary, AB Canada T3E 6N3

Aug. 17. 2001
Date

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **Multi-state System Reliability Analysis and Evaluation** submitted by **Jinsheng Huang** in partial fulfillment of the requirements for the degree of **Doctor of Philosophy**.

Dr. Mingjian Zuo

for Dr. Way Kuo

Dr. Don Koval

Dr. Peter C. Flynn

Dr. Armann Ingolfsson

Dr. Stanislav Karapetrovic

Aug 17, 2001

Date

Abstract

In the traditional reliability theory, a system or a component is allowed to take only two possible states: working or failed. The binary reliability theory has been well developed and has been used in various industries over the past half century. However, a system or a component may experience more than two levels of performance. The multi-state reliability theory is attracting more and more research attention in the past two decades.

This thesis provides a systematic literature review on both binary and multi-state reliability theories. The multi-state reliability theory is built on the basis of coherent structures in the same way as binary reliability theory. Many concepts in the binary case have been extended to the multi-state case. The literature review focuses on the fundamentals of the multi-state reliability theory, which include multi-state relevancy conditions, binary decomposition methods of multi-state systems, multi-state system modeling techniques, and algorithms for system performance evaluation.

Binary k-out-of-n systems and binary consecutive k-out-of-n systems are commonly used models in engineering practice. Their definitions have been extended to the multi-state case by assuming that the system has a constant k-out-of-n structure or constant consecutive-k-out-of-n structure at all system levels. In this thesis, we propose the definitions of the generalized multi-state k-out-of-n systems and the generalized consecutive k-out-of-n systems. Under the proposed definitions, the systems are allowed to have different structures at different system levels. Such models are more flexible for describing some engineering problems. Algorithms for system performance evaluation or bounding techniques are developed.

Since the state space of a multi-state system is usually complex, it is difficult to evaluate the probabilities for the system to be in different states. An approach for multi-state system performance evaluation is to extend binary algorithms to the multi-

state context. In this thesis, we investigate the internal relationship between binary systems and multi-state systems and then propose a definition of the dominant multi-state system. A dominant system has some binary properties and can be bounded by two binary structures. A dominant system with binary image can be treated like a binary system. Various properties of dominant systems are explored. The proposed definition has excellent potential for multi-state system performance evaluation.

In summary, this thesis contributes to multi-state system modeling technique, system analysis, and performance evaluation. The proposed generalized multi-state k-out-of-n systems and multi-state consecutive k-out-of-n systems are more flexible than the models proposed by other researchers. The definition of the dominant system provides a new tool for classifying multi-state systems and evaluating or bounding the system performance indexes.

Acknowledgements

First and foremost I would like to thank Dr. Mingjian Zuo, my supervisor, for his patience, advice and support throughout the course of this research.

I am very grateful to Dr. Don Koval (Electrical and Computer Engineering), Dr. Peter Flynn (Mechanical Engineering), Dr. Armann Ingolfsson (Finance and Management Science, School of Business) and Dr. Stanislav Karapetrovic (Mechanical Engineering), University of Alberta, for their scholarly guidance and insightful suggestions throughout the development of this thesis. I would like to thank Dr. Way Kuo (Industrial Engineering), Texas A&M University, USA, for taking the time to extensively review my thesis.

I would like to express my appreciation to Professor Dayi Li and Professor Lanqing Gao, Department of Mining Engineering, Beijing University of Science and Technology, Beijing, China, for recommending me to pursue a Ph. D. degree. My sincere thanks also go to Dr. Zhide Fang, Department of Mathematical Science, University of New Orleans, USA, and my friends, Dr. Yan Guo and Ms. Dian Wang.

Above all I am indebted to my wife and my parents for their continued understanding, encouragement and support throughout my quest for knowledge.

Tabel of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Reliability is the branch of quality assurance that deals specifically with functionability upon demand. A commonly quoted definition of reliability is *the probability of a system or a component performing its intended function adequately for the period of time intended under stated conditions* (Koval, 1996). In this definition, a system is regarded as a set of interacting components. If improving the performance of a component does not cause the system to deteriorate, then such a system is called a coherent system. Based on the characteristics of the system structure, the reliability of a system can be evaluated with different analytical methodologies.

Reliability practices can be traced back to the late 1940's. With the introduction of solid-state electronics, equipment became more and more complex, and the electronic industries recognized that the concern of their customers with the reliable performances of the products was not adequately addressed by the quality control practices of their day. They began using the term *reliability* to refer to the capability of a product not only meeting the customer's needs but also functioning according to the customer's needs when the customer need occured. During the late 1950's, a comprehensive study on the reliability of electronic components was established by AGREE (Advisory Group on Reliability of Electronic Equipment).

The concept of *performing its intended function* in the definition of reliability dictates that a component or a system must perform adequately according to some specifications. The classical reliability theory assumes that a system and its components can only be in one of two states: either working or failed. Hence the meanings of "function" is unique. There is a one-to-one correspondence between the reliability of a component or system and its intended functions. Such a system (component) is called a binary system (component). In the past half century, the binary system reliability has been well developed and it services a vast range of industries including electronics.

However, a system or a component could experience more than two levels of performance varying from perfectly functioning to completely failed in many real-world situations. For example, a vehicle dealer usually uses more than two states to describe the running condition of a vehicle: perfect, good, average, rough or failed such that a vehicle can be considered to be a multi-state system. The idea of multi-state system was first introduced by Hirsch and Meisner (1968). Although both the binary system reliability and the multi-state system reliability are built on the coherent structure theory, the contents of the latter are much more extensive and complex than the former. The first multi-state coherent system model was not presented until 1978 (Barlow and Wu, 1978). Lately, many authors made their contributions to enrich the multi-state reliability theory.

In a multi-state system, it is usually assumed that both the system and the components may be in one of $M+1$ possible states: $0, 1, 2, \cdots, M$, where M is the perfect functioning state while 0 is the complete failure state. If $M = 1$, it reduces to a binary system. Regarding to system level j ($j = 1, 2, \cdots, M$), if the state of a system is equal to j or above, then the system may be considered to be functioning; Otherwise, the system is considered to be failed. The function of a multi-state system has dynamic meanings with respect to different system performance levels. The reliability of a multi-state system or a multi-state component is no longer one single value. We use "performance distribution" instead of "reliability" to denote the probability of a multi-state system or a multi-state component performing its intended function at different performance levels. In the past two decades, over 150 papers relevant to multi-state systems were published. As a more flexible tool for analysis of practical systems, the multi-state system reliability theory is attracting more and more attention.

This thesis strives to contribute to the multi-state system reliability theory, especially in multi-state system modeling and its applications, the relationship between binary system and multi-state system, and multi-state system reliability evaluation. This research work aims to achieve three objectives: generalized multi-state k-out-of-n system; multi-state consecutive k-out-of-n system; and dominant system and its properties. The thesis is presented in seven chapters.

In Chapter 2, we review the fundamental theory of both binary system reliability and multi-state system reliability. The key concepts in the binary context, including the definition of a coherent system, minimal path sets and minimal cut sets, duals, modules and the algorithms for system reliability evaluation, are discussed. These concepts are then extended to the multi-state case. The inherent differences between binary systems and multi-state systems are discussed. Binary k-out-of-n systems and consecutive k-out-of-n systems are also reviewed.

Chapter 3 provides an overview of advanced topics on multi-state reliability theory. The topics discussed include:

1. multi-state relevancy conditions. Over half of the papers in multi-state system reliability are on defining multi-state systems either using component relevancy

conditions or binary structure functions. Various multi-state component relevancy conditions are reviewed and their classifications are presented.

2. binary decomposition of multi-state systems. Two binary decomposition methods for multi-state systems are discussed.

3. multi-state system models and their properties. The techniques to model multi-state systems are presented.

4. algorithms for multi-state system performance evaluation or its bound computation. The advantages and shortcomings of the algorithms are compared.

Other related issues about multi-state reliability are also summarized in this chapter.

In Chapter 4, we propose a definition of the generalized multi-state k-out-of-n system. Other researchers define a multi-state k-out-of-n system as a system with a single k-out-of-n structure at all the system levels. Thus the system can be treated as a binary k-out-of-n system. In our definition, we allow the system to have different k-out-of-n structures at different system levels. It implies that k could take different values at different system levels. Two special cases, the increasing system and the decreasing system, are considered. Algorithms for the system performance evaluation are developed. Examples are provided to illustrate the applications of the proposed definition.

Chapter 5 provides a definition of the multi-state consecutive k-out-of-n:F system. Again two special cases, the decreasing system and the increasing system, are considered. The dual relationship between multi-state consecutive k-out-of-n:F system and consecutive k-out-of-n:G system is explored. We develop an algorithm for the performance evaluation of the decreasing systems and a bound computation method for the increasing systems. Examples are provided to illustrate the applications of the proposed definitions and methods..

In Chapter 6, we investigate how a multi-state system can be dichotomous at all the performance levels and how a multi-state system can be treated as a binary system in its performance evaluation. The definition of dominant system is proposed. Under the proposed definition, multi-state systems are divided into two groups: dominant systems and non-dominant systems. A dominant system with binary image can be treated as a binary system such that existing binary algorithms for system reliability evaluation can be applied on it directly. The properties of minimal path sets and minimal cut sets of dominant systems are discussed. The dual relationship between a dominant system and a non-dominant system is explored. Examples are given to illustrate the potentials of the concept of dominant system in performance evaluation of multi-state systems.

Chapter 7 provides a summary of this research and discusses future research directions in the area of multi-state system reliability.

Chapter 2

Fundamentals of Binary System and Multi-state System Reliability Theory

2.1 Introduction

The reliable performance of a system is of utmost importance in many industrial, commercial and military situations. The fundamental problem in reliability theory is determining the relationship between the reliability of a system and the reliabilities of its components. The theory of binary coherent structures has served as a unifying foundation for the mathematical theory of reliability (Barlow and Proschan, 1975). Various modeling techniques for binary systems have been developed and many models have been applied in practice. In this chapter, we present a literature review related to the fundamentals of the binary system reliability theory and the multi-state system reliability theory. We will show how relevant concepts in the binary case are extended to the multi-state case. Other advanced topics of the multi-state system reliability theory will be discussed in Chapter 3. Section 2.2 reviews the definition of binary coherent systems and various key concepts in the binary context. Section 2.3 briefly reviews binary k-out-of-n systems and consecutive k-out-of-n systems. A practical example is used to illustrate the applications of binary k-out-of-n structures in oil-sands industry. Section 2.4 focuses on important concepts of multi-state systems and their relations to those in the binary system reliability theory. The inherent differences between binary systems and multi-state systems are also pointed out. Section 2.5 provides concluding remarks.

2.2 Binary system reliability theory

Notation:

n	number of components.
C	collection of components, $C = \{1, 2, \cdots, n\}$.
x_i	state of component i, $x_i \in \{0, 1\}$.
$\mathbf{x}$	vector of component states, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.
$\mathbf{1}$	$= (1, 1, \cdots, 1)$.
$\mathbf{0}$	$= (0, 0, \cdots, 0)$.
ϕ	system structure function representing the state of the system, $\phi(\mathbf{x}) \in \{0, 1\}$.
ϕ^D	dual of ϕ.
p_i	$\Pr(x_i = 1)$, reliability of component i.
p	$p = p_i$ when components are s-independent and identically distributed.
q_i	$1 - p_i$.
q	$1 - p$.
h	$\Pr(\phi = 1)$, system reliability function.

$\emptyset$	empty set.
$\subset$	proper subset of.
$\in$	basic element of.
$\cup$	intersection.
$\cap$	union.
$\prod_{i=1}^{n} x_i$	$\equiv \min(x_1, \cdots, x_n)$.
$\coprod_{i=1}^{n} x_i$	$\equiv \max(x_1, \cdots, x_n)$.

In a binary system, the system and all of its components have only two states: functioning or failed. Consider a system comprising n components. To indicate the state of the ith component, we assign a binary indicator variable x_i to component i:

$$x_i = \begin{cases} 1, & \text{if component } i \text{ is functioning} \\ 0, & \text{if component } i \text{ is failed} \end{cases}$$

The vector $\mathbf{x} = (x_1, \cdots, x_n)$ is called a component state vector. The state of the system is determined completely by the states of the components so that the structure function of the system ϕ may be written as:

$$\phi = \phi(\mathbf{x}).$$

Definition 2.1: A system comprising n components, often denoted by (C, ϕ), is called a coherent system if its structure function ϕ: $\{0, 1\}^n \mapsto \{0, 1\}$ satisfies:

1. $\phi(\mathbf{x})$ is non-decreasing in each argument;

2. For any component i, there exists a vector $(\cdot_i, \mathbf{x})$ such that $\phi(1_i, \mathbf{x}) = 1$ and $\phi(0_i, \mathbf{x}) = 0$;

3. $\phi(\mathbf{j}) = j$ for $j = 0, 1$.

where $(\cdot_i, \mathbf{x}) = (x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n)$. In Definition 2.1, Condition 1 implies that improving the performance of a component never causes the system to deteriorate, i.e., its structure function ϕ is a monotonically non-decreasing function of each argument. Condition 2 implies that each component is relevant. In mathematical form, Condition 2 can be written as: for any component i, there exists a vector $(\cdot_i, \mathbf{x})$ such that $\phi(1_i, \mathbf{x}) > \phi(0_i, \mathbf{x})$. Condition 3 indicates that if all the components work, then the system works, and if all the components are failed, then the system is failed. This condition is automatically satisfied with coherent systems.

Series and parallel structures are two of the most commonly seen coherent systems. Figure 2.1 and 2.2 show a series configuration and a parallel configuration respectively. A series system works if and only if all the components work and its structure function is given by:

$$\phi(\mathbf{x}) = \prod_{i=1}^{n} x_i = \min(x_1, \cdots, x_n)$$

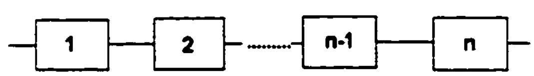

Figure 2.1: A series configuration

A parallel system works if and only if at least one redundant component works and its structure function is given by:

$$\phi(\mathbf{x}) = \coprod_{i=1}^{n} x_i = \max(x_1, \cdots, x_n)$$

A component state vector $\mathbf{x}$ is a path vector if $\phi(\mathbf{x}) = 1$. A path vector $\mathbf{x}$ is a minimal path vector if $\mathbf{y} < \mathbf{x} \Rightarrow \phi(\mathbf{y}) = 0$ for all $\mathbf{y}$. A component state vector $\mathbf{x}$ is a cut vector if $\phi(\mathbf{x}) = 0$. A cut vector $\mathbf{x}$ is a minimal cut vector if

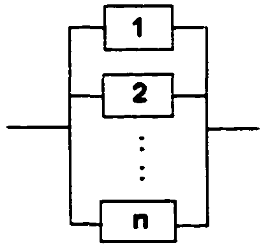

Figure 2.2: A parallel configuration

$\mathbf{y} > \mathbf{x} \Rightarrow \phi(\mathbf{y}) = 1$ for all $\mathbf{y}$, where $\mathbf{y} > \mathbf{x}$ means that $y_i \geq x_i$ for all i and at least one $y_j > x_j$, $i, j = 1, 2, \cdots, n$. Physically, a minimal path set is a minimal set of components whose functioning ensures the functioning of the system. A minimal cut set is a minimal set of components whose failure causes the system to fail. A series system of n components has one minimal path and n minimal cuts. A parallel system of n components has n minimal paths and one minimal cut. Minimal path sets and minimal cut sets play important roles in the analysis of system reliability. The reliability of a system is equal to the probability that at least one of the minimal path sets works. The unreliability of the system is equal to the probability that at least one minimal cut set is failed.

A coherent system can be represented in terms of minimal path sets and minimal cut sets. Consider a coherent system ϕ with s minimal path sets $\{P_1, P_2, \cdots, P_s\}$ and t minimal cut sets $\{K_1, K_2, \cdots, K_t\}$. The system is functioning if and only if all the components in at least one minimal path set are functioning, or alternatively, if and only if at least one component in each minimal cut set is functioning. Thus we have the following two representations of the structure function ϕ in terms of minimal path sets or minimal cut sets:

$$\phi(\mathbf{x}) = \begin{cases} \max_{1\le j\le s} \min_{i\in P_j} x_i \\ \min_{1\le j\le t} \max_{i\in K_j} x_i. \end{cases} \quad (2.1)$$

The following inequality holds for any system structure ϕ (Barlow and Proschan, 1975):

$$\prod_{i=1}^{n} x_i \le \phi(\mathbf{x}) \le \coprod_{i=1}^{n} x_i, \quad (2.2)$$

Equation (2.2) means that the performance of a coherent system is bounded below by the performance of a series structure and above by the performance of a parallel structure.

Given a structure function ϕ, we define its dual ϕ^D by:

$$\phi^D(\mathbf{x}) = 1 - \phi(\mathbf{x}) \quad (2.3)$$

The concept of dual structure has a physical interpretation. For example, the dual of a series structure is a parallel structure and vice versa. If $\mathbf{x}$ is a minimal path for ϕ, then $\mathbf{1} - \mathbf{x}$ is a minimal cut for ϕ^D and vice versa.

In reliability analysis of large and complicated systems, one often computes the reliability of each of its disjoint sub-systems first, and then compute the overall system reliability from these sub-system reliabilities. Modular decomposition provides such a method for breaking a system into several sub-systems. A coherent sub-system (A, χ) is a module of the coherent system (C, ϕ) if and only if

$$\phi(\mathbf{x}) = \psi[\chi(\mathbf{x}^A), \mathbf{x}^{A^C}] \quad (2.4)$$

where ψ is a coherent structure function and $A \subset C$, $\mathbf{x}^A$ denotes the vector with elements x_i for $i \in A$ and A^C is a subset of C and complementary to A. Intuitively, a module is a coherent sub-system that acts as if it were just a component. $\chi(\mathbf{x}^A)$ is a binary variable indicating the state of $\mathbf{x}^A$. Knowing whether χ is 1 or 0 is as informative as knowing the value of x_i for each i in A as far as system reliability is concerned. A modular decomposition of a coherent system (C, ϕ) is a set of disjoint modules $\{(A_1, \chi_1), \cdots, (A_r, \chi_r)\}$ together with an organizing structure ψ, i.e.,

1. $C = \cup_{i=1}^{r} A_i$, where $A_i \cap A_j = \emptyset$ for $i \neq j$;

2. $\phi(\mathbf{x}) = \psi[\chi_1(\mathbf{x}^{A_1}), \cdots, \chi_r(\mathbf{x}^{A_r})]$.

Various approaches and algorithms have been reported for reliability evaluation or bound computation of binary systems. The system reliability function h is a deterministic function of component reliabilities. Assume that components are statistically independent and let $\mathbf{p} = (p_1, p_2, \cdots, p_n)$ be the reliability vector of components. We have:

$$h(\mathbf{p}) = Pr(\phi(\mathbf{x}) = 1) = E(\phi(\mathbf{x})). \tag{2.5}$$

The pivotal decomposition of the system reliability function is as follows:

$$h(\mathbf{p}) = p_i h(1_i, \mathbf{p}) + (1 - p_i) h(0_i, \mathbf{p}). \tag{2.6}$$

where $h(1_i, \mathbf{p}) = \Pr(\phi(1_i, \mathbf{x}) = 1)$ and $(1_i, \mathbf{x}) = (x_1, \cdots, x_{i-1}, 1, x_{i+1}, \cdots, x_n)$, and $h(0_i, \mathbf{p}) = \Pr(\phi(0_i, \mathbf{x}) = 1)$ and $(0_i, \mathbf{x}) = (x_1, \cdots, x_{i-1}, 0, x_{i+1}, \cdots, x_n)$. Equation (2.6) permits us to express a reliability function of order n in terms of reliability functions of order $n - 1$.

The minimal path and minimal cut representations provide a means for computing the exact system reliability.

$$h(\mathbf{p}) = E \coprod_{j=1}^{s} \prod_{i \in P_j} x_i \tag{2.7}$$

$$h(\mathbf{p}) = E \prod_{j=1}^{t} \coprod_{i \in K_j} x_i \tag{2.8}$$

Usually, finding the exact system reliability is a formidable task for complex systems. The bounds on system reliability sometimes are used to approximate the reliability of these systems. Assume that all components are independent in a system, the following are the commonly-used approaches for system reliability evaluation or bound computation (Barlow and Proschan, 1975):

1. The Inclusion-Exclusion method (IE)

This method provides successive upper and lower bounds on system reliability which converge to the exact system reliability. Assume that the system has s minimal path sets. Let E_r be the event that all components in minimal path set P_r function, where $r = 1, 2, \cdots, s$.

$$\Pr(E_r) = \prod_{i \in P_r} p_i \tag{2.9}$$

System success corresponds to the event $\cup_{r=1}^{s} E_r$. Then we have:

$$h = \Pr(\phi = 1) = \Pr(\cup_{r=1}^{s} E_r) \tag{2.10}$$

Let $S_k = \sum_{1 \leq i_1 < i_2 < \cdots < i_k \leq s} Pr(E_{i_1} \cap E_{i_2} \cap \cdots \cap E_{i_k})$. By the inclusion-exclusion principle:

$$h = \sum_{k=1}^{s} (-1)^{k-1} S_k. \tag{2.11}$$

2. The Min-Max method

This method results from the minimal path and minimal cut representations of the system structure function. Assume that a system has s minimal path sets $\{P_1, \cdots, P_s\}$ and t minimal cut sets $\{K_1, \cdots, K_t\}$. The upper bounds of the system reliability depend on minimal cuts and lower bounds depend on minimal paths.

$$\max_{1\leq r\leq s} \prod_{i\in P_r} p_i \leq Pr(\phi(\mathbf{x}) = 1) \leq \min_{1\leq r\leq t} \coprod_{i\in K_r} p_i$$

3. The Sum of Disjoint Products method (SDP)

The SDP method produces a formula which expresses the union of events as a sum of disjoint products. The formula is entirely additive. Let E_r be the event that all components in minimal path set P_r function, $r = 1, 2, \cdots, s$.

$$Pr(\phi = 1) = Pr(E_1) + Pr(\overline{E_1} \cdot E_2) + \cdots + Pr(\overline{E_1} \cdot \overline{E_2} \cdots \overline{E_{s-1}} \cdot E_s) \qquad (2.12)$$

2.3 Binary k-out-of-n systems and consecutive k-out-of-n systems

As a special class of binary systems, k-out-of-n systems receive considerable research interest because: (1) Such systems are more general than series or parallel systems; and (2) They are frequently encountered in practice (for example, Zuo et al, 1998; Huang and Zuo, 1997). Many researchers have been focusing on k-out-of-n systems and consecutive k-out-of-n systems (for example, Chiang and Niu, 1981; Barlow and Heidtmann, 1984; Rushdi, 1986). In the following, we review k-out-of-n system and consecutive k-out-of-n system separately.

2.3.1 k-out-of-n systems

We again assume that all components are s-independent. A k-out-of-n:G system with n components works if and only if at least k components work. A k-out-of-n:F system with n components is failed if and only if at least k components are failed. A k-out-of-n:G system becomes a series system when $k = n$ and a parallel system when $k = 1$.

A k-out-of-n:G system is equivalent to an $(n-k+1)$-out-of-n:F system. Since the value of n is usually larger than the value of k, redundancy is built in a k-out-of-n system. There are $\binom{n}{k}$ minimal paths in a k-out-of-n:G system and $\binom{n}{n-k+1}$ minimal cuts in an $(n-k+1)$-out-of-n:F system. Figure 2.3 shows a 2-out-of-3:G structure in terms of its minimal paths. If we find the reliability of a k-out-of-n:G system, we have found the reliability of a $(n-k+1)$-out-of-n:F system. The dual of a k-out-of-n:G system is a k-out-of-n:F system with the same components. The dual and equivalence relationships between the k-out-of-n G and F systems are summarized below:

1. A k-out-of-n:G system is equivalent to an $(n-k+1)$-out-of-n:F system.
2. A k-out-of-n:F system is equivalent to an $(n-k+1)$-out-of-n:G system.
3. The dual of a k-out-of-n:G system is a k-out-of-n:F system.
4. The dual of a k-out-of-n:G system is an $(n-k+1)$-out-of-n:G system.
5. The dual of a k-out-of-n:F system is a k-out-of-n:G system.
6. The dual of a k-out-of-n:F system is an $(n-k+1)$-out-of-n:F system.

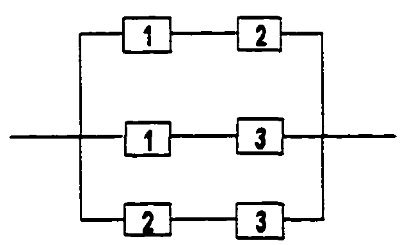

Figure 2.3: A 2-out-of-3:G structure

In the case that all the components are i.i.d., the following equation can be used for reliability evaluation of a k-out-of-n:G system:

$$R_s = \sum_{i=k}^{n} \binom{n}{i} p^i (1-p)^{n-i}, \tag{2.13}$$

where R_s is the system reliability. When the components are non identical, the following recursive equation can be used (Barlow and Heidtmann, 1984; Rushdi, 1986):

$$R(n,k) = p_n R(n-1,k-1) + (1-p_n) R(n-1,k), \tag{2.14}$$

where $R(n,k)$ is the probability that at least k out of n components are good. The complexity of Equation (2.14) is $O(kn)$. The following boundary conditions are needed.

$$R(n,0) = 1 \tag{2.15}$$

$$R(n,k) = 0 \quad \text{for} \quad k > n \tag{2.16}$$

2.3.2 Consecutive k-out-of-n systems

Consider a system which consists of a sequence of n components numbered consecutively from 1 to n. Such a system is a consecutive k-out-of-n:F system if the system fails whenever at least k consecutive components fail. An efficient formula for reliability evaluation of consecutive k-out-of-n:F systems is (Hwang, 1982):

$$Q(n;k) = Q(n-1;k) + (1 - Q(n-k-1;k)) p_{n-k} \prod_{j=n-k+1}^{n} q_j, \tag{2.17}$$

where $Q(n-k-1;k)$ is the failure probability of a consecutive-k-out-of-$(n-k-1)$:F sub-system with components 1 through $n-k-1$. Equation (2.17) can be applied recursively to calculate the reliability of a consecutive-k-out-of-n:F system with the following boundary conditions:

$$Q(j;k) = 0, \quad \text{for} \quad j < k, \tag{2.18}$$

$$p_0 \equiv 1. \tag{2.19}$$

The complexity of Equation (2.17) is O(n). The dual relationship between the consecutive k-out-of-n:F system and the consecutive k-out-of-n:G system was reported by Kuo et al, (1990). A system is a consecutive k-out-of-n:G system if the system works whenever at least k consecutive components work. For the reliability evaluation problem of a consecutive k-out-of-n:G system with component reliability vector $\mathbf{p}$, there is a corresponding dual unreliability evaluation problem of a consecutive k-out-of-n:F system with component reliability vector $\mathbf{1} - \mathbf{p}$ (Zuo, 1993). Let R and Q be the reliability and unreliability, respectively, as a function of system structure, k, n and $\mathbf{p}$. Then we have:

$$Q(F, n, k, \mathbf{1} - \mathbf{p}) = R(G, n, k, \mathbf{p}) \tag{2.20}$$

From Equation (2.20), we can summarize the following procedure for reliability evaluation of the dual system if the available algorithms are for the primal system.

1. Given: k, n, p_1, p_2, ..., p_n for a consecutive k-out-of-n:G system
2. Calculate $q_i = 1 - p_i$ for $i = 1, 2, \ldots, n$
3. Treat q_i as the reliability of component i in a consecutive k-out-of-n:F system and use the algorithms for the F system discussed above to evaluate the reliability of the F system

4. Subtract the calculated reliability of the F system from 1 to obtain the reliability of the original consecutive k-out-of-n:G system

2.3.3 A practical example on binary k-out-of-n structures

In this section, we provide an application of the k-out-of-n:F and the consecutive k-out-of-n:F system reliability models in evaluation of the life distribution of furnaces in oil-sands industry (Zuo, et al, 1998).

A petro-chemical company produces crude oil. It uses several hydrogen plants to provide hydrogen for hydro-treating. The company's oil production is proportional to the amount of hydrogen supplied, i.e., more hydrogen production results in more oil production until the maximum production capacity of the company is reached. Therefore, it is very important that the hydrogen plants operate without interruptions of equipment failures. Of specific interest to the company is the operation of three methane reformer furnaces. These furnaces have hundreds of tubes which are filled with catalyst. Methane and steam are passed through these tubes at high temperature and hydrogen is produced.

The furnaces are considered to be systems while the tubes in the furnaces are components of the corresponding systems. A tube is designed to provide an environment for methane, steam, and catalyst to react at high temperature to produce hydrogen. When we say that a tube is "failed", we mean that the tube is unable to perform its intended function any more. In practical terms, a tube "failure" may mean that the tube is ruptured, or the tube is pinched for whatever reasons. The function of a furnace is to produce hydrogen at certain output, temperature, pressure, and efficiency. If too many tubes are "failed" or pinched, the furnace's proper operation is affected, i.e., the efficiency may be too low. Thus, the system's performance can be defined in terms of the performances of the components.

1	......	15	16	......	31	32	......	46
47	......	61	62	......	77	78	......	92
93	......	107	108	......	123	124	......	138
139	......	153	154	......	169	170	......	184
185	......	199	200	......	215	216	......	230
231	......	245	246	......	261	262	......	276
277	......	291	292	......	307	308	......	322
323	......	337	338	......	353	354	......	368

Figure 2.4: Tube arrangement in Furnace 3

One of the furnaces has 368 tubes evenly divided into 8 rows. Each row has 46 tubes. Figure 2.4 shows the tube arrangement in the furnace. The following furnace failure scenarios are generated with the assistance of engineers in the company.

- **Scenario 1:** The furnace is failed whenever a certain percentage of the tubes in the furnace are all failed. For example, the furnace is failed whenever at least 10% of the tubes (i.e., at least a total of 37 tubes) of the 368 tubes in the furnace are all failed.

- **Scenario 2:** The furnace is failed if at least one row of tubes has at least a certain number of consecutive tubes that are all failed. For example, we may say that the furnace is failed when there are five or more consecutive pinched tubes in a row.

Assuming that the failures of the components are independent of one another, we can describe Scenario 1 using a k-out-of-n:F system model, for example, a 37-out-of-368:F system, and Scenario 2 using a consecutive k-out-of-n:F system, for example, a consecutive 5-out-of-46:F system. Equation (2.14) and (2.17) can be used in evaluation of the reliability of the furnace system.

2.4 Fundamentals of multi-state system reliability theory

Generally speaking, multi-state system reliability models can be divided into discrete models and continuous models based on the properties of the state space. In this section, we will focus on discrete models. We will discuss continuous models in Chapter 3.

Acronym:

MMS: multi-state monotone system.

Notation:

n	number of components.
$M+1$	number of states of system and its components.
S	$\{0, 1, \cdots, M\}$.
x_i	state of component i, $x_i \in \{0, 1, \cdots, M\}$.
$\mathbf{x}$	an n-dimensional vector representing the states of all components, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.
$\mathbf{j}$	$\mathbf{j} = (j, j, \cdots, j)$.
$\mathbf{M}$	$\mathbf{M} = (M, M, \cdots, M)$.
$\phi(\mathbf{x})$	system structure function representing the state of the system, $\phi(\mathbf{x}) \in \{0, 1, \ldots, M\}$.
P_{ij}	$\Pr(x_i \geq j)$.
P_j	$P_j = P_{ij}$ when components are i.i.d.
p_{ij}	$\Pr(x_i = j)$.

p_j	$p_j = p_{ij}$ when components are i.i.d.
R_{sj}	$\Pr(\phi(\mathbf{x}) \geq j)$
r_{sj}	$\Pr(\phi(\mathbf{x}) = j)$

Various approaches have been proposed to define the multi-state structure function ϕ. All papers that we have reviewed consider a multi-state system to be a coherent system. We present the definition of a multi-state coherent system given by Griffth (1980) first.

Definition 2.2: A multi-state system comprising n components is said to be a coherent system if its structure function ϕ satisfies:

1. $\phi(\mathbf{x})$ is non-decreasing in each argument;

2. For any component i, there exists a vector $\mathbf{x}$ such that $\phi(M_i, \mathbf{x}) > \phi(0_i, \mathbf{x})$;

3. $\phi(\mathbf{j}) = j$ for $j = 0, 1, \cdots, M$.

Comparing Definition 2.2 with Definition 2.1, we see that condition 1 in Definition 2.1, the non-decreasing requirement of binary structure functions, is reasonably extended to multi-state structures. Condition 3 in Definition 2.1 is also extended to the multi-state context by requiring $\phi(\mathbf{j}) = j$ for all levels $j = 0, 1, 2, \cdots, M$. This condition is always true in binary coherent systems. However, there are two different assumptions on the number of states of a multi-state system and its components. In Definition 2.2, it is assumed that a system and all of its components have the same number of states, i.e., $M_1 = M_2 = \cdots = M_n = M$, where M_i is the best state of component i while M is the best state of the system. In this case, condition 3 means that if all the components are in a certain state, the system will also be in that state. Most multi-state models reported in the literature require this assumption. In the more general case, a multi-state system and its components may have different numbers of states. For example, a system with three components has $M = 2$ while $M_1 = 3, M_2 = 2$ and $M_3 = 4$. If this kind of model is used, condition 3 in Definition 2.2 is not satisfied. In this thesis, we will focus on models in which the system has the same number of states as all components.

Condition 2, the relevancy condition, can be extended in many ways. In Definition 2.2, it simply says that all the components are relevant to the system but not necessarily to every system level (to be illustrated in Example 2.1). A relevancy condition represents the relationship between the state of a system and the states of its components. The reason that there are many different ways to define the relevancy condition (condition 2) is that there are different degrees of relevancy that a component may have on the system in the multi-state context. A component has $M + 1$ levels and the system has $M + 1$ levels as well. A level of a component may be relevant to some levels of the system but not others. Every level of a component may be relevant to every level of the system (this is the case in the binary case since there are only two levels). Definition 2.2 represents the weakest relevancy condition. More discussions on multi-state relevancy conditions will be presented in Chapter 3.

<u>Example 2.1</u> : Consider a multi-state coherent system wherein both the system and the components may be in one of three possible states, 0, 1, and 2. The following table illustrates the relationship between system state and component states. In this example, $\phi(0,0) < \phi(2,0)$ and $\phi(0,0) < \phi(0,2)$ so ϕ is coherent based on Definition 2.2.

$\phi(\mathbf{x})$:	0	1	2
$\mathbf{x}$:	(0,0)	(0,1)	(2,0)
	(1,0)	(1,1)	(2,1)
		(0,2)	(1,2)
			(2,2)

Table 2.1: System state table for Example 2.1

2.4.1 Minimal path sets and cut sets

The concepts of minimal path sets and minimal cut sets used in the binary systems are extended to the multi-state context in two different ways. El-Neweihi et al (1978) propose the definitions of connection vectors, lower critical connection vectors and upper critical connection vectors as follows:

Definition 2.3 (El-Neweihi et al, 1978)

- A vector $\mathbf{x}$ is said to be a connection vector to level j if $\phi(\mathbf{x}) = j$;
- A vector $\mathbf{x}$ is said to be a lower critical connection vector to level j if $\phi(\mathbf{x}) = j$ and $\phi(\mathbf{y}) < j$ for all $\mathbf{y} < \mathbf{x}$;
- A vector $\mathbf{x}$ is said to be an upper critical connection vector to level j if $\phi(\mathbf{x}) = j$ and $\phi(\mathbf{y}) > j$ for all $\mathbf{y} > \mathbf{x}$.

In this definition, each connection vector to level j, including the lower critical and upper critical connection vectors, results in exactly system level j. The following concepts of minimal paths and minimal cuts in the multi-state context are more analogous to the corresponding concepts in the binary case.

Definition 2.4 (Natvig, 1980):

- A vector $\mathbf{x}$ is called a minimal path vector to level j if $\phi(\mathbf{x}) \geq j$ and $\phi(\mathbf{y}) < j$ for all $\mathbf{y} < \mathbf{x}$;
- A vector $\mathbf{x}$ is called a minimal cut vector to level j if $\phi(\mathbf{x}) < j$ and $\phi(\mathbf{y}) \geq j$ for all $\mathbf{y} \geq \mathbf{x}$.

In this definition, both minimal path vectors and minimal cut vectors are defined in terms of whether they result in a system state "equal to or greater than j" or not. A minimal path vector to level j may or may not be a connection vector to level j as it may result in a system state higher than j. A minimal cut vector to level j may or may not be a connection vector to level $j-1$ as it may result in a system state below $j-1$. Example 2.2 is given to illustrate these concepts.

Example 2.2: Consider a two-component system wherein both the system and the components may be in one of three possible states, 0, 1, and 2. The following table illustrates the relationship between system state and component states.

$\phi(\mathbf{x})$:	0	1	2
	$(0,0)$	$(1,1)$	$(2,0)$
	$(1,0)$		$(0,2)$
$\mathbf{x}$:	$(0,1)$		$(2,1)$
			$(1,2)$
			$(2,2)$

Table 2.2: System state table for Example 2.2

In this system, the lower critical connection vector to level 1 is $(1,1)$ and to level 2 are $(2,0)$ and $(0,2)$. However, $(2,0)$ and $(0,2)$ are also the minimal path sets to level

1 according to Definition 2.4. We will use Definition 2.4 throughout this dissertation.

Similar to Equation (2.1), a multi-state structure function ϕ can be represented using the concepts of minimal path sets or minimal cut sets (Barlow and Wu, 1978):

$$\phi(\mathbf{x}) = \begin{cases} \max_{1\le j\le s} \min_{i\in P_j} x_i \\ \min_{1\le r\le t} \max_{i\in K_r} x_i. \end{cases} \tag{2.21}$$

where P_j represents the jth minimal path set for $j = 1, 2, \cdots, s$ and K_r represents the rth minimal cut set for $r = 1, 2, \cdots, t$. In words, Equation (2.21) says that the system state is equal to the state of the "worst" component in the "best" minimal path or the "best" component in the "worst" minimal cut. This relevancy condition simply extends the domain of the state space from $\{0, 1\}$ in the binary case to $\{0, 1, \cdots, M\}$ in the multi-state case. Hence there is a one-to-one correspondence between the structure of a multi-state structure and a binary structure whenever Equation (2.21) is satisfied.

A multi-state series system is defined as (El-Neweihi et al, 1978):

$$\phi(\mathbf{x}) = \min_{1\le i\le n} x_i,$$

and a multi-state parallel system is defined as:

$$\phi(\mathbf{x}) = \max_{1\le i\le n} x_i.$$

These definitions are natural generalizations from the binary case. The following inequality gives simple bounds on system state similar to Equation (2.2).

$$\min_{1\le i\le n} x_i \le \phi(\mathbf{x}) \le \max_{1\le i\le n} x_i, \tag{2.22}$$

Equation (2.22) indicates that the performance of a multi-state system is bounded below by the performance of a series system and above by the performance of a parallel system.

2.4.2 Duality in multi-state systems

In a multi-state system, let ϕ be the structure function of the system, then its dual structure function ϕ^D is defined as (Xue, 1985):

$$\phi^D(\mathbf{x}) = M - \phi(\mathbf{M} - \mathbf{x}) = M - \phi(M - x_1, \cdots, M - x_n) \tag{2.23}$$

The following summarizes the results related to the duality of multi-state systems, which form the theoretical basis of dual transformation of multi-state systems.

- The dual of a multi-state coherent system is still a multi-state coherent system (El-Neweihi et al, 1978);
- For any multi-state systems ϕ, the dual ϕ^D is in the same class of coherence as ϕ (Griffith, 1980);
- $(\phi^D)^D(\mathbf{x}) = \phi(\mathbf{x})$ (Xue, 1985);
- $\mathbf{x}$ is an upper critical connection vector for level j of ϕ if and only if $\mathbf{M} - \mathbf{x}$ is a lower critical connection vector for level $M - j$ of ϕ^D; $\mathbf{x}$ is a lower critical connection vector for level j of ϕ if and only if $\mathbf{M} - \mathbf{x}$ is an upper critical connection vector for level $M - j$ of ϕ^D (Block and Savits, 1982).

These results are analogous to the binary case. But some dual relationship existing in a binary system may be lost in a multi-state system as illustrated in Example 2.3.

Example 2.3: Let us examine the dual of the system in Example 2.2. In the primal system, the system is a series structure in level 1, in other words, the system is in state 1 or above if both components are in state 1 or above, and the system is a parallel structure in level 2, in other words, the system is in state 2 if at least one component is in state 2. Using Equation (2.23), we obtain the dual of the system as shown in the following table. This duality converts the system neither from a series structure to a parallel structure at level 1 nor from a parallel structure to a series structure at level 2. This contradicts with what happens in the binary case. We will investigate this issue further in Chapter 5 and 6.

$\phi^D(\mathbf{x})$:	0	1	2
	$(0,0)$	$(1,1)$	$(2,1)$
	$(1,0)$		$(1,2)$
$\mathbf{x}$:	$(0,1)$		$(2,2)$
	$(2,0)$		
	$(0,2)$		

Table 2.3: System state table for Example 2.3

2.4.3 Modules of a multi-state system

Since multi-state systems are much more complicated than binary systems, exploring an efficient modular decomposition method is very important for the analysis of multi-state systems. In this section, we review the definition of modules in a multi-state system and summarize some important conclusions. The modular decomposition algorithms will be discussed in Chapter 3.

A module of a multi-state system is defined as (Butler, 1982; Hudson and Kapur, 1983):

Definition 2.5: let (C, ϕ) be a multi-state system and let A be a nonempty proper subset of C. Let $\mathbf{x}_A$ and $\mathbf{x}_{A'}$ be the component state vectors for A and its complement A'. If the coherent structure (A, χ_A) and its structure function ψ satisfy:

$\phi(\mathbf{x}) = \phi(\mathbf{x}_A, \mathbf{x}_{A'}) = \psi[\chi_A(\mathbf{x}_A), \chi_{A'}(\mathbf{x}_{A'})]$, then A is a modular set of (C, ϕ), (A, χ_A) is a module of (C, ϕ) and ψ is called an organizing structure function. A modular decomposition of a multi-state system (C, ϕ) is a set of modules of $\{(A_1, \chi_1), \cdots, (A_k, \chi_k)\}$ together with an organizing structure ψ such that: (1) $A_1, \cdots, A_k$ is a partition of C, i.e, $\cup_{i=1}^{k} A_i = C$ and $A_i \cap A_j = \emptyset$ for $i \neq j$; (2) $\phi(\mathbf{x}) = \psi[\chi_1(\mathbf{x}_{A_1}), \cdots, \chi_k(\mathbf{x}_{A_k})]$.

Some important results about multi-state system modules are listed below (Hudson and Kapur, 1983):

- Unlike binary systems, it is claimed that any subset A of a multi-state system qualifies as a module, i.e., there exists a way to define A and χ_A that always makes (A, χ_A) a module;
- Let $(\{(A_1, \chi_1), \cdots, (A_k, \chi_k)\}, \psi)$ be a modular decomposition of (C, ϕ), then $(\{(A_1^D, \chi_1^D), \cdots, (A_k^D, \chi_k^D)\}, \psi^D)$ is a modular decomposition of (C^D, ϕ^D);
- An index I, named "modular efficiency", is defined to measure whether reliability computation using the modular decomposition is more efficient than without the decomposition.

$$I = \frac{n(\psi) + \sum_{j=1}^{k} n(\chi_j)}{n(\phi)} \tag{2.24}$$

  where $n(*)$ is the number of components in the corresponding system or subsystems and k is the total number of modules of the system. The decompositions with an index less than 1 are desirable. This formula is easy to apply and deserves further investigation.

In the dichotomous opinion, the states of a multi-state system can be decomposed into two groups: $\phi \geq j$ and $\phi < j$ at any level j, $j = 1, 2, \ldots, M$ so that the system performance with respect to level j can be defined as $R_{sj} = \Pr(\phi \geq j)$. The component performance with respect to level j can be defined as $P_{ij} = \Pr(x_i \geq j)$ for $i = 1, \cdots, n$. The algorithms for system reliability evaluation in the binary context can then be applied in the multi-state context. For example, assume that the performance function $h(\mathbf{p})$ of a multi-state system is given, then it can be expressed using the pivotal decomposition method as (El-neweihi et al, 1978; Hudson and Kapur, 1983):

$$h(\mathbf{p_1}, \mathbf{p_2}, \cdots, \mathbf{p_n}) = \sum_{j=0}^{M} p_{ij} h(j_i; \mathbf{p_1}, \mathbf{p_2}, \cdots, \mathbf{p_n}) \tag{2.25}$$

where $h(j_i; \mathbf{p_1}, \mathbf{p_2}, \cdots, \mathbf{p_n}) = E\phi(X_1, \cdots, X_{i-1}, j, X_{i+1}, \cdots, X_n)$.

IE and SDP techniques are also applicable to system performance evaluation or bound computation. We will discuss them in Chapter 3.

In addition, many stochastic performance properties, such as mean time to failure, mean time between failures, the generalized IFRA(Increasing Failure Rate Average) closure theorem and so on (Barlow and Proschan, 1975), can also be extended from the binary case to the multi-state case (El-Neweihi et al, 1978). But not much work has been done on investigating the conditions for the use of these concepts in the multi-state case.

2.5 Concluding remarks

This chapter reviewed various key concepts in the binary system reliability theory and in the multi-state system reliability theory. Almost all concepts in the binary case can be extended to the multi-state case and the latter inherits many properties from the former. However, the progression from the binary theory to the multi-state theory is not simply extending the domain of the system state space from $\{0, 1\}$ to $\{0, 1, \cdots, M\}$. We have seen that there exist some essential distinctions between them. For example, the minimal path sets of a multi-state system could have more complicated meanings as shown in Example 2.2. Such distinctions result from the complicated relevancy conditions of multi-state systems. In Chapter 3, we will discuss various multi-state relevancy conditions and then proceed to advanced topics of multi-state system reliability theory, which include binary decomposition of multi-state system, generalization of multi-state models, the algorithms and bounding techniques for multi-state system performance distribution evaluation and so on, and then propose the research motivation of this thesis.

Chapter 3

Advanced Topics of Multi-state System Reliability Theory

3.1 Introduction

The concept of multi-state coherent structures was reported as early as 1968 by Hirsch and Meisner (1968). However, the generalizations to multi-state reliability theory had not been fruitful until the first multi-state coherent structure was suggested by Barlow and Wu (1978). The reasons that it took such a long time for the multi-state reliability theory to develop are (1) the binary coherent reliability theory was able to serve many practical requirements well; and (2) multi-state coherent systems have much more complicated properties than binary systems.

In Chapter 2, we have reviewed many fundamental concepts of the multi-state system reliability theory. The contents of the multi-state reliability theory is more profound and extensive than the binary reliability theory. In this chapter, we present a literature review on advanced topics of multi-state reliability. Section 3.2 discusses multi-state relevancy conditions and how to define a multi-state system using relevancy conditions. Section 3.3 briefly reviews two binary decomposition methods for multi-state systems. Some multi-state systems can be expressed in terms of a binary structure function and binary variables. Section 3.4 is on various multi-state modeling techniques and the properties of multi-state reliability models. Section 3.5 discusses the algorithms and bounding techniques for system performance evaluation. In Section 3.6, we propose the research motivation of this thesis.

3.2 Relevance conditions

In a binary system, the component relevancy condition is unique as defined in Definition 2.1. In a multi-state system, the component relevancy conditions have been defined in many ways. For example, consider a coherent system with two components, under the relevancy condition as defined in Definition 2.2 (Barlow and Wu, 1978), the system has only two possible structures: either series or parallel just like in the binary case. But under the relevancy condition imposed by El-Neweihi et al (1978), the system could have 12 possible structures. As the central notion of the multi-state reliability theory, component relevancy conditions contain many interesting issues. Over half of the papers on multi-state reliability theory are about defining relevancy conditions. It has led to the introduction of several classes of multi-state coherent systems.

3.2.1 Fundamental relevance conditions

The first multi-state system was proposed using the concepts of minimal path sets or minimal cut sets (Barlow and Wu, 1978). Assume that a system has s minimal path sets and t minimal cut sets. The system structure function ϕ is defined:

• **BW Class** (Barlow and Wu, 1978):

$$\phi(\mathbf{x}) = \begin{cases} \max_{1\le j\le s} \min_{i\in P_j} x_i \\ \min_{1\le r\le t} \max_{i\in K_r} x_i. \end{cases} \quad (3.1)$$

where P_j represents the jth minimal path set, $j = 1, 2, \cdots, s$ and K_r represents the rth minimal cut set, $r = 1, 2, \cdots, t$.

Based on the BW relevancy condition, the system state is equal to the state of the "worst" component in the "best" minimal path or the "best" component in the "worst" minimal cut. BW class simply extends the domain of the state space from $\{0, 1\}$ in the binary case to $\{0, 1, \cdots, M\}$ in the multi-state case. Hence there is a one-to-one correspondence between the structure of a multi-state system and a binary structure under this definition. For example, for a two-component system, BW class yields only two possible systems: the parallel system when $t = 1$ and the series system when $s = 1$. Although this definition is very simple and specific, it lays the foundation for a theory of multi-state systems.

Later, other classes of multi-state relevancy conditions were proposed following the BW class to reflect the complex inherent properties of multi-state systems. Griffth (1980) divided the multi-state relevancy conditions into three types: strong coherent, coherent and weak coherent.

- **EPS Class: strong coherent** (El-Neweihi, Proschan and Sethuraman, 1978):

A system of n components is said to be a coherent system if its structure function ϕ satisfies: (i) ϕ is non-decreasing; (ii) for every level j of component i, there exists a vector $(\bullet_i, \mathbf{x})$ such that $\phi(j_i, \mathbf{x}) = j$ while $\phi(l_i, \mathbf{x}) \neq j$ for $l \neq j, j = 0, 1, \cdots, M$; (iii) $\phi(\mathbf{j}) = j$.

The relevancy condition (ii) means that every level of each component is relevant to the same level of the system. This condition is very strong such that it is called strong coherency by Griffth (1980). EPS class includes BW class as a sub-class. This means that the BW class also specifies a strong coherent system.

Example 3.1: A multi-state structure under EPS class

Consider a two-component system wherein both the system and the components may be in one of three possible states, 0, 1, and 2. The following table specifies the relationship between system state and component states.

$\phi(\mathbf{x})$:	0	1	2
	$(0,0)$	$(1,1)$	$(2,1)$
	$(1,0)$		$(1,2)$
$\mathbf{x}$:	$(0,1)$		$(2,2)$
	$(2,0)$		
	$(0,2)$		

Table 3.1: System state table for Example 3.1

Based on the definitions of multi-state series system and parallel system defined in Chapter 2, at system level 2, the system structure is neither series nor parallel so it does not belong to BW class. But it is one of the structures under EPS class.

- **GRI1 Class: coherent** (Griffith, 1980):

  If there exists a vector $(\bullet_i, \mathbf{x})$ such that $\phi((j-1)_i, \mathbf{x}) < \phi(j_i, \mathbf{x})$ for any component i and state $j \geq 1$, then $\phi(\mathbf{x})$ is said to be coherent.

- **GRI2 Class: weak coherent** (Griffith, 1980):

  If there exists a vector $(\bullet_i, \mathbf{x})$ such that $\phi(l_i, \mathbf{x}) \neq j$ for some $l \neq j$, then $\phi(\mathbf{x})$ is said to be weakly coherent.

The relevancy condition of GRI1 class requires that every level of each component is relevant to the system but not necessarily to the same level of the system. GRI2 class only requires that each component is relevant to the system. GRI2 class is equivalent to that for any component i, there exists a vector $(\bullet_i, \mathbf{x})$ such that $\phi(0_i, \mathbf{x}) < \phi(M_i, \mathbf{x})$. We have discussed GRI2 class in Chapter 2.

Example 3.2: A multi-state structure under GRI1 class

Consider a two-component system wherein both the system and the components may be in one of three possible states, 0, 1, and 2. The following table specifies the relationship between system state and component states.

$\phi(\mathbf{x})$:	0	1	2
	(0,0)	(1,0)	(2,1)
	(0,1)	(1,1)	(2,2)
$\mathbf{x}$:		(2,0)	
		(0,2)	
		(1,2)	

Table 3.2: System state table for Example 3.2

This structure belongs to GRI1 class, but not to EPS class. At system level 1, $\phi(1,0) = \phi(1,1) = \phi(1,2)$, it does not satisfy the strong coherency condition.

Example 3.3: A multi-state structure under GRI2 class

Consider a two-component system wherein both the system and the components may be in one of three possible states, 0, 1, and 2. The following table illustrates the relationship between system state and component states.

$\phi(\mathbf{x})$:	0	1	2
	(0,0)	(0,1)	(2,0)
$\mathbf{x}$:	(1,0)	(1,1)	(2,1)
		(0,2)	(2,2)
		(1,2)	

Table 3.3: System state table for Example 3.3

This structure belongs to GRI2 class, but not to GRI1 class since level 1 of component 1 is irrelevant to any system state.

There are many other classes of multi-state relevancy conditions. Their definitions and properties are summarized as below:

- **NAT1 Class** (Natvig, 1980):

  A system of n components is said to be coherent if there exists a vector $(\bullet_i, \mathbf{x})$ such that $\phi(j_i, \mathbf{x}) \geq j$ and $\phi((j-1)_i, \mathbf{x}) \leq j-1$ for all components $i = 1, \cdots, n$ and all states $j = 1, \cdots, M$.

- **EBR Class** (Ebrahimi, 1983):

  For any components i, there exists a vector $(\bullet_i, \mathbf{x})$ such that $\phi(j_i, \mathbf{x}) \geq j$ and $\phi((j-1)_i, \mathbf{x}) \leq j-1$ for at least one j.

Considering that the states of both a system and its components can be divided into two groups: $\geq j$ or $< j$, NAT1 class can be rephrased as : there exists a vector $(\bullet_i, \mathbf{x})$ such that $\phi(l_i, \mathbf{x}) \geq j$ for $l \geq j$ and $\phi(k_i, \mathbf{x}) < j$ for $k < j$. NAT1 class implies that every level j of each component is relevant to the same level or above of the system while GRI1 class just claims that $\phi((j-1)_i, \mathbf{x}) < \phi(j_i, \mathbf{x})$. Clearly, NAT1 class is stronger than GRI1 class. EBR class is equivalent to NAT1 class if it holds for every level j so it has a weaker requirement than NAT1 class.

- **NAT2 Class** (Natvig, 1980):

If a multi-state structure function ϕ has the representation $\phi(\mathbf{x}) = \sum_{j=1}^{M} \phi_j(\mathbf{I_j}(\mathbf{x}))$, where $\phi_1 \geq \phi_2 \cdots \geq \phi_M$ are M binary coherent structures and $\mathbf{I_j}(\mathbf{x}) = (I_j(x_1), \cdots, I_j(x_n))$ is a vector of indicator functions defined as:

$$I_j(x_i) = \begin{cases} 1, & \text{if } x_i \geq j \\ 0, & \text{otherwise} \end{cases}$$

then $\phi(\mathbf{x}) \geq j \iff \phi_j(\mathbf{I_j}(\mathbf{x})) = 1$.

NAT2 class represents a multi-state structure in terms of a uniquely determined binary structure function, but it is far more general than BW class. It allows the system to have different structures at different levels while BW class assumes the system structure is the same at all levels. If all ϕ_j's have the same structure, NAT2 class reduces to BW class. But NAT2 is not the same as EPS class to be illustrated in Example 4.

<u>Example 3.4: A multi-state structure under NAT2 class</u>

Consider a two-component system wherein both the system and the components may be in one of three possible states, 0, 1, and 2. The following table specifies the relationship between system state and component states.

The system belongs to NAT2 class. At system level 1, it is a parallel structure while at level 2, it is a series structure. But it does not belong to the EPS class.

The relevancy conditions discussed above are accepted as the fundamental ones. We use $A \subset B$ to denote A *is properly contained in* B. According to the order from the strongest to the weakest, we develop a detailed classification scheme for the relevancy conditions as follows.

$\phi(\mathbf{x})$:	0	1	2
	(0,0)	(0,1)	(2,2)
		(1,0)	
		(1,1)	
$\mathbf{x}$:		(2,0)	
		(0,2)	
		(2,1)	
		(1,2)	

Table 3.4: System state table for Example 3.4

BW⊂ EPS(or NAT2)⊂ NAT1⊂ GRI1(or EBR)⊂ GRI2⊂ MMS

Andrzejezak (1992) discussed various relevancy conditions. His conclusion is the same as the above scheme except replacing MMS with the BW class since he believes that the BW model is suitable for any MMS. A MMS is irrespective of any relevancy condition so it should be the weakest. Each class of relevancy condition has its own advantages and limitations. The strong relevancy conditions may restrict the analyst's freedom, but the enumeration of the system structure provides a tighter range because they allow fewer possible structures. The weak relevancy conditions are flexible but may be too general to analyze. It is easy to verify that when $M = 1$, all the above relevancy conditions are reduced to binary coherent system. These relevancy conditions are often cited in building multi-state system models. For example, Hudson (1983) used a relevance condition which is the same as GRI2 class. Xue (1985) used one similar to GRI1 class. This is why we call them the fundamental relevancy conditions.

3.2.2 Generalizations of relevance conditions

Since there are so many ways to define relevancy conditions, it seems necessary to generalize the relevancy conditions using some criterion. Such attempts have been made by: (1) Producing a sequence of relevancy classes which contains those existing relevancy conditions in a unified form (Ohi and Nishida, 1983). (2) Producing a unified formula for expressing various classes of relevancy conditions (Abouammoh and Al-Kadi, 1991 and 1997).

Ohi and Nishida (1983) defined a sequence of relevancy conditions, which contains most of the fundamental relevancy conditions. Consider a multi-state coherent system ϕ with state space $S = \{0, 1, \cdots, M\}$. For each $s \in S$ and each $t \geq s$, there exist $(k_i, \mathbf{x})$ and $(l_i, \mathbf{x})$, k and $l \in S$ and $k \leq l$ such that:

1. OH1 Class: $\phi(k_i, \mathbf{x}) = s$, and $\phi(l_i, \mathbf{x}) = t$ for $i = 1, \cdots, n$;
2. OH2 Class: $\phi(k_i, \mathbf{x}) = s - 1$, and $\phi(l_i, \mathbf{x}) = s$ for $i = 1, \cdots, n$;
3. OH3 Class: $\phi(k_i, \mathbf{x}) \neq s$, and $\phi(l_i, \mathbf{x}) = s$ for $i = 1, \cdots, n$, equivalent to ESP class;
4. OH4 Class: $\phi(k_i, \mathbf{x}) \leq s - 1$, and $\phi(l_i, \mathbf{x}) \geq s$ for $i = 1, \cdots, n$, equivalent to NAT1 class;
5. OH5 Class: $\phi(k_i, \mathbf{x}) \neq \phi(l_i, \mathbf{x})$ for $i = 1, \cdots, n$, equivalent to GRI2 class;
6. OH6 Class: equivalent to MMS;
7. OH7 Class: there exists $(\bullet_i, \mathbf{x})$ such that $\phi(k_i, \mathbf{x}) \neq \phi(l_i, \mathbf{x})$, equivalent to GRI1 class.

In the sequence, only OH1 class and OH2 class are not contained in those fundamental classes discussed in Section 3.2.1. In these two classes, the relations among the states, s, t, k and l, are not well defined so it is difficult to give their physical interpretations. Similarly, the rephrasing of other classes in the sequence does not look any clearer than the original definitions.

By observing the fundamental classes of relevancy conditions and the sequence by Ohi and Nishida (1983), it is possible to express most of them in some unified mathematical form. Abouammoh and Al-Kadi (1991 and 1997) made such effort to recognize the relevancy conditions in terms of the combination of two binary indicator functions. They presented a new definition for multi-state coherent systems of order k, where k refers to the minimum number of levels for which the underlying components of the system are relevant.

1. Let A_j and B_j be two subsets of levels of performance with respect to level j and let ϕ_j be a binary structure function such that

$$\phi_j(\mathbf{x}) = \begin{cases} 1, & \text{if } \phi(\mathbf{x}) \geq j \\ 0, & \text{otherwise} \end{cases}$$

2. A monotone system is a multi-state coherent system of order k, where $k \in S = \{1, 2, \cdots, M\}$, if each component i is relevant to at least k levels of performance of the system ϕ according to the relevance condition $\mid \phi_l(j_i, \mathbf{x}) - \phi_l(k_i, \mathbf{x}) \mid = 1$ for some $k \in A_j$ and for some $l \in B_j$.

The class of multi-state coherent systems of order k is in accordance with most of the classes of the fundamental relevance conditions by specifying k values. EPS class, GRI1 and GRI2 classes, NAT1 class and EBR class can be viewed as the special case by selecting different k and l values. For example, setting $A_j = \{j-1\}$ and $B_j = \{1, 2, \cdots, M\}$, we get: $\phi_l(j_i, \mathbf{x}) - \phi_l((j-1)_i, \mathbf{x}) = 1$ for some $l = 1, \cdots, M$. This is reduced to GRI1 class and then GRI1 class is of order M, i.e., the minimum number of levels for which the underlying components of the system are relevant for GRI1 class is M. This method is helpful for better explaining the relevancy conditions. But BW class and NAT2 class can not be expressed using this definition. Generally speaking, there does not exist a unified method commonly accepted for classifying all existing relevancy conditions.

3.2.3 The relevancy conditions based on L-superadditive function

Another approach concerning a multi-state system is to explore the mathematical properties of its structure function rather than the underlying relevance condition. Structure functions relate the level of performance of a system to the performance levels of its components. L-superadditive (LSP) function and L-subadditive function (LSB) are two particular types of structure functions, which have an intuitive property to describe whether a system is more series-like or more parallel-like (Block et al, 1989).

Definition 3.1: A monotone function ϕ is called LSP if it satisfies $\phi(\mathbf{x} \vee \mathbf{y}) + \phi(\mathbf{x} \wedge \mathbf{y}) \geq \phi(\mathbf{x}) + \phi(\mathbf{y})$ for all $\mathbf{x}, \mathbf{y}$, where $\mathbf{x} \vee \mathbf{y} = (x_1 \vee y_1, x_2 \vee y_2, \cdots, x_n \vee y_n)$ and $x_i \vee y_i = \max\{x_i, y_i\}$; $\mathbf{x} \wedge \mathbf{y} = (x_1 \wedge y_1, x_2 \wedge y_2, \cdots, x_n \wedge y_n)$ and $x_i \wedge y_i = \min\{x_i, y_i\}$.

If the inequality signs are reversed, the function is called LSB. The dual of LSP is LSB. LSP and LSB functions have important reliability interpretations. They can be used to describe whether a system is more series-like or parallel-like. If the structure function of a system is a LSP, then we say it is more series-like. If the structure function of a system is a LSB, then we say it is more parallel-like. In the binary case, the only systems having a LSP function are series systems. The only systems having a LSB are parallel systems. In the multi-state case, there are many structure functions which are LSP and LSB. However, if we restrict our attention to structure functions of BW class, then only the series system is LSP and the parallel system is LSB.

Example 3.5: A multi-state system with LSP

Consider a two-component system wherein both the system and the components may be in one of three possible states, 0, 1, and 2. The following table specifies the relationship between system state and component states.

$\phi(\mathbf{x})$:	0	1	2
	$(0,0)$	$(1,1)$	$(2,1)$
$\mathbf{x}$:	$(0,1)$	$(1,2)$	$(2,2)$
	$(0,2)$	$(2,0)$	
	$(1,0)$		

Table 3.5: System state table for Example 3.5

It is easy to check that ϕ is MMS and the LSP property holds, but it is not a series system. We say it is more series-like because if component 1 is in state 0, then the system will be in state 0 too. From this example, it is of interest to raise such a question: under what relevancy conditions, is a system with LSP exactly a series one and a system with LSB exactly a parallel one? The following two classes of relevancy conditions provide answers to this question (Meng, 1993).

- **MFC1 Class** (Meng, 1993): For each component i, level $k > 0$, there exists a vector $(\bullet_i, \mathbf{x})$ such that $\phi(0_i, \mathbf{x}) < k \leq \phi(k_i, \mathbf{x})$.

- **MFC2 Class** (Meng, 1993): For each component i, level $k > 0$, there exists a vector $(\bullet_i, \mathbf{x})$ such that $\phi((k-1)_i, \mathbf{x}) < k \leq \phi(M_i, \mathbf{x})$.

In MFC1 class, at system level k, functioning ($\geq k$) of component i can ensure the system to be functioning and removing this component can cause system failure ($< k$). MFC2 class means that failure of the component will cause system failure and improving the component truly improves the system. The properties enjoyed by MFC1 class or MFC2 class have their counterparts in each another. If $\phi \in$ MFC1 class, then $\phi^D \in$ MFC2 class. By imposing these two relevancy conditions, series and parallel structures are characterized within LSP and LSB functions, i.e, if ϕ is a LSP function, then $\phi \in$ MFC2 class implies ϕ is a series structure; if ϕ is a LSB function, then $\phi \in$ MFC1 class implies ϕ is a parallel structure. The relation between these two classes of the relevancy conditions and others is: NAT1$\subset$ MFC1$\cap MFC2 \subset$ MFC1 $\cup$ MFC2 $\subset$ GRI2.

3.2.4 Characterization results on BW class

Of all the relevancy conditions, BW class is the most intimately related to that of a binary system and it possesses many properties inherited from the binary system. Although many classes of relevancy conditions include BW class as a special case, only BW class provides an intuitive expression for the system structure function. Characterizing BW class within other classes is a natural task that has been addressed by some authors. Many axiomatic characterization results relating BW class to other classes have been developed. Two important issues are: (1) The equivalent condition of BW class and (2) The elimination of those strict assumptions imposed on BW class, for example, BW class requires the system structures to be the same at binary structures.

The first group of equivalent conditions for the characterization of BW model is presented as below (Borges and Rodrigues, 1983):

1. For every $\mathbf{x} \in S^n$ with $\phi(\mathbf{x}) \geq k \geq 1$, there exists $\mathbf{y} \in \{0, k\}^n$ such that $\mathbf{y} \leq \mathbf{x}$ and $\phi(\mathbf{y}) \geq k$;

2. For every $\mathbf{x} \in \{0, M\}^n$, $\phi(\mathbf{x}) =$ either 0 or M;

3. For every i, there exists $\mathbf{x} \in S^n$ such that $\phi(0_i, \mathbf{x}) < \phi(M_i, \mathbf{x})$.

A monotone system belongs to BW class if its structure function ϕ satisfies these three conditions. Condition 1 is equivalent to: a system structure function ϕ can be expressed by a set of minimal paths. Condition 2 means that $\phi(\mathbf{j}) = j$ just for $j = 0$ and M, but not necessarily for $j = 1, 2, \cdots, M-1$. Condition 3 has been introduced to the weak relevancy in GRI2 class. ϕ is under NAT2 class if only conditions 1 and 3 hold.

When the system state space is continuous, it leads to a continuous multi-state system. An axiomatic characterization for BW class given a continuous state space is investigated by Mak (1989), where the system structure function ϕ is assumed to be a continuous function in the interval $[0, 1]$. A continuous structure function $\phi : [0, 1]^n \rightarrow [0, 1]$ has the BW form if and only if it satisfies the following conditions:

1. ϕ is non-decreasing in each of its arguments;

2. For each i, there exists $x_j \in [0, 1]$ for all $j \neq i$ such that with all x_j's fixed the function : $x_i \in [0, 1] \rightarrow \phi(x_1, \cdots, x_i, \cdots, x_n)$ is non-constant;

3. ϕ is continuous;

4. for each $\mathbf{x} \in [0, 1]^n$, there exists a i such that $\phi(\mathbf{x}) = x_i$.

Conditions 1 and 2 say that the system is coherent. The essential feature of a BW continuous structure function is the property stated in Condition 4. It says that the system must perform at the same level as that of some component.

Realizing that BW model is the class of least generality but with most dichotomous characterization among relevancy conditions, it is natural to attempt to develop a new class of multi-state systems which is similar to BW class except that it possesses no relevancy restrictions (Ansell and Bendell, 1987). A coherent system is defined in the wide sense if and only if its structure function ϕ satisfies $\phi(\mathbf{0}) = 0$ and $\phi(\mathbf{M}) = M$; or in the narrow sense if and only if $\phi(\mathbf{j}) = j$ for all $j = 0, 1, \cdots, M$. Wide-sense coherency is identical to Condition 2 proposed by Borges and Rodrigues (1983). Narrow-sense coherency is included in most of the other classes, for examples, EPS class and GRI1 class.

Definition 3.2: A wide-sense coherent system ϕ is said to have a well-defined binary image if and only if $I_j(\mathbf{x}) = I_j(\mathbf{y})$ implies that $\phi^j(\phi(\mathbf{x})) = \phi^j(\phi(\mathbf{y}))$ for any j, where $I_j(\mathbf{x})$ is a binary indicator vector, i.e., $I_j(\mathbf{x}) = (I_j(x_1), I_j(x_2), \cdots, I_j(x_n))$ and $I_j(x_i) = 1$ if $x_i \geq j$ and $= 0$ otherwise. ϕ^j is a binary indicator function (Ansell and Bendell, 1987).

A coherent system with a well-defined binary image is such that no ambiguity is obtained if the multi-state system modeled as a two-state one. In such a system, if the number of minimal paths at every level is the same and each one is of the form $(j, \cdots, j, 0, \cdots, 0)$, it is called a coherent system with a constant binary image (Ansell and Bendell, 1987). Coherent systems with constant binary images lead to BW class by an additional relevancy requirement. All BW coherent systems are contained in the class of narrow sense coherent systems possessing a well-defined binary image. We have noticed that this definition contains strong restrictions. For example, a system could be divided into a two-state one, but it is not necessarily one that is called a coherent system with a well-defined binary image. A system that is defined as a narrow-sense coherent system with a constant binary image actually may not be coherent. We will discuss more about this definition in Chapter 6.

Example 3.6: A narrow-sense coherent system with a constant binary image

Consider a two-component system wherein both the system and the components may be in one of three possible states, 0, 1, and 2. The following table specifies the relationship between system state and component states.

$\phi(\mathbf{x})$:	0	1	2
	(0, 0)	(1, 0)	(2, 0)
$\mathbf{x}$:	(0, 1)	(1, 1)	(2, 1)
	(0, 2)	(1, 2)	(2, 2)

Table 3.6: System state table for Example 3.6

It is easy to check ϕ satisfies Definition 3.2 and thus the system is a narrow-sense coherent system with constant binary image, but component 2 is irrelevant to the system.

To relate BW class to other classes more explicitly, we can use the following two conditions provided by Meng (1994):

1. A non-decreasing function $\phi : S^n \rightarrow S$ satisfies condition A: if $\phi(\mathbf{x} + a \cdot \mathbf{1}) = \phi(\mathbf{x}) + a$ for every $\mathbf{x}$ and some $a \geq 0$;

2. A non-decreasing function $\phi : S^n \rightarrow S$ satisfies condition B: if $\phi(\mathbf{x} \cdot a) = a \cdot \phi(\mathbf{x})$ for every $\mathbf{x}$ and $a \geq 0$.

where a is a constant and $a \cdot \mathbf{x} = (ax_1, ax_2, \cdots, ax_n)$.

The structure functions satisfying condition A or B have intersections with other classes of multi-state structures. For example, if an EPS class also satisfies condition

A, then it leads to the BW class. Conditions A and B imply condition 1 and 2 presented by Borges and Rodrigues (1983). But conditions A and B are easier to be applied. At the same time, these two conditions can also be used for a continuous system structure function.

3.2.5 Concluding remarks

As the foundation of multi-state reliability theory, relevancy conditions have received much attention. Various relevancy conditions reveal a world of complex multi-state structures. Multi-state coherent theory is a new development milestone of reliability theory rather than a simple extension of the binary coherent theory. Although many results about multi-state relevancy conditions have been obtained. there are many unexplored questions in this research area. In the following, we summarize some major conclusions:

1. Barlow and Wu (1978) first introduced a multi-state model and extended many concepts from the binary coherent theory. Although their model is simple, it has attracted attention from other authors since it enjoys many nice properties extended from the binary reliability theory.

2. Among various fundamental multi-state relevancy condition, EPS class, GRI1 class and GRI2 class are popularly accepted as strong coherent, coherent and weak coherent conditions. Other relevancy conditions can be classified among these three.

3. Relevancy conditions are so complex that the system structure function can not be mathematically formulated except BW class. In this case, it is almost impossible to know the exact number of system structures under a given relevancy condition for a given component number n and a maximum state M. In some

articles, the authors used some simple examples to show how the structures vary for the given relevancy condition. More work is needed to verify how these structures reflect the problems in real-life.

4. A few authors tried to build up a unified form of relevancy conditions. Their contributions are helpful for analysis of multi-state systems. But there is still not a real unified model which includes all the existing relevancy conditions.

3.3 Binary decomposition of multi-state systems

Relevancy conditions show one aspect of the multi-state reliability theory which is quite different from the binary reliability theory. Another central notion in the multi-state reliability theory is to investigate the inherent relationship between binary systems and multi-state systems. In NAT2 class, we have seen that a multi-state function can be expressed using M binary functions, namely $\phi_1, \cdots, \phi_M$. Similarly, a multi-state variable can be expressed using M binary variables. As a result, a multi-state system can be analyzed from the relationship between the binary system function and the binary variables. Here the questions to be answered are: (1) How to model a multi-state system using the binary decomposition method? (2) How to utilize the binary decomposition method to find multi-state system performance distribution? Block and Savits (1982) provided a solution for the first question with a decomposition theorem for multi-state structure functions. Their result can be used to interpret BW class. The answer for the second question results from another decomposition technique by Wood (1985). His method allows existing binary algorithms for block diagrams to be applied to some multi-state systems such that it is called the multi-state block diagram method. In the followings, we will discuss these two decomposition methods.

3.3.1 BS model (Block and Savits, 1982)

BS Model considers a non-decreasing multi-state system with a structure function ϕ as an MMS by imposing the conditions that $\phi(\mathbf{0}) = 0$ and $\phi(\mathbf{M}) = M$. It does not make the assumption that $\phi(\mathbf{j}) = j$ for $j = 1, \cdots, M-1$ imposed by other classes.

Given a system function ϕ and its minimal path sets denoted by $P_j, j = 1, \cdots, M$, where P_j is the collection of the minimal path sets to level j. To decompose such a function into binary functions, the following procedure is proposed:

1. For each $\mathbf{x} \in P_j$, let $L_j(\mathbf{x}) = \{(i, x_i) : x_i \neq 0\}$; Those components, which are not at state 0 in the minimal path sets to level j, are recognized.

2. Define the binary functions ϕ_k in terms of $M \times n$ binary variables $y = (y_{ij} : 1 \leq i \leq n, 1 \leq j \leq M)$ by

$$\phi_j(y) = \max_{\mathbf{x} \in P_j} \min_{(i,j) \in L_j(\mathbf{x})} y_{ij}$$

   where $j = 1, \cdots, M$; Note that each ϕ_j is a function of $2 \times M$ variables. It implies that the system state is equal to the state of the "worst" component in the "best" minimal path.

3. Define the binary indicator $\alpha(\mathbf{x}) = (\alpha_{ij}(\mathbf{x}) : 1 \leq i \leq n, 1 \leq j \leq M)$, where $\alpha_{ij}(\mathbf{x}) = 1$, if $x_i \geq j$ or $= 0$ otherwise. For a selected component state vector, convert it into a binary vector using the binary indicator first and then compare with the state of the "worst" component in the "best" minimal path.

4. $\phi(\mathbf{x}) = \sum_{j=1}^{M} \phi_j(\alpha(\mathbf{x}))$.

Example 3.7: Decomposition of a multi-state system

Consider a two-component system wherein both the system and the components may be in one of three possible states, 0, 1, and 2. The following table specifies the relationship between system state and component states.

$\phi(\mathbf{x})$:	0	1	2
	(0,0)	(0,1)	(2,0)
$\mathbf{x}$:	(1,0)	(1,1)	(2,1)
		(0,2)	(2,2)
		(1,2)	

Table 3.7: System state table for Example 3.7

The minimal path sets to level 1 are (0,1) and (2,0). $L_1(0,1) = (2,1)$ and $L_1(2,0) = (1,2)$. $y = (y_{11}, y_{12}, y_{21}, y_{22})$. We have:

$$\phi_1(\mathbf{y}) = \max_{\mathbf{x} \in P_1} \min_{(i,j) \in L_1(\mathbf{x})} y_{ij} = \max(y_{21}, y_{12}) \tag{3.2}$$

Equation(3.2) implies that for the system function ϕ_1 to be equal to 1, at least one of the binary variables y_{21} and y_{12} must be equal to 1. For example, we check component state vector $\mathbf{x} = (1,2)$. Since $\alpha(\mathbf{x}) = \alpha(1,2) = (1,0,1,1)$, then $\phi_1(\alpha(\mathbf{x})) = \max\{0,1\} = 1$. Similarly, we can obtain $\phi_2(\mathbf{y}) = y_{12}$ and $\phi_2(\alpha(\mathbf{x})) = 0$. As a result, $\phi(\mathbf{x}) = 1$.

A similar decomposition can be obtained using minimal cut sets. The decomposition theorem provides a general expression for multi-state system functions in terms of binary functions and binary variables. It is analogous to express a system function in terms of minimal path sets in the binary case. Another application of BS model is that some relevancy conditions can be interpreted using the relationship between the binary functions ϕ_j and the binary variables y_{ij}. For example, GRI1 class can be rephrased that ϕ is coherent if and only if y_{ij} is relevant to some ϕ_k for each component i and state $k = 1, 2, \cdots, M$.

3.3.2 Multi-state block diagram method

A block diagram is used by reliability engineers to find the probability distribution for the state of the system. In multi-state systems, the concept of multi-state block diagrams was first introduced by Wood (1985). Binary variables x_{ij} and binary function $\phi^j(\mathbf{x})$ are used to represent the states of the components and the system, where x_{ij} and $\phi^j(\mathbf{x})$ are the binary indicators defined as follows:

$$\phi^j(\mathbf{x}) = \begin{cases} 1, & \text{if } \phi(\mathbf{x}) \geq j \\ 0, & \text{otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1, & \text{if } x_i \geq j \\ 0, & \text{otherwise} \end{cases}$$

The state of component i is the sum of the x_{ij} and the system state $\phi(\mathbf{x})$ is the sum of the $\phi^j(\mathbf{x})$, i.e., $x_i = \sum_{j=1}^{M} x_{ij}$ and $\phi(\mathbf{x}) = \sum_{j=1}^{M} \phi^j(\mathbf{x})$. Obviously, each of $\phi^j(\mathbf{x})$ is a binary monotone structure function which only depends on those binary variables.

$$\phi^j(\mathbf{x}) = \phi^j(x_{11}, \cdots, x_{1M}, x_{21}, \cdots, x_{2M}, \cdots, x_{nM}) \tag{3.3}$$

From Equation (3.3), multi-state block diagrams can be formulated in terms of binary variables for evaluation of system performance distribution at any given level. In a multi-state block diagram, each block represents a binary random variable. If the components in a block diagram are s-independent, the calculation of system reliability is simple, just replace x_{ij} with p_{ij}. Otherwise, conditional probability expansions are required. In the following, we use a simple example to show how a multi-state system can be expressed by multi-state block diagrams.

Example 3.8: Multi-state block diagram

Assume that a multi-state system has 3 components and both the system and its components have 4 states, $j = 0, 1, 2, 3$. Let ϕ be parallel at level 1, 2-out-of-3:G structure at level 2 and series at level 3. Figure 3.1 is the block diagram of the system.

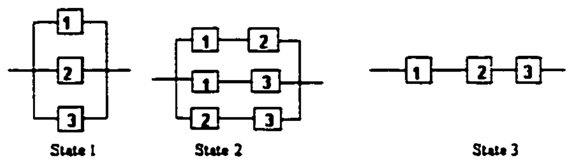

Figure 3.1: Multi-state block diagram for Example 3.8

$$\phi^1(\mathbf{x}) = 1 - (1 - x_{11})(1 - x_{21})(1 - x_{31})$$

$$\phi^2(\mathbf{x}) = 1 - (1 - x_{12}x_{22})(1 - x_{12}x_{32})(1 - x_{22}x_{32})$$

$$\phi^3(\mathbf{x}) = x_{13}x_{23}x_{33}$$

The system performance distribution can be calculated using s-independence for ϕ^1 and ϕ^3, but conditional probability is required for calculating $\Pr(\phi^2(\mathbf{x}) = 1)$:

$$\Pr(\phi^1(\mathbf{x}) = 1) = 1 - (1 - p_{11})(1 - p_{21})(1 - p_{31})$$

$$\Pr(\phi^2(\mathbf{x}) = 1) = p_{12}p_{22} + p_{12}p_{32} + p_{22}p_{32} - 2p_{12}p_{22}p_{32}$$

$$\Pr(\phi^3(\mathbf{x}) = 1) = p_{13}p_{23}p_{33}$$

In Equation (3.3), each binary function ϕ^j is determined by $n \times M$ binary variables. Many of these variables may be irrelevant for the calculation of system performance distribution so the computation efficiency needs to be enhanced. This modeling technique can be extended to analyze multi-state fault-trees.

3.3.3 Concluding remarks

Generally speaking, BS model and the multi-state block diagram method provide a binary decomposition for multi-state systems, which releases the relevancy restriction. These ideas are similar to each other. A multi-state system is treated as M binary systems while its n multi-state components are represented using $n \times M$ binary variables. Multi-state components and MMS functions are represented in terms of binary components and binary structure functions.

BS model relates multi-state system structure functions to binary system structure functions in terms of minimal path sets or minimal cut sets. The model is quite theoretical and not easy to be applied for the calculation of system performance distribution or the computation of its bounds. The decomposition theorem can also be used to explain the relevancy conditions of some other classes.

Compared with BS model, the multi-state block diagram method is much more intuitive. It allows existing binary algorithms to be used for the performance distribution evaluation of multi-state systems. Computer programs for the binary case can be applied to the multi-state case with adjustments in models rather than the programs.

A common disadvantage of both methods is that many irrelevant binary variables are included or involved. In principle, it is possible to eliminate those irrelevant binary variables, but further research is needed.

3.4 Multi-state reliability models and their properties

A key problem in reliability analysis is to find out how reliability characteristics of a complex system can be determined from a knowledge of reliability indices of its components. In the binary case, many well-known systems have been used to model various practical problems, for examples, series and parallel systems, k-out-of-n systems, bridge systems and so on. These simple models can be extended to the multi-state case. But restricted by the complicated relevancy conditions, there has been little work on general multi-state modeling techniques. Alternatively, some authors stress to investigate the properties in multi-state systems and develop their models by imposing some specific properties. In the following, we will review existing multi-state reliability models and their properties.

3.4.1 Multi-state series, parallel and k-out-of-n systems

Series system, parallel system and k-out-of-n system are the most fundamental systems in the binary context. They are then naturally extended to the multi-state context. The definitions of multi-state series, parallel and k-out-of-n systems are first introduced by El-Neweihi et al (1978) as follows:

1. A system is a series system if its structure function satisfies:

$$\phi(\mathbf{x}) = \min_{1\leq i\leq n} x_i$$

2. A system is a parallel system if its structure function satisfies:

$$\phi(\mathbf{x}) = \max_{1\leq i\leq n} x_i$$

3. A system is a k-out-of-n:G system if its structure function satisfies:

$$\phi(\mathbf{x}) = x_{(n-k+1)}$$

where $x_{(1)} \leq x_{(2)} \leq \cdots \leq x_{(n)}$ is an increasing arrangement of $x_1, x_2, \cdots, x_n$.

These definitions are natural generalization of the corresponding binary systems and have been popularly accepted. For example, a k-out-of-n:G system can be defined in term of minimal path sets or minimal cut sets as: ϕ is a k-out-of-n:G structure function if and only if ϕ has $\binom{n}{k}$ minimal path sets to level j and $\binom{n}{k-1}$ minimal cut sets to level j, where $j = 1, 2, \cdots, M$ (Boedigheimer and Kapur, 1994). A series system is a n-out-of-n:G system and a parallel system is a 1-out-of-n:G system.

3.4.2 Equivalent and symmetric relations in multi-state systems

The focus of multi-state reliability modeling is on finding an appropriate definition for the states of the system and its components and their relationship. Equivalent and symmetric relations are two types of relations in multi-state systems.

Equivalent classes of component state vectors

Considering that different components in a system may have different effects on the system, the concept of equivalent classes is introduced for the analysis of multi-state systems (Hudson and Kapur, 1983). Assume $S = \{0, 1, \cdots, M-1, M\}$. A multi-state system consists of $M+1$ equivalent classes $\{S_0, S_1, \cdots, S_M\}$, where $S_j = \{\mathbf{x} \mid \phi(\mathbf{x}) = j\}, j = 0, 1, \cdots, M$. $\{S_0, S_1, \cdots, S_M\}$ partitions the set of component state vector S and S_j is non-empty.

Definition 3.3: Two component state vectors $\mathbf{x}$ and $\mathbf{y}$ are said to be equivalent if $\phi(\mathbf{x}) = \phi(\mathbf{y}) = j$, indicated by $\mathbf{x} \leftrightarrow \mathbf{y}$. If $\mathbf{x} \leftrightarrow \mathbf{y}$, then we say that $\mathbf{x}$ and $\mathbf{y}$ belong to the same equivalence class (Hudson and Kapur, 1983).

For example, assume that there are two component state vectors $(1, 2)$ and $(1, 1)$. If $\phi(1, 2) = 1$ and $\phi(1, 1) = 1$, then we say component state vectors $(1, 2)$ and $(1, 1)$ are in the same equivalent class. In the model proposed by Hudson and Kapur (1983), a system and its components could have different numbers of states. For example, a system is allowed to have 3 states while some of its components may have only 2 states. To find the system performance distribution in each equivalent class, only upper and lower critical vectors to every system level are needed. The relevancy condition of GRI2 class is imposed on this model.

The series, parallel and k-out-of-n systems are defined different from those defined in 3.4.1. A series system is defined such that if at least one component is at state 0, then the system is at state 0, and if all the components are at state M, then the system is at state M. A parallel system is defined such that if all the components are at state 0, then the system is at state 0, and if at least one component is at state M, then the system is at state M. A k-out-of-n structure is defined such that if at least $(n - k + 1)$ components are at state 0, then the system is at state 0, and if at least k components are at state M, then the system is at state M. In other words, only the perfect state (state M) and the failure state (state 0) of a system are concerned. Those intermediate states (states $1, 2, \cdots, M - 1$) of the system are determined by the combinations of the component states, which depend on the definition of each component state. A basic opinion of the authors (Hudson and Kapur, 1983) is to relax the restrictions on modeling a multi-state system as much as possible such that the model can be widely used in engineering practice.

Symmetric relation in multi-state systems

Although minimal path sets and minimal cut sets play important roles in the computation of multi-state system performance distribution, it is usually difficult to find all minimal path sets or minimal cut sets of a multi-state system. Finding a solution for this problem is equivalent to finding a simplified mathematical form of the multi-state structure function. In the binary case, a system structure can be described by a Boolean function. In the multi-state case, a system structure can be described by a discrete function. However, a general discrete function can not be expressed using a simple mathematical form in most cases. Wood (1985) noticed that a multi-state system function is only related to n binary variables in multi-state k-out-of-n structures but he did not make further exploration. This characteristic was further investigated and generalized by Xue and Yang (1995). They defined symmetric function as a special class of discrete functions.

Definition 3.4: Given a multi-state function ϕ, if $\phi(x_1, \cdots, x_s, \cdots, x_t, \cdots, x_n) = \phi(x_1, \cdots, x_t, \cdots, x_s, \cdots, x_n)$, then we say that components s and t are symmetric, indicated by $s \sim t$.

The symmetric relation between two components implies that these two components play the same role in the system.

Definition 3.5: Two component state vectors $\mathbf{x}$ and $\mathbf{y}$ are said to be symmetric if all the components in the vectors are symmetric and $\phi(\mathbf{x}) = \phi(\mathbf{y})$, written as $\mathbf{x} \sim \mathbf{y}$.

$\mathbf{x} \sim \mathbf{y}$ implies that $\mathbf{x}$ can be obtained by some permutation on $\mathbf{y}$. The symmetric relation between two state vectors is a special equivalent relation. ϕ is a symmetric function if all its components are symmetric at all levels. A symmetric relation is common in engineering systems. Series, parallel and k-out-of-n systems all belong to symmetric systems. If $\phi(\mathbf{x})$ is a symmetric function, then its domain S can be represented by its subset s. The number of elements in S is $(M+1)^n$ while the number of elements in s is $\binom{M+n}{M}$ so the domain of ϕ is significantly simplified. Correspondingly, the number of minimal path sets or minimal cut sets in the subset is also decreased.

Concluding remarks

Multi-state reliability models are complicated and it is often difficult or impossible to find a mathematical form for its structure function. However, as illustrated in this section, there exist some special properties in multi-state structure functions. Both equivalent and symmetric relations are good examples of such properties. They provide a valuable tool for simplifying multi-state reliability models.

3.4.3 Generalized multi-state reliability models

In the following, we discuss two generalized multi-state reliability models. In fact, it is hard to say which model is a generalized one since there are so many classes of relevancy conditions. With respect to the meanings of "generalized" here, it at least implies that (1) the weakest relevancy condition, i.e., GRI2 class is used in the model; (2) some restrictions on multi-state reliability models are relaxed. The first model, "Customer-driven reliability model", is proposed by Bodeighemier and Kapur (1994). This model allows a different number of discrete states for a system and for each component. The second model, "Partially-ordered reliability model", releases the restriction that the system states are totally ordered from $0, 1, \cdots, M$. Instead, some states in $\{1, 2, \cdots, M-1\}$ may not be normally ordered.

Customer-driven reliability model (Bodeighemier and Kapur, 1994)

This model is developed based on the model proposed by Hudson and Kapur (1983). It is described as:

Assume $S_0, S_1, \cdots, S_M$ are disjoint sets that partition the multi-state system ϕ into $M+1$ equivalent classes and then ϕ is a coherent system if it satisfies:

1. *ϕ is non-decreasing;*

2. *S_0 and S_M are not empty;*

3. *ϕ belongs to GRI2 class.*

Such a model is called a customer-driven reliability model since it allows the customer to define the states for the system and for each component. In other words, it considers the effect of the component importance on the system. For example, given two component state vectors $(2,3,3)$ and $(3,2,3)$, we define $\phi(2,3,3) < \phi(3,2,3)$ if component 1 is more important to the customer than component 2. The new ideas in this model is that the authors re-defined multi-state series, parallel and k-out-of-n systems since they believe that those definitions of multi-state series systems, parallel systems and k-out-of-n systems proposed by Hudson and Kapur (1983) do not specify which vectors belong to the equivalent classes between S_1 and S_{M-1}. The definitions proposed by Bodeighemier and Kapur (1994) have been discussed in Section 3.4.1. A binary decomposition similar to BS model (Block and Savits, 1982) is developed for calculating the system performance distribution. Brundell and Kapur (1997) renamed such a model as "customer-centered" reliability model by further explaining the relationship between quality of a product and its reliability. Reliability is a time-oriented quality-characteristic. As quality must be defined from the viewpoint of the customer, so must reliability. A series of dynamic measures for the model is developed.

Partially-ordered reliability model (Yu et al, 1994)

Realizing that the states of a system and its components are not necessarily completely ordered, Yu et al (1994) proposed a generalized coherent system model which allows the states of a system and its components to be partially ordered . The key idea in this model is that the system and the components can possess more than one performance parameter, and each of the parameters can degrade independently. For example, neither fail-open nor fail-close of a valve is a degradation from the other. Both the system and the components are assumed to have a perfect functioning state and several performance deterioration modes. Every intermediate state is a degradation from the perfect-functioning state, but not every complete failure state needed to be a degradation from this intermediate state such that there need not exist a smallest element in the state set. The model is described as:

Let E_i be the collection of states of component i and E be the complete collection of component states, $E_i = \{0, 1, \cdots, M_i\}$, and $E = \prod_{i=1}^{n} E_i \rightarrow S$, and S be the collection of system state. The elements in both E and S are partially ordered. The definition of such a MMS is described as follows:

1. Reachability: For any $s_i \in S$, there exists $\mathbf{x} \in E$ such that $\phi(\mathbf{x}) = s_i$;
2. Normality: The mapping $\phi : E \rightarrow S$ maps the greatest element of E onto the greatest element of S, and maps every minimal element of E onto some minimal element of S;
3. Coherence: For any component i and some minimal element of E_i, $e_i^{0_j}$, there exists an $\mathbf{x} \in E$ such that $\phi((e_i^{M_j})_i, \mathbf{x}) > \phi((e_i^{0_j})_i, \mathbf{x})$.

The reachability implies that for every intermediate state s_i of the system, there is at least one component state vector $\mathbf{x}$ such that $\phi(\mathbf{x}) = s_i$. The normality indicates that if all the components are in the perfect functioning state, then the system will be in the perfect-functioning state. The different combinations of the intermediate states of the components may result in the system to be in different intermediate states. For example, consider a 3-state system with two components. ϕ can take values from $\{0, 1, 2\}$, where 2 is the perfect state while 0 and 1 indicate two failure states. At the same time, the components also take values from $\{0, 1, 2\}$, where 2 is the perfect state while 0 and 1 are two failure states. The normality is equivalent to that $\phi(2,2) = 2$ and possibly $\phi(1,0) = 1$ and $\phi(0,1) = 0$. The relevancy condition used in this model is equivalent to GRI2 class. An interesting phenomenom is that many common properties contained in other totally-ordered models are lost in a partially-ordered model. For example, the dual function of a partially-ordered system does not always exist and its structure function ϕ is not always bounded above by a series structure and below by a parallel structure. However, if some of its subsets is totally ordered, these properties do exist in this subset. The concepts of minimal path sets and minimal cut sets in the totally ordered case are extended to the partially ordered case. Since the expression of the system structure function in terms of minimal path sets or minimal cut sets is quite different from the totally ordered case, there may exist several paths along which the system can degrade from the perfect functioning state to some complete failure state, the system reliability is re-defined and bounds for the system performance distribution are derived.

In summary, these two models consider wider application conditions for multi-state reliability models. However, both of them have their limitations. The customer-driven reliability model re-defined multi-state series, parallel and k-out-of-n systems. In these definitions, the systems are assumed to have a constant structure at all levels. In this case, the relevancy condition should be GRI1 class rather than GRI2 class. Otherwise, if a system could have different structures at different levels, then the definitions do not apply. We will further discuss this issue in Chapter 6. In the second model, the relation between the totally ordered case and the partially ordered case has not been well studied.

3.4.4 Continuous multi-state reliability models

We have been considering discrete multi-state reliability models, i.e., the state spaces of both the system and the components are a set of discrete numbers: $\{0, 1, 2, \cdots, M-1, M\}$. However, a state space could be continuous in some applications. For example, a system state can be described by a continuous variable if its percentage effort or percentage capacity is recorded. In this case, the states of a system and its components can take values from an interval. Ross (1979) proposed that a multi-state system or a component may take performance levels from an interval. The first continuous model was proposed by Block and Savits (1984). An integral representation for a general multi-state structure function is defined. Various concepts corresponding to discrete models, for example, upper sets and extreme points corresponding to path sets and minimal path sets, are proposed. A decomposition similar to BS model (Block and Savits, 1982) is also developed.

One of the major difficulties in implementing continuous state reliability analysis is its mathematical complexity. Yang and Xue (1997) discuss how to simplify and use continuous random processes in product reliability modeling and how to use statistical methods to estimate parameters in reliability models. Zuo et al (1999) investigated three approaches for reliability modeling of continuous state devices. The first uses the random process to fit model parameters of a statistical distribution as functions of time. The second approach uses the general path model to fit model parameters as functions of time. The third uses multiple linear regression to fit the distribution of lifetime directly. Generally speaking, not many results have been reported on continuous models.

3.4.5 Dynamic multi-state reliability models

Markov processes have been used to analyze dynamic multi-state systems. Here "dynamic" means time-dependent. Many results about dynamic multi-state reliability models have been reported, for examples, Karpinski (1986); Cao and Wu (1988); Levitin and Lisinanski (2000) and so on. Regarding to Markov processes, readers may refer to the references by Chung (1979). In a Markov process, the system state is expressed using a transition matrix of the component states rather than the system structure function. For examples, Cafaro, Corsi and Vacca (1986) have shown how to use Markov models for evaluating the availability and reliability of a system when the transition rates of each component depend on the state of the system. The structural properties of the system transition rate matrices based on the transition-rate matrices of the components are analyzed. The key achievement of their work is the potential of using parallel computing resources with this approach, which promises a reduction in computing time, thus renewing interest in Markov models. Lesanovsky (1988) dealt with a system that can be described by a homogeneous continuous time discrete state Markov process. The case when transition rates of each unit depend on the current state of the system is considered. Usually, Markov process models consider certain inspection or repair activities. For example, Huang and Wu (1990) used the Markov decision model to analyze the maintenance activity of a shovel-truck system for an open-pit mine. A shovel-truck system consists of a fleet of trucks. The system state is defined as the monthly production amount of the shovel while the component state as the monthly production amount of the trucks. Various maintenance and repair activities are considered. The transition probability of the component state with or without maintenances and repairs is calculated. The objective of this model is to optimize the expected gross benefit in infinite time span. By means of dynamic programming, the optimal maintenance and repair periods matching such an objective are obtained. Buihan and Allan (1995) and Bai and Zhang (1997) analyzed large power system reliability applying Markov processes.

At present, Markov process is still a powerful tool for analyzing dynamic multi-state system reliability. However, this technique has certain limitations: first, it requires the system to have Markov property, i.e, the sojourn time distribution in each system state is exponentially distributed. This assumption is not always satisfied. If this property is not applicable to a system, more complicated stochastic process models are required. For example, Yieh (1996) used a semi-Markov model to find the optimal inspection and replacement policies for a multi-state deteriorating system. For such a model, mathematical formulation becomes so complicated that it is essentially intractable. An approximation method must be used to transform a semi-Markov process into a Markov process in this case. Secondly, as the number of components increases, the system state transition matrix becomes very large so it can only be applied to small multi-state systems. To analyze large a multi-state systems, we have to use system structure function to define system states.

A different approach for the dynamic reliability analysis of multi-state systems is to combine Markov processes and multi-state system theory (Xue and Yang, 1995; Amari and Misra, 1997). In such a analysis, Markov process technique is used to describe the dynamic characteristics of component state transition, and through the system structure function, the dynamic characteristics of the system are analyzed. Usually, a system structure function can be expressed by the sum-of-product form of its minimal path sets at every level (Xue, 1985). By generalizing reliability measure parameters in the binary case to the multi-state case, the problem of dynamic reliability analysis of multi-state systems can be transformed into a series of dynamic reliability analysis of binary systems. The generalized multi-state reliability parameters include system reliability $R(t)$, unreliability $F(t)$, failure rate $\lambda(t)$ and MTTF

(Mean-Time-To-Failure). Existing methods of reliability testing and parameter estimation and system modeling can be applied for multi-state reliability. This method also requires the system to have Markov property.

Example 3.9: Dynamic system reliability measures of a series structure

Consider a series system ϕ with n components. The system performance distribution $R(t, j)$ is defined as $R(t, j) = \Pr(\phi(t) \geq j)$, where $j = 0, 1, \cdots, M$. Then

$$R(t, j) = \prod_{i=1}^{n} R_i(t, j)$$

where $R_i(t, j) = \exp[-\int_0^t \lambda_i(u, j)du]$ is the performance distribution function of component i and $\lambda_i(t, j)$ is the transition rate for component i from state set $\{j, j + 1, \cdots, M\}$ to state set $\{0, 1, \cdots, j - 1\}$ at time t. And then the system transition rate

$$\lambda(t, j) = \sum_{i=1}^{n} \lambda_i(t, j)$$

In particular, if all the components are i.i.d., i.e., $\lambda_i(t, j) = \lambda(j)$, then $R(t, j) = \exp[-n\lambda(i)t]$.

There are many other issues on dynamic multi-state system reliability models, for example, multi-state fault-tree analysis by Xue (1985); the redundancy optimization problem for series-parallel multi-state systems by Levitin et al (1998); the structure optimization problem of multi-state system with time redundancy by Lisinanski et al (2000).

3.4.6 Concluding remarks

Generally speaking, the techniques of multi-state reliability modeling are still in its infant stage of development. The key problem is how to find a simple expression for a multi-state system structure function. More attention should be paid to common properties of multi-state systems, for examples, equivalent relation and symmetric relation.

3.5 System performance distribution evaluation

As discussed in Chapter 2, almost all algorithms for reliability evaluation of binary systems can be extended to the multi-state case. But determining the exact system performance distribution of a multi-state system can involve extremely large amounts of computations because of its huge state space and complex relevancy conditions. Alternatively, finding the bounds of a system performance distribution is more practical sometimes. Basically, we can divide the algorithms for performance evaluation of a multi-state systems into three types: decomposition methods, other methods and bounding techniques. Decomposition methods include binary decomposition, pivotal decomposition and modular decomposition. Other methods include Inclusion-Exclusion method(IE) and Sum-of-Disjoint-Product method(SDP). We will review these algorithms one by one.

3.5.1 Finding minimal path sets and minimal cut sets

In a multi-state system, the minimal path sets and minimal cut sets play important roles in computing the system performance distribution or establishing the bounds of the system performance distribution. As a result, finding all minimal path sets or minimal cut sets is very important. A multi-state system can always be expressed in terms of its minimal path sets and minimal cut sets using the lattice exponentiation form of a discrete function (Xue and Yang, 1995). In other words, the complexity of the structure function of a multi-state system is proportional to the number of its minimal path sets or minimal cut sets. Hudson and Kapur (1983) illustrated the existence of minimal path sets and minimal cut sets for any levels of multi-state systems. But the computation of finding all the minimal path sets of a multi-state system could be overwhelming and using the enumeration method may be more effective in some cases (Xue, 1985).

From the viewpoint of system theory, any large system has a hierarchical structure. Thus the structure function of a large system may be in hierarchical decomposition form (Xue, 1985). Modular decomposition is a commonly-used hierarchical decomposition method for a large system. If a system has a complete modular decomposition, then its minimal path sets at level j can be found by implementing: (1) finding the minimal path sets of the sub-systems; (2) obtaining the minimal path sets of the modular system; (3) obtaining the minimal path sets of the original system. If a multi-state system does not have a modular decomposition form, an enlarged-system is needed. Since it is required to delete some vectors which are not the minimal path sets of the original system from Step (2) in order to obtain Step (3), this method is still not a direct way for finding minimal path sets of a system. Actually, whether a component state vector is a minimal path set or not depends on the relevancy condition of the system. A minimal path to system level j could result in the system to be in a higher level. Thus it is not easy to find an efficient method for finding minimal path sets of multi-state systems. We will discuss the properties of minimal path sets of multi-state systems further in Chapter 6.

3.5.2 Decomposition methods

Pivotal decomposition

Pivotal decomposition is a commonly-used tool for reliability evaluation of binary systems. This method can be used for calculating the exact performance distribution of multi-state systems (El-Neweihi et al, 1978; Hudson and Kapur, 1983; Korczak, 1993). The system performance function of n components may be expressed in terms of system performance functions of $n-1$ components by:

$$h(\mathbf{p_1}, \mathbf{p_2}, \cdots, \mathbf{p_n}) = \sum_{j=0}^{M} p_{ij} h(j_i; \mathbf{p_1}, \mathbf{p_2}, \cdots, \mathbf{p_n}) \tag{3.4}$$

where $h(j_i; \mathbf{p_1}, \mathbf{p_2}, \cdots, \mathbf{p_n}) = E\phi(X_1, \cdots, X_{i-1}, j, X_{i+1}, \cdots, X_n)$, and $\mathbf{p_i} = (p_{i0}, \cdots, p_{iM})$. A decomposition identity is as below:

$$\Pr(\phi(\mathbf{x}) \geq j) = \sum_{j=0}^{M} \Pr(\phi(\mathbf{x}) \geq j \mid X_i = j) \Pr(X_i = j) \tag{3.5}$$

Binary decomposition

In Section 3.3, we have discussed two binary decomposition methods presented by Block and Savits (1982) and Wood (1985). The common idea of these two methods is: dividing a multi-state system into M binary structures and a multi-state variable into M binary variables. Usually, the following indicators are used for such transformations: $\phi^j(\mathbf{x}) = 1$ if $\phi(\mathbf{x}) \geq j$ and $\phi^j(\mathbf{x}) = 0$ otherwise; $x_{ij} = 1$ if $x_i \geq j$ and $x_{ij} = 0$, otherwise. Thus a multi-state system structure function can be expressed using Equation (3.3). The problem is that mutual s-independence among components may be lost in this equation even components are assumed s-independent. For example, let $\mathbf{x} = (3,2,0), \mathbf{y} = (3,2,1)$ and $\mathbf{z} = (3,1,0)$, $\phi(\mathbf{x}) = \phi(\mathbf{y}) = 3$ while $\phi(\mathbf{z}) = 2$. then we can write: $\phi^3(\mathbf{x}) = \phi^3(\mathbf{y}) = 1$, $\phi^3(\mathbf{z}) = 0$. However, if now we express these three vectors $\mathbf{x}$, $\mathbf{y}$ and $\mathbf{z}$ in binary variable vectors for system state 3, they all become $(1,0,0)$. We obtain a conflicting conclusion. It seems that the same binary vector $(1,1,0)$ is giving different system state indicator values (0 for $\mathbf{z}$ and 1 for $\mathbf{x}$ and $\mathbf{y}$). To avoid this contradiction, all decomposition methods for a multi-state systems requires minimal path sets or minimal cut sets to be known first before the calculation of system performance distribution. Wood (1985) claimed that binary algorithms can be applied for evaluation of system performance distribution by means of block diagrams. Indeed, his method also requires to find minimal path sets of systems first and then either the pivotal decomposition or conditional probability method is applied.

To consider a binary decomposition for multi-state systems, we need to answer the following questions first:

1. Under what conditions can a multi-state system be divided into a dichotomous system at the same level as its components? In other words, when is a multi-state system dichotomous at all levels?

2. How to eliminate the s-dependence among the binary components so that we can find an equivalent binary system for a multi-state system at any level? In other words, when can existing binary algorithms be used for multi-state systems directly?

We will discuss these two issues in Chapter 6.

Modular decomposition

For the analysis of a large system, modular decomposition is often used. The modular decomposition method can be applied for either exact system performance evaluation or bound computation. In Chapter 2, we have reviewed the definition of modular decomposition and the measurement of decomposition efficiency proposed by Hudson and Kapur (1983). Based on this measurement, we can judge whether the system performance distribution computation using modular decomposition is more efficient than without the decomposition.

One of the main reasons for modular decomposition of a system is to facilitate the calculation or approximation of system performance distribution. In most cases, by using the modular decomposition, the computations involved in computing bounds can be greatly reduced and the bounds produced are as good as or better than other approaches, for examples, IE and SDP methods (Butler, 1982).

In the binary case, the procedure of obtaining reliability bounds using modular decomposition is to determine lower bounds (or upper bounds) on the reliability of each module first, and then to treat these lower bounds(or upper bounds) as if they were actual reliabilities of the modules in determining a lower bound (upper bound) for the reliability of the overall system. Such a procedure can not be directly extended to multi-state systems since the bounds on the performance of each module at every level will not in general constitute a performance-distribution vector (Butler, 1982), where the performance-distribution vector is defined as a vector $v = (v_0, v_1, v_2, \cdots, v_M)$ satisfying $0 \leq v_i \leq 1$ and $v_0 = 1$. For example, the performance distribution of a multi-state component i $(P_{i0}, P_{i1}, \cdots, P_{iM})$ is a performance distribution vector. For a multi-state system, we have to construct a performance distribution vector for the module from these lower bounds (or upper bounds) such that the results in the binary case can be generalized to the multi-state case. Butler (1982) described the procedures for constructing such performance distribution vectors for modules. Examples are given to illustrate that the bounds with the modular decomposition is better than a direct approach.

One of the advantages of the modular decomposition is avoiding to find minimal path sets and minimal cut sets. The problem is that not all multi-state systems can be modularly decomposed and there is not an easy shortcut for judging whether a multi-state system can be modularly decomposed or not.

3.5.3 Other methods

IE and SDP methods

Both IE and SDP methods can be used for performance distribution evaluation of multi-state systems. The IE method follows the same procedure as in the binary case (El-Neweihi et al, 1978; Natvig, 1980; Butler, 1982; Boedigheimer and Kapur, 1985).

Let $P_1^j, P_2^j, \cdots\cdots, P_r^j$ be r minimum paths to level j, $j = 1, 2, \cdots M$. To calculate $R_{sj} = P(\phi \geq j)$, let A_i^j denote that $\mathbf{x} \geq P_i^j, i = 1, 2, \cdots, r$.

$$R_{sj} = \Pr(\phi \geq j) = \Pr(\bigcup_{i=1}^{r} A_i^j) \tag{3.6}$$

Let $S_k = \sum_{1 \leq i_1 < i_2 < \cdots i_k \leq r} \Pr(A_{i_1} \cap A_{i_2} \cdots \cap A_{i_k})$

By the Inclusion-Exclusion principle, we have:

$$P(\phi \geq j) = \sum_{k=1}^{r} (-1)^{k-1} S_k$$

And then a sequence of upper and lower bounds on R_{sj} can be obtained:

$$R_{sj} \leq S_1$$
$$R_{sj} \geq S_1 - S_2$$
$$R_{sj} \leq S_1 - S_2 + S_3$$
$$\vdots$$

The exact value of R_{sj} can be found by using IE method continuously.

SDP method for multi-state systems is extended by Hudson and Kapur (1983) and Korczak (1993). Again let $P_1^j, P_2^j, \cdots\cdots, P_r^j$ be the minimal paths to level j, $j = 1, 2, \cdots M$. Define sets $B_k^j = \{\mathbf{x} \mid \mathbf{x} \geq P_k^j\}$ for $k = 1, \cdots, r$ and $T_k^j = \Pr(\mathbf{x} \in \cup_{i=1}^{k} B_i^k) = \Pr(\mathbf{x} \in B_1^j \cup B_2^j \cup \cdots \cup B_k^j)$. Then the exact system performance distribution can be generated from a sequence of lower bounds $\{T_1^j, T_2^j, \cdots, T_M^j\}$:

$$\Pr(\phi(\mathbf{x}) \geq j) \geq T_k^j$$

The calculation of each T_k^j is shown by Hudson and Kapur (1983). The bounds resulting from SDP are always between 0 and 1 and SDP only generates lower bounds, but the bounds are strictly monotone increasing. Technically, both of IE and SDP can result in satisfactory bounds. SDP is better when computation time is concerned (Hudson and Kapur, 1983).

Other algorithms

Aven (1985) reported two algorithms for evaluation of multi-state system performance distribution, which are based on Doulliez and Jamoulle decomposition method (Doulliez and Jamoulle, 1972), in which the state space is decomposed into certain sets: a set of acceptable states, a set of non-acceptable states, and sets of unspecified states. Algorithm 1 requires known minimal path sets. Regarding to level j, a component state vector $\mathbf{x}$ is acceptable if and only if $\phi(\mathbf{x}) \geq j$ and non-acceptable if and only if $\phi(\mathbf{x}) < j$. Algorithm 2 requires known minimal cut sets. A component state vector $\mathbf{x}$ is acceptable if and only if $\phi(\mathbf{x}) < j$ and non-acceptable if and only if $\phi(\mathbf{x}) \geq j$. Each unspecified state is again decomposed and so forth until there are no unspecified states left. The fundamental idea is that solving a group of sub-systems, which are obtained by decomposing a large system, is easier than solving the whole system. But the algorithms are not well explained. Comparisons with IE method and state enumeration method are performed. The presented algorithms seem to be better.

3.5.4 Simple bounding techniques

Although modular decomposition and IE and SDP can be used to find the exact system performance distribution, it is usually very time-consuming. Alternatively, finding bounds is preferred sometimes. As we know, both IE and SDP are commonly-used bounding techniques. The modular decomposition methods can also be used to generate bounds. Many simple formulas have been developed to bound the performance distribution of a multi-state system. The simplest bounding formula is:

$$\min x_i \leq \phi(\mathbf{x}) \leq \max x_i,$$

i.e., $\phi(\mathbf{x})$ is bounded below by the series structure and above by the parallel structure. Of course, this formula generates the loosest bounds. To obtain sharper bounds, a bounding technique using minimal path sets and minimal cut sets is developed (Butler, 1982). Consider a multi-state coherent system ϕ with a set of minimal path $P_1, \cdots, P_s$ and a set of minimal cut $K_1, \cdots, K_t$ at level j. Define the min-path structure function $P_j(\mathbf{x})$ and the min-cut structure function $K_j(\mathbf{x})$ as: $P_j(\mathbf{x}) = \prod_{i=1}^{n} I_{(x_i \geq x_{ik})}$ for $k = 1, \cdots, s$, $x_{ik} = \{x_i \mid i \in P_k\}$; and $K_j(\mathbf{x}) = 1 - \prod_{i=1}^{n} I_{(x_i \leq x_{ik})}$ for $k = 1, \cdots, t$, $x_{ik} = \{x_i \mid i \in K_k\}$, where $I_{(\bullet)}$ is the indicator function.

- Path-Cut bounds:

$$\prod_{k=1}^{t} \Pr(K_j(\mathbf{x}) = 1)) \leq \Pr(\phi(\mathbf{x}) \geq j) \leq \prod_{k=1}^{s} \Pr(P_j(\mathbf{x}) = 1))$$

- Max-Min bounds:

$$\max_{1 \leq j \leq s} \{\Pr(P_j(\mathbf{x}) = 1)\} \leq \Pr(\phi(\mathbf{x}) \geq j) \leq \min_{1 \leq j \leq t} \{\Pr(K_j(\mathbf{x}) = 1)\}$$

The Path-Cut bounds can be applied to associated components and provides a lower bound based upon cuts and an upper bound based upon paths. The Max-Min bounds can only be applied to independent components and provide a lower bound based upon paths and an upper bound based upon cuts. Similar to the binary case, the Max-Min lower bound is usually better than the Path-Cut lower bound for systems composed of very unreliable components, and the Path-Cut lower bound is better for systems composed of highly reliable components. The Max-Min upper bound is better than the Path-Cut lower bound for systems composed of highly reliable components, and the Path-Cut lower bound is better for systems composed of very unreliable components. If only one-sided bound is required, the above comments do not apply. We classify these bounds as simple bounding techniques as they do not bound system reliability in a tight form.

Utkin (1993) presented methods to evaluate variance, interval width and fuzziness importance of multi-state component. Since it is difficult to estimate precise probabilities of multi-state component states, one may use stochastic, fuzzy and interval description, which are characterized by uncertainty measures, variance, interval width and fuzziness respectively, of these probabilities.

3.5.5 Concluding remarks

In summary, almost all existing algorithms for evaluation of multi-state system performance distribution or bounds are extended from the binary case. Among them, the modular decomposition seems to have excellent potential.

3.6 Research motivation

In this chapter, we have reviewed most of the publications on multi-state reliability theory, which are published in the span between 1978 and 2000. Generally speaking, the papers published before 1990 concentrated on establishing the fundamental theory of multi-state system reliability, including relevancy conditions and the extensions of various concepts from the binary case to the multi-state case. Papers published since 1990 mainly aim to illustrate the applications of multi-state models in practical engineering. A framework of multi-state reliability theory has been established. There were two important issues on multi-state reliability theory: relevancy conditions and the relationship between binary systems and multi-state systems.

Various multi-state relevancy conditions have been discussed. Multi-state relevancy conditions are so complicated that there are many different versions. A major drawback of the existing work is that most of the papers did not illustrate how their definitions are applicable to real-world problems. The focus of this thesis is not on relevancy conditions.

The multi-state reliability theory extends various concepts and theorems from the binary reliability theory. When a multi-state system has the same structure at all state levels, many properties of binary systems exist in such a system. However, in general, a multi-state system may have different structures at different state levels. When this happens, such a multi-state system usually has different properties from binary systems, for example, the dual property illustrated in Example 2.3. Under various relevancy conditions except the BW class, multi-state systems may have different structures at different levels. To investigate the relationship between binary systems and multi-state systems, the following issues will be addressed in this thesis.

1. The existing definitions of multi-state series, parallel and k-out-of-n systems assume that the systems have a constant structure at different system states. Such systems are similar to binary systems. We will define generalized multi-state k-out-of-n and consecutive k-out-of-n systems, in which the systems are allowed to have different structures at different state levels, and then investigate their properties. The definitions to be proposed will be more general than the existing ones and have wider applications;

2. The inherent relationship between binary systems and multi-state systems is an interesting topic. Wood (1985) noticed that the indicator functions of some multi-state systems are only related to n binary variables. However, he did not study the conditions under which such a phenomenon exists. We will investigate the equivalent conditions under which a multi-state system can be treated like a binary system. Finding such conditions will be beneficial for simplifying the calculation of multi-state system reliability;

3. Finding efficient algorithms for reliability evaluation of multi-state systems is usually difficult, especially for large systems. Although all binary algorithms can be extended to the multi-state case, the computation time for multi-state systems rapidly increases as M and n increase. More efficient algorithms are to be developed.

In summary, the theory of multi-state systems is still under development. There exist various problems to be addressed in order to make the multi-state system model a useful tool in practical system modeling. This Ph.D. is motivated to address some of these issues in order to make multi-state models more useful to engineers.

Chapter 4

Generalized Multi-state k-out-of-n Systems

4.1 Introduction

The k-out-of-n system is one of the fundamental models in the binary case. It finds wide applications in both industrial and military areas. Lately a few researchers have extended the definitions of binary parallel, series, and k-out-of-n:G systems to the multi-state cases by allowing both the system and its components to have more than two possible states. As we have reviewed in Chapter 3, the state of a multi-state k-out-of-n:G system is defined to be equal to the state of the kth best component (El-Neweihi et al 1978) or as a system with $\binom{n}{k}$ lower boundary points to system state j ($j = 1, 2, \ldots, M$) and $\binom{n}{k-1}$ upper boundary points to system state j ($j = 1, 2, \cdots, M$) (Boedigheimer and Kapur, 1994). These definitions of the k-out-of-n:G system are consistent with one another. The series and the parallel systems are special cases of the k-out-of-n:G system. A series system is a n-out-of-n:G system. A parallel system is a 1-out-of-n:G system. For the k-out-of-n:G system, maintaining at least a certain system state level requires the same number of components to be at that state or above. That is, at least k components must be at state j or better for the system to be at state j or better ($j = 1, 2, \cdots, M$), where k is independent of the value j of the system state level. Such a k-out-of-n system is the simplest case because it requires the system to have the same structure at all system levels.

However, a multi-state system may have different structures at different system levels. For a multi-state k-out-of-n system, it implies that k is allowed to take different values. For example, considering a 3-component system with four possible states. The system could be a 1-out-of-3:G structure at level 1, in other words, it requires at least one component to be in state 1 or above for the system to be in state 1 or above. It may have a 2-out-of-3:G structure at level 2, in other words, for the system to be in state 2 or above, at least 2 components must be in state 2 or above. It may have a 3-out-of-3:G structure at level 3, namely, at least three components have to be in state 3 for the system to be in state 3. Such a k-out-of-n:G system is more flexible for describing some real-life problems. In this chapter, we first propose a definition of the generalized multi-state k-out-of-n:G system in Section 4.2. Section 4.3 provides examples to illustrate the ideas and applications of the proposed definition. In Section 4.4, reliability evaluation algorithms are provided for the proposed multi-state k-out-of-n:G systems. Section 4.5 presents concluding remarks.

Notation and Assumptions

Notation:

n	number of components.
$M+1$	number of states of the system and its components.
x_i	state of component i, $x_i \in \{0, 1, \cdots, M\}$.
$\mathbf{x}$	vector of component states, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$.

$\phi(\mathbf{x})$	system structure function representing the state of the system, $\phi(\mathbf{x}) \in \{0, 1, \ldots, M\}$.
k_j	minimum number of components with $x_i \geq j$ for $j = 1, 2, \cdots, M$.
k	a constant when $k_1 = k_2 = \cdots = k_M = k$.
P_{ij}	$\Pr(x_i \geq j)$.
P_j	$P_j = P_{ij}$ when the components are i.i.d.
p_{ij}	$\Pr(x_i = j)$.
p_j	$p_j = p_{ij}$ when the components are i.i.d.
q_{ij}	$\Pr(x_i < j)$.
R_{sj}	$\Pr(\phi(\mathbf{x}) \geq j)$
r_{sj}	$\Pr(\phi(\mathbf{x}) = j)$
MP_i^j	The ith lower boundary point or min path to level j

Assumptions:

1. The system is a multi-state monotone system (Griffith 1980):
   - $\phi(\mathbf{x})$ is non-decreasing in each argument.
   - $\phi(\mathbf{j}) = \phi(j, j, \ldots, j) = j$ for $j = 0, 1, \cdots, M$.
2. The x_i's are mutually s-independent.

4.2 Definition of multi-state k-out-of-n:G system

The definition of the multi-state k-out-of-n:G system is proposed as follows:

Definition: *$\phi(\mathbf{x}) \geq j$ ($j = 1, 2, \ldots, M$) if there exists an integer value l ($j \leq l \leq M$) such that at least k_l components are in states at least as good* *as l. An n-component system with such a property is called a multi-state k-out-of-n:G system.*

In this definition, k_j's do not have to be the same for different system states j ($1 \leq j \leq M$). This means that the structure of the multi-state system may be different for different system state levels. Generally speaking, k_j's values are not necessarily in a monotone ordering. But the following two special cases of this definition will be particularly considered:

- When $k_1 \leq k_2 \leq \cdots \leq k_M$, the system is called an increasing multi-state k-out-of-n:G system. In this case, for the system to be at a higher state level j, a larger number of components must be at state j or above. In other words, there is an *increasing* requirement on the number of components that must be at a certain state or above for the system to be at a higher state level. That is why we call it the increasing multi-state k-out-of-n:G system.

- When $k_1 \geq k_2 \geq \cdots \geq k_M$, the system is called a decreasing multi-state k-out-of-n:G system. In this case, for a higher system state level j, there is a *decreasing* requirement on the number of components that must be at state level j or above. That is why we call it the decreasing multi-state k-out-of-n:G system

When k_j is a constant, i.e., $k_1 = k_2 = \cdots = k_M = k$, the structure of the system is the same for all system state levels. This reduces to the definitions of the multi-state k-out-of-n:G system in the literature as reviewed in Section 4.1. We call such systems constant multi-state k-out-of-n:G systems. All the concepts and results of binary k-out-of-n:G systems can be easily extended to the constant multi-state k-out-of-n:G systems. We will treat the constant multi-state k-out-of-n:G system as a special case of the increasing multi-state k-out-of-n:G system in our later discussions.

4.3 Illustrative examples

In this section, we provide examples to illustrate the ideas and applications of the proposed multi-state k-out-of-n:G systems.

Example 4.1: An increasing multi-state k-out-of-n:G system:

Consider a three-component system with $k_1 = 1$, $k_2 = 2$, and $k_3 = 3$. Both the system and the components may be in one of four possible states, namely, 0, 1, 2, and 3. The following table illustrates the relationship between system state and component states.

$\phi(\mathbf{x})$:	0	1	2	3
	(0,0,0)	(1,0,0)	(2,2,0)	(3,3,3)
		(1,1,0)	(2,2,1)	
		(1,1,1)	(2,2,2)	
		(2,0,0)	(3,2,0)	
$\mathbf{x}$:		(2,1,0)	(3,2,1)	
		(2,1,1)	(3,2,2)	
		(3,0,0)	(3,3,0)	
		(3,1,0)	(3,3,1)	
		(3,1,1)	(3,3,2)	
		+	+	

Table 4.1: System state table for Example 4.1

In this table, the "+" sign in the table represents the permutations of the component states listed above the "+" sign. Because $k_1 < k_2 < k_3$ in this case, we have a simpler version of the proposed definition of the multi-state k-out-of-n:G system. The system is in state 3 if all three components are in state 3. The system is in state 2 or above if at least two components are in state 2 or above. The system is in state 1 or above if at least one component is in state 1 or above.

The system in this example has a series structure at system state 3 (3-out-of-3:G), a 2-out-of-3:G structure at system state 2, and a parallel structure at system state 1 (1-out-of-3:G).

Example 4.2: A decreasing multi-state k-out-of-n:G System:

Consider a three-component system wherein both the system and the components may be in one of four possible states, 0, 1, 2, and 3. A decreasing k-out-of-n:G system satisfies the relationship between system state and component states as specified in the following table. In this example, we have $k_1 = 3$, $k_2 = 2$, and $k_3 = 1$.

$\phi(\mathbf{x})$:	0	1	2	3
	$(0,0,0)$	$(1,1,1)$	$(2,2,0)$	$(3,0,0)$
	$(1,0,0)$	$(2,1,1)$	$(2,2,1)$	$(3,1,0)$
	$(2,0,0)$	+	$(2,2,2)$	$(3,1,1)$
	$(1,1,0)$		+	$(3,2,0)$
$\mathbf{x}$:	$(2,1,0)$			$(3,2,1)$
	+			$(3,2,2)$
				$(3,3,0)$
				$(3,3,1)$
				$(3,3,2)$
				$(3,3,3)$
				+

Table 4.2: System state table for Example 4.2

Again, the "+" sign represents the permutations of the states of the components listed above the "+" sign. For example, system state level 2 can result from component states (2, 2, 0) and their permutations, namely, (0, 2, 2) and (2, 0, 2). This is because all the components in a k-out-of-n:G system perform the same functions.

In terms of the definition of a multi-state k-out-of-n:G system we have provided, the system in this example is in state 3 if at least one component is in state 3 ($k_3 = 1$). The system is in state 2 or above if at least two components are in state 2 or above ($k_2 = 2$) or at least one component is in state 3 ($k_3 = 1$). The system is in state 1 or above if all three components are in state 1 or above ($k_1 = 3$), or at least two components are in state 2 or above ($k_2 = 2$), or at least one component is in state 3 ($k_3 = 1$). We can see that in this example, $k_1 > k_2 > k_3$ indicating a strictly decreasing multi-state k-out-of-n:G system.

We can say that the system in this example has a 1-out-of-3:G structure at system state 3, a 2-out-of-3:G structure at system state 2, and a 3-out-of-3:G structure at system state 1.

Example 4.3: A constant multi-state k-out-of-n:G system:

Consider a three-component system wherein both the system and the components may be in one of four possible states, 0, 1, 2, and 3. A constant multi-state 2-out-of-3 system satisfy the following relationship between system state and the states of the components:

Again, the "+" sign represents the permutations of the component states listed above the "+" sign. Because $k_1 = k_2 = k_3 = 2$, we also have a simpler version of the proposed definition of the multi-state k-out-of-n:G system in this case. The system is in state 3 if at least two components are in state 3. The system is in state 2 or above if at least two components are in state 2 or above. The system is in state 1 or above

$\phi(\mathbf{x})$:	0	1	2	3
$\mathbf{x}$:	(0, 0, 0)	(1, 1, 0)	(2, 2, 0)	(3, 3, 0)
	(1, 0, 0)	(1, 1, 1)	(2, 2, 1)	(3, 3, 1)
	(2, 0, 0)	(2, 1, 0)	(2, 2, 2)	(3, 3, 2)
	(3, 0, 0)	(2, 1, 1)	(3, 2, 0)	(3, 3, 3)
	+	(3, 1, 0)	(3, 2, 1)	+
		(3, 1, 1)	(3, 2, 2)	
		+	+	

Table 4.3: System state table for Example 4.3

if at least two components are in state 1 or above. The system has a 2-out-of-3:G structure at each of the system states 1, 2, or 3.

Example 4.4: A production management problem:

Suppose that a plant has five production lines for producing a certain product. The plant has four different production levels: full scale for maximum customer demand (state 3), average scale for normal customer demand (state 2), low scale when the customer demand is low (state 1), and zero scale when the plant is shut down. All the five production lines have to work full scale (at state 3) for the system to be in state 3. At least three lines have to work at least at the average scale for the system to be at least in state 2 or above. At least 2 lines have to work at least at the low scale (state 1) for the system to be in state 1 or above. Such a system can be represented by an increasing multi-state k-out-of-n:G system model with $k_1 = 2$, $k_2 = 3$, and $k_3 = 5$.

Example 4.5: A mining operation example:

Take a shovel-truck system in an open-pit mine as an example. A shovel-truck system usually consists of a shovel and a fleet of trucks (say 20 trucks). The deteriorating rates of the trucks are often higher than that of the shovel. Each truck and the system may be in five possible states. The system will be in state 4 if at least 14 trucks are in state 4. The system is in at least state 3 if at least 15 trucks are in at least state 3 or at least 14 trucks are in state 4. The system is in at least state 2 if at least 16 trucks are in at least state 2, or at least 15 trucks are in at least state 3, or at least 14 trucks are in state 4. The system is in at least state 1 if at least 18 trucks are in at least state 1, or at least 16 trucks are in at least state 2, or at least 15 trucks are in at least state 3, or at least 14 trucks are in state 4. This system can be represented by a decreasing multi-state k-out-of-n:G system with $k_1 = 18$, $k_2 = 16$, $k_3 = 15$, and $k_4 = 14$.

From these examples, we can see that multi-state k-out-of-n:G systems are more flexible to describe practical problems than binary k-out-of-n:G systems.

4.4 System performance evaluation

A commonly-used approach for multi-state system performance evaluation is to extend the results from existing binary algorithms. In binary systems, the "domination phenomenon" always exists. A vector $\mathbf{x}$ dominates another vector $\mathbf{y}$ if $\mathbf{x} \geq \mathbf{y}$ and $\phi(\mathbf{x}) \geq \phi(\mathbf{y})$. This phenomenon also exists in the increasing or constant multi-state k-out-of-n: G systems. That's why the wordings are simpler for these systems as illustrated in the examples. In these cases, the system is at state j or above if at least k_j components are at state j or above. If the states equal to j or above are treated as "functioning" and the states below j are treated as "failure", these multi-state systems can be treated the same as how binary systems are treated. Thus, the algorithms for binary k-out-of-n:G systems can be extended to the increasing or constant k-out-of-n:G systems. But for decreasing k-out-of-n:G systems, the domination phenomenon does not exist with all vectors. For instance, in Example 4.1, (2, 2, 2) results in system state 2 and (3, 0, 0) results in system state 3. However, vector (2, 2, 2) does not dominate vector (3, 0, 0). Both vectors $(2,2,0)$ and $(3,0,0)$ and their permutations are the minimal paths to system level 2. If we use them to calculate $\Pr(\phi(\mathbf{x}) \geq 2)$, there exists an overlapping problem for vectors $(3,2,2)$, $(3,3,2)$, $(3,3,3)$ and their permutations. The binary algorithm can not be used directly in the decreasing systems. In the followings, we discuss the algorithms for performance evaluation of the increasing systems and the decreasing systems separately.

Case I: Increasing or constant k-out-of-n:G systems, i.e., $k_1 \leq k_2 \leq \cdots \leq k_M$:

In this case, the definition of a multi-state k-out-of-n:G is equivalent to that $\phi(\mathbf{x}) \geq j$ if and only if at least k_j components have $x_i \geq j$.

If at least k_j components are in at least state j (these components can be considered "functioning" as far as state level j is concerned), then the system will be in state j or above (the system is considered to be "functioning"). The algorithms for binary k-out-of-n:G system reliability evaluation by Barlow and Heidtmann (1982) and Rushdi (1986) can be extended in this case for multi-state k-out-of-n:G system performance distribution evaluation:

$$R_j(n,k_j) = P_{nj}R_j(n-1,k_j-1) + (1-P_{nj})R_j(n-1,k_j), \tag{4.1}$$

where $R_j(a,b)$ is the probability that at least b out of a components are at state j or above. The following boundary conditions are needed in equation (4.1):

$$R_j(a,b) = 0, \quad \text{for} \quad b > a > 0, \tag{4.2}$$

$$R_j(a,0) = 1, \quad \text{for} \quad a \geq 0, \tag{4.3}$$

where $R_j(n,k_j)$ is the probability that the system is in state j or above.

When all the components have the same state probability distribution, i.e., $p_{ij} = p_j$ for all i, the probability that the system is in at least state j, R_{sj} can be expressed as:

$$R_{sj} = \sum_{k=k_j}^{n} \binom{n}{k} P_j^k (1 - P_j)^{n-k} \tag{4.4}$$

Equations (4.1) and (4.4) are similar to (2.13) and (2.14). Once we find $\Pr(\phi(\mathbf{x}) = j)$ or $\Pr(\phi(\mathbf{x}) \geq j)$ for all j values, we have found the probability distribution of the system in different states. The probability that the system is in state j can be calculated as:

$$r_{sj} = R_{sj} - R_{s(j+1)}. \tag{4.5}$$

Example 4.6:

In Example 4.2, assume $p_0 = 0.1$, $p_1 = 0.3$, $p_2 = 0.4$, $p_3 = 0.2$. We can use Equation (4.4) to calculate the system probabilities at all levels.

Solution:

We have $P_1 = 0.9, P_2 = 0.6, P_3 = 0.2$.

At level 3, $k_3 = 3$. $R_{s3} = P_3^3 = 0.2^3 = 0.008$

At level 2, $k_2 = 2$. $R_{s2} = \binom{3}{2} \times 0.6^2 \times (1 - 0.6) + 0.6^3 = 0.648$

At level 1, $k_1 = 1$. $R_{s1} = \binom{3}{1} \times 0.9 \times (1 - 0.9)^2 + \binom{3}{2} \times 0.9^2 \times (1 - 0.9) + 0.9^3 = 0.999$

The system probabilities at all levels are as follows:

$r_{s3} = 0.008$

$r_{s2} = R_{s2} - R_{s3} = 0.64$

$r_{s1} = R_{s1} - R_{s2} = 0.351$

$r_{s0} = 1 - 0.008 - 0.64 - 0.351 = 0.001$

Case II: Decreasing k-out-of-n:G systems, i.e., $k_1 \geq k_2 \geq \cdots \geq k_M$

We assume that at least one of the inequalities is a strict inequality because otherwise the system is a constant k-out-of-n:G system. When k_i is the same for all i values, we can use the formulas for Case I to calculate system probability distribution. In Case II, the definition of the multi-state k-out-of-n:G system is equivalent to the following:

> $\phi(\mathbf{x}) = j$ if and only if at least k_j components are at or above state j and at most $k_l - 1$ components are at state l or above for $l = j+1, j+2, \cdots, M$ where $j = 1, 2, \cdots, M$.

In the following, we will separate the case with i.i.d. components and the case with non i.i.d. components. When all the components have the same state probability distribution, the following equation can be used to calculate the probability that the system is in state j:

$$r_{sj} = \sum_{k=k_j}^{n} \binom{n}{k} \left(\sum_{m=0}^{j-1} p_m \right)^{n-k} \left(p_j^k + \sum_{l=j+1, k_l>1}^{M} \beta_l(k) \right), \tag{4.6}$$

where $\beta_l(k)$ is the probability that there are at least one and at most $k_l - 1$ components that are in state l, at most $k_u - 1$ components that are in state u for $j \le u < l$, and the total number of components that are at states between j and l inclusive is k.

As shown in Equation (4.6), $n - k$ components are in states below j and the remaining k components must be in state j or above. At the same time, all these component states must make sure that the system is exactly in state j. In Equation (4.6), we are summing up the probabilities that there are exactly k components that are in state j or above without bringing the system state above j for $k = k_j, k_j + 1, \ldots, n$ where $\beta_l(k)$ is dependent on the value of k. The quantity in the last parentheses is the probability that at least k components are in state j or above without causing the system to be in a state above j. Within the last parentheses, p_j^k is the probability that all the k components are exactly in state j, $\beta_{j+1}(k)$ is the probability that there are at least one and at most $k_{j+1}-1$ components that are in state $j+1$ and the other components are in state j. When $l=j+2$, $\beta_l(k)=\beta_{j+2}(k)$ is the probability that there are at least one and at most $k_{j+2}-1$ components that are in state $j+2$, at most $k_{j+1}-1$ components are in state $j+1$, and the remaining components are in state j. Generally speaking, $\beta_l(k)$ is the probability that there are at least one and at most k_l-1 components that are in state l, and at most k_u-1 components that are in state u for $j\le u<l$. If $k_u=1$ for any u, $j+1\le u\le M$, there is no need to calculate $\beta_l(k)$ for $l>u$ because k_l is non-increasing. To calculate $\beta_l(k)$, we can use the following equation:

$$\begin{aligned}\beta_l(k) &= \sum_{i_1=1}^{k_l-1}\binom{k}{i_1}p_l^{i_1}\sum_{i_2=0}^{k_{l-1}-1-i_1}\binom{k-i_1}{i_2}p_{l-1}^{i_2}\times\cdots\\ &\times\sum_{i_{l-j}=0}^{k_{j+1}-1-I_{l-j-1}}\binom{k-I_{l-j-1}}{i_{l-j}}p_{j+1}^{i_{l-j}}p_j^{k-I_{l-j}}\end{aligned}\tag{4.7}$$

where $I_{l-j-1}=\sum_{m=1}^{l-j-1}i_m$ and $I_l=\sum_{m=1}^{l-j}i_m$. The programming procedure for calculating β_l is as follows:

1. Calculating β_{j+1} at state $j+1$.

$\beta_{j+1}=\sum_{i_1=1}^{k_{j+1}-1}\binom{k}{i_1}p_{j+1}^{i_1}p_j^{k-i_1}$

let $s_1=0$

for $i_1=1$ to $k_{j+1}-1$

$\alpha_1=\binom{k}{i_1}p_{j+1}^{i_1}p_j^{k-i_1}$

$s_1=s_1+\alpha_1$

next i_1

$\beta_{j+1}=s_1$

2. Calculating β_{j+2} at state $j+2$.

Here, we still allow $r=1\sim k_{j+2}-1$ components are in state $j+2$, $k-r$ components are in state j or $t=1\sim k_{j+1}-1-r$ components are in state $j+1$ and other $k-r-t$ components are still in state j.

$\beta_{j+2} = \sum_{i_1=1}^{k_{j+2}-1} \binom{k}{i_1} p_{j+2}^{i_1} \sum_{i_2=0}^{k_{j+1}-i_1-1} \binom{k-i_1}{i_2} p_{j+1}^{i_2} p_j^{k-i_1-i_2}$

let $s_1 = 0$ and $s_2 = 0$

for $i_1 = 1$ to $k_{j+2} - 1$

$\alpha_1 = \binom{k}{i_1} p_{j+2}^{i_1}$

$I_1 = i_1$

for $i_2 = 0$ to $k_{j+1} - 1 - I_1$

$I_2 = I_1 + i_2$

$\alpha_2 = \binom{k-I_1}{i_2} p_{j+1}^{i_2} p_j^{k-I_2}$

$s_2 = s_2 + \alpha_2$

next i_2

$s_1 = s_1 + s_2\alpha_1$

$s_2 = 0$

next i_1

$\beta_{j+2} = s_1$

3. Calculating β_l at state l

Basically, we can extend the above algorithm to state l, here $k_l > 1$.

$\beta_l = \sum_{i_1=1}^{k_l-1} \binom{k}{i_1} p_l^{i_1} \cdots \sum_{i_{l-j}=0}^{k_{j+1}-1-I_{l-j-1}} \binom{k-I_{l-j-1}}{i_{l-j}} p_{j+1}^{i_{l-j}} p_j^{k-I_l-j}$

here, $I_{l-j-1} = \sum_{m=1}^{l-j-1} i_m$ and $I_l = \sum_{m=1}^{l-j} i_m$

let $s_1 = 0; s_2 = 0; \cdots s_{l-j} = 0$

for $i_1 = 1$ to $k_l - 1$

$\alpha_1 = \binom{k}{i_1} p_l^{i_1}$

$I_1 = i_1$

for $i_2 = 0$ to $k_{l-1} - 1 - I_1$

$I_2 = i_2 + I_1$

$\alpha_2 = \binom{k-I_1}{i_2} p_{l-1}^{i_2}$

for $i_3 = 0$ to $k_{l-1} - 1 - I_2$

$I_3 = i_3 + I_2$

$\alpha_3 = \binom{k-I_2}{i_3} p_{l-2}^{i_3}$

$\vdots$

for $i_{l-j} = 0$ to $k_{j+1} - 1 - I_{l-j-1}$

$I_{l-j} = i_{l-j} + I_{l-j-1}$

$\alpha_{l-j} = \binom{k-I_{l-j-1}}{i_{l-j}} p_{j+1}^{i_{l-j}} p_j^{k-I_{l-j}}$

$s_{l-j} = s_{l-j} + \alpha_{l-j}$

next i_{l-j}

$s_{l-j-1} = s_{l-j-1} + s_{l-j}\alpha_{l-j-1}$

$s_{l-j} = 0$

next i_{l-j-1}

$\vdots$

next i_3

$s_2 = s_2 + s_3\alpha_2$

$s_3 = 0$

next i_2

$s_1 = s_1 + s_2\alpha_1$

$s_2 = 0$

next i_1

$\beta_l = s_1.$

The number of terms to be summed up in Equation (4.7) is equal to the number of ways to assign k identical balls to $l - j$ different cells with at least 1 and at most $k_l - 1$ balls in cell l, at least 0 and at most $k_t - 1$ balls in cells t for $j < t < l$, and the remaining balls in cell j. The complexity for such a problem can not be expressed using a simple mathematical form (Chung, 1979). A computer program has been developed based on the programming procedure above and we have found the computation time for $\beta_l(k)$ in Equation (4.7) will be much smaller than k^{l-j} where $k = \max\{k_j, j = 1, 2, \ldots, M\}$. In turn, the computation time for r_{sj} in Equation (4.6) is much less than nk^M. For most practical engineering problems, a limited state number of M, for example $M = 10$, is big enough to describe the performances of the system and its components. Thus, the proposed algorithm is practical.

Example 4.7:

Given a 4-component multi-state system, assume both the system and its components may be in state 0, 1, 2, 3, or 4. Let $k_1 = 4$, $k_2 = 3$, $k_3 = 2$, $k_4 = 1$. The components are assumed to be i.i.d. with $p_0 = 0.1$, $p_1 = 0.2$, $p_2 = 0.3$, $p_3 = 0.3$, $p_4 = 0.1$. Use the above algorithm to calculate the system probabilities at all levels.

Solution:

At level 4, $k_4 = 1$. By Equation (4.6), $r_{s4} = \sum_{k=1}^{4} \binom{4}{k} \left(\sum_{m=0}^{3} p_m \right)^{4-k} p_4^k = \sum_{k=1}^{4} \binom{4}{k} (0.9)^{4-k} \times 0.1^k = 0.3439$

At level 3, $k_3 = 2$. By Equation (4.6), $r_{s3} = \sum_{k=2}^{4} \binom{4}{k} \left(\sum_{m=0}^{2} p_m \right)^{4-k} p_3^k = \sum_{k=2}^{4} \binom{4}{k} (0.6)^{4-k} \times 0.3^k = 0.2673$

At level 2, $k_2 = 3$. Since $k_3 > 1$, we need to find the expression of $\beta_3(k)$. Based on Equation (4.7), $\beta_3 = \sum_{i_1=1}^{1} \binom{k}{i_1} p_3^{i_1} p_2^{k-i_1} = \binom{k}{1} 0.3 \times 0.3^{k-1} = k \times 0.3^k$.

$r_{s2} = \sum_{k=3}^{4} \binom{4}{k} \left(\sum_{m=0}^{1} p_m \right)^{4-k} \left(p_2^k + \beta_3(k) \right) = \sum_{k=3}^{4} \binom{4}{k} (0.3)^{4-k} (0.3^k + \beta_3(k)) = 0.1701.$

At level 1, $k_1 = 4$. Since $k_2 > 1$ and $k_3 = 2 > 1$, we need to find $\beta_2(k)$ and $\beta_3(k)$. By Equation (4.7), $\beta_2 = \sum_{i_1=1}^{2} \binom{4}{i_1} p_2^{i_1} p_1^{k-i_1}$ and $\beta_3 = \sum_{i_1=1}^{1} \binom{4}{i_1} p_3^{i_1} \left(\sum_{i_2=0}^{1} \binom{3}{i_2} p_2^{i_2} p_1^{3-i_2} \right)$.

By Equation (4.6), $r_{s1} = \sum_{k=4}^{4} \binom{4}{4} p_0^{4-4} \left(p_1^k + \sum_{l=2}^{3} \beta_l(k) \right) = 0.2^4 + \beta_2(k) + \beta_3(k) = 0.0856.$

$r_{s0} = 1 - r_{s1} - r_{s2} - r_{s3} - r_{s4} = 0.1331.$

When the components are non i.i.d., the following equation can be used to calculate the system probability at level j.

$$r_{sj} = \sum_{k=k_j}^{n} \left\{ R_e^j(k,n) + \sum_{l=j+1, l>1}^{M} r_s^j(l) \right\}, \tag{4.8}$$

where $R_e(k,n)$ is the probability that there are exactly k components that are in state j and the other components are below state j and $r_s^j(l)$ is the probability that at least one and at most $k_l - 1$ components are at state l, at most $k_u - 1$ components are at state u for $j \leq u < l$, the total number of components at state j or above is equal to k, and $n - k$ components are at states below j. The first term in Equation (4.8), $\sum_{k=k_j}^{n} R_e^j(k,n)$ is the probability that there are at least k_j components that are in state j and the other components are below state j.

$R_e^j(k,n) = p_{nj} R_e^j(k-1, n-1) + q_{nj} R_e^j(k, n-1)$

$R_e^j(k,k) = \prod_{i=1}^{k} p_{ij}$

$R_e^j(1,k) = \sum_{i=1}^{k} q_{1j} q_{2j} \cdots q_{(i-1)j} p_{ij} q_{(i+1)j} \cdots q_{kj}$

To calculate $r_s^j(l)$, we use the following equation:

$$r_s^j(l) \quad = \sum_{i_1=1}^{k_l-1} \sum_{i_2=0}^{k_{l-1}-1-i_1} \cdots \sum_{i_{l-j}=0}^{k_{j+1}-1-I_{l-j-1}} R((l^{i_1}, (l-1)^{i_2}, \cdot, (j+1)^{i_{l-j}}, j^{k-I_{l-j}}), n) \tag{4.9}$$

where, $I_{l-j-1} = \sum_{m=1}^{l-j-1} i_m$ and $I_{l-j} = \sum_{m=1}^{l-j} i_m$

In Equation (4.9), let $R((l^{i_1}, (l-1)^{i_2}, \cdots, (j+1)^{i_{l-j}}, j^{k-I_{l-j}}), n) = R((l \sim j), n)$ be the probability that there are exactly i_1 components at level l, i_2 components at level $l-1, \cdots$, i_{l-j} components at level $j+1$, $k - I_{l-j}$ components at level j, and the other components at states below j out of n components. It can be calculated using the following recursive relation:

$$R((l \sim j), n) \quad = \quad p_{nl} R((l^{i_1 - 1}, *), n-1) + p_{n(l-1)} R(((l-1)^{i_2 - 1}, *), n-1) + \cdots$$

$$+p_{n(j+1)}R(((j+1)^{i_{l-j}-1},*),n-1)+p_{nj}R((j^{k-I_{l-j}-1},*),n-1)$$
$$+q_{nj}R((l\sim j),n-1) \qquad (4.10)$$

Here, $R((l^{i_1-1},*),n-1) = R((l^{i_1-1},l-1^{i_2},\cdots,j+1^{i_{l-j}},j^{k-I_{l-j}}),n-1)$ and so on.

The boundary conditions are as follows:

$R((l^0,(l-1)^0,\cdots,i^1,\cdots,(j+1)^0,j^0),1) = p_{1i}$

$R((l^0,(l-1)^0,\cdots,(j+1)^0,j^0),1) = q_{1j}$

$R((l^0,(l-1)^0,\cdots,(j+1)^0,j^0),n) = \prod_{k=1}^{n} q_{kj}$

$R((l^0,(l-1)^0,\cdots,i^n,\cdots,(j+1)^0,j^0),n) = \prod_{k=1}^{n} p_{ki}$

$R((l^0,(l-1)^0,\cdots,i^1,\cdots,(j+1)^0,j^0),n) = \sum_{k=1}^{n} q_{1j}q_{2j}\cdots p_{ki}\cdots q_{nj}$

The computation complexity for r_{sj} in the non-i.i.d. case is much less than n^2k^M, where $k = \max\{k_i, i=1,2,\ldots,n\}$.

Example 4.8:

There is a three-component multi-state k-out-of-n system. The component's probabilities satisfy the following table:

Probabilities			
State	1	2	3
0	0.1	0.1	0.1
1	0.2	0.1	0.2
2	0.3	0.2	0.4
3	0.4	0.6	0.3

Assume $k_1 = 3, k_2 = 2, k_3 = 2$. Use Equations (4.8), (4.9) and (4.10) to calculate the system probabilities at all levels.

Solution

At level 3, $k_3 = 2$, $r_{s3} = \sum_{k=k_3}^{n} R_e^3(k,n) = \sum_{k=2}^{3} R_e^3(k,n)$

When $k = 2$, $R_e^3(2,3) = p_{33}R_e^3(1,2)+q_{33}R_e^3(2,2) = p_{33}(p_{13}q_{23}+q_{13}p_{23})+q_{33}p_{13}p_{23} = 0.324$

When $k = 3$, $R_e^3(3,3) = p_{13}p_{23}p_{33} = 0.072$

$r_{s3} = 0.324 + 0.072 = 0.396$

At level 2, $k_2 = 2, r_{s2} = \sum_{k=k_2}^{n} R_e^2(k,n) + \sum_{l=3}^{3} r_s^2(l) = \sum_{k=2}^{3} R_e^2(k,n) + r_s^2(l)$

When $k = 2$,

$R_e^2(2,3) = p_{32}R_e^2(1,2) + q_{32}R_e^2(2,2) = p_{32}(p_{12}q_{22} + q_{12}p_{22}) + q_{32}p_{12}p_{22} = 0.066$

$r_s^2(3) = R((3^{i_1}, 2^{k-i_1}), 3) = R((3^1, 2^1), 3) = p_{33}R((3^0, 2^1), 2) + p_{32}R((3^1, 2^0), 2) + q_{32}R((3^1, 2^1), 2) = p_{33}(p_{12}q_{22}+q_{12}p_{22})+p_{32}(p_{13}q_{22}+q_{13}p_{23})+q_{32}(p_{12}p_{23}+p_{13}p_{22}) = 0.14$

When $k = 3$, $R_e^2(3,3) = p_{12}p_{22}p_{32} = 0.024$

$r_s^2(3) = R((3^{i_1}, 2^{k-i_1}), 3) = R((3^1, 2^2), 3) = p_{33}R((3^0, 2^2), 2) + p_{32}R((3^1, 2^1), 2) = p_{33}p_{22}p_{12} + p_{32}(p_{13}p_{22} + p_{12}p_{23}) = 0.2$

$r_{s2} = 0.066 + 0.024 + 0.14 + 0.2 = 0.43$

At level 1, $k_1 = 3, r_{s1} = \sum_{k=k_1}^{n} R_e^1(k,n) + \sum_{l=2}^{3} r_s^1(l) = R_e^1(3,3) + r_s^1(2) + r_s^1(3)$

$R_e^1(3,3) = p_{11}p_{21}p_{31} = 0.004$

$r_s^1(2) = R((2^{i_1}, 1^{k-i_1}), 3) = R((2^1, 1^2), 3) = p_{32}R((2^0, 1^2), 2) + p_{31}R((2^1, 1^1), 2) = p_{32}p_{21}p_{11} + p_{31}(p_{12}p_{21} + p_{11}p_{22}) = 0.022$

$r_s^1(3) = R((3^{i_1}, 2^{i_2}, 1^{k-i_1-i_2}), 3) = R((3^1, 2^0, 1^2), 3) = p_{33}R((3^0, 2^0, 1^2), 2) + p_{31}R((3^1, 2^0, 1^1), 2) = p_{33}p_{21}p_{11} + p_{31}(p_{23}p_{11} + p_{13}p_{21}) = 0.038$

$r_{s1} = 0.004 + 0.022 + 0.038 = 0.064$

$r_{s0} = 1 - 0.064 - 0.43 - 0.396 = 0.11$

As we see in this example, the algorithm for the performance evaluation of the decreasing k-out-of-n:G system is complex. Alternatively, various bounding techniques can be used to establish bounds on the decreasing systems. In the following example, we use IE method to see how to establish the bounds of a decreasing k-out-of-n:G system.

Example 4.9:

Consider a 3-component decreasing k-out-of-n system. Both of the system and its component may be in state 0, 1, 2, or 3. Let $k_1 = 3, k_2 = 3, k_3 = 1$. The components are assumed to be i.i.d. with $p_0 = 0.1, p_1 = 0.2, p_2 = 0.3, p_3 = 0.4$. Use Inclusion-Exclusion method to find the system performance distribution.

Solution

The performance probability distribution of the components are: $P_1 = 0.9, P_2 = 0.7, P_3 = 0.3$

At level 3, the minimal paths are $MP_1^3 = (3,0,0), MP_2^3 = (0,3,0), MP_3^3 = (0,0,3)$ and let $A_1^3 = \{\mathbf{x} \geq MP_1^3\}, A_2^3 = \{\mathbf{x} \geq MP_2^3\}, A_3^3 = \{\mathbf{x} \geq MP_3^3\}$

$S_1 = \sum_{i=1}^{3} \Pr(A_i^3) = 3 \times 0.4 = 1.2$

$S_2 = \sum_{1 \leq i < j \leq 3} \Pr(A_i^3 \cap A_j^3) = 3 \times 0.4^2 = 0.48$

$S_3 = \Pr(A_1^3 \cap A_2^3 \cap A_3^3) = 0.4^3 = 0.064$

$R_{s3} = S_1 - S_2 + S_3 = 0.784$

For level 2 or above, the minimal paths are $MP_1^2 = (2,2,2), MP_2^2 = (3,0,0), MP_3^2 = (0,3,0), MP_4^2 = (0,0,3)$ and let $A_1^2 = \{\mathbf{x} \geq MP_1^2\}, A_2^2 = \{\mathbf{x} \geq MP_2^2\}, A_3^2 = \{\mathbf{x} \geq MP_3^2\}$, $A_4^2 = \{\mathbf{x} \geq MP_4^2\}$.

$S_1 = \sum_{i=1}^{4} \Pr(A_i^2) = 0.7^3 + 3 \times 0.4$

$S_2 = \sum_{1 \leq i < j \leq 4} \Pr(A_i^2 \cap A_j^2) = 3 \times 0.4^2 + 3 \times 0.4 \times 0.7^2$

$S_3 = \sum_{1 \leq i < j < k \leq 4} \Pr(A_i^2 \cap A_j^2 \cap A_k^2) = 3 \times 0.4^2 \times 0.7 + 0.4^3$

$S_4 = \Pr(A_1^2 \cap A_2^2 \cap A_3^2 \cap A_4^2) = 0.4^3$

$R_{s2} = S_1 - S_2 + S_3 - S_4 = 0.811$

Similarly, for level 1 or above, the minimal paths are $MP_1^1 = (1, 1, 1)$, $MP_2^1 = (3, 0, 0)$, $MP_3^1 = (0, 3, 0)$, $MP_4^1 = (0, 0, 3)$

$R_{s1} = 0.9^3 + 3 \times 0.4 - 3 \times 0.4^2 - 3 \times 0.4 \times 0.9^2 + 3 \times 0.4^2 \times 0.9 + 0.4^3 - 0.4^3 = 0.909$

$r_{s3} = 0.784$

$r_{s2} = R_{s2} - R_{s3} = 0.027$

$r_{s1} = R_{s1} - R_{s2} = 0.098$

$r_{s0} = 1 - R_{s1} = 0.091$

From this example, IE method seems to be simpler than the proposed algorithm. However, no matter applying IE or SDP method, one has to find all minimal paths of the system first. This could be time-consuming. As the number of minimal paths is increased, the computation of IE will be increased rapidly. For performance evaluation of some multi-state systems, it may be time-saving by using the enumeration method than finding minimal paths to establish the bounds (Xue, 1985). The decreasing k-out-of-n system is an example of such cases.

4.5 Concluding remarks

In this chapter, we propose a definition of the generalized k-out-of-n:G system. The proposed systems are more flexible than those previously defined in the literature. The algorithms for the system performance evaluation are developed. Examples are used to illustrate the ideas of the proposed definition. The generalized k-out-of-n:G system provides an example about how to model a multi-state system. The following summerizes some issues for our further consideration.

1. The binary algorithms can be extended to the constant and increasing k-out-of-n:G systems, but not to the decreasing systems. This is determined by the different properties of their minimal path sets. More work is needed to investi-

gate how a general multi-state system can be treated like a binary system.

2. Since the algorithm for the decreasing system is enumerative in nature. More work is needed to find other efficient bounding techniques or formulas for the decreasing system.

3. In the binary case, there exists duality and equivalent relationships between k-out-of-n:G and k-out-of-n:F systems. In this chapter, we only dealt with multi-state k-out-of-n:G systems. It deserves to investigate the multi-state k-out-of-n:F systems and the relationships between the multi-state k-out-of-n:G and k-out-of-n:F systems.

Chapter 5

Multi-state Consecutive k-out-of-n Systems

5.1 Introduction

In binary k-out-of-n structures, consecutive k-out-of-n system is one class of the most important. We have reviewed the definitions of binary consecutive k-out-of-n:F and consecutive k-out-of-n:G systems and the formulas for the system reliability evaluation in Chapter 2. Many research results have been reported on such systems, for example, see Chiang and Niu (1981), Hwang (1982), Kuo et al. (1994) and Chao et al. (1995). An example often used to interpret a consecutive k-out-of-n:F system is to add a component 0 (source) and a component $n + 1$ (sink) to the system and assume that each component, if working, is directly connected to the subsequent k components (or all the remaining components if the number is less than k), and that the source and the sink always work. The system works if and only if a flow can be sent from the source to the sink. One limitation of such a system is that it does not model the system in which components have different transmitting capabilities.

Lately a few researchers have partially extended the definition of binary consecutive k-out-of-n:F system to the multi-state case by allowing the system to remain binary and its components to have more than two possible states. Shanthikumar (1987) extended the consecutive k-out-of-n:F system to the *consecutively-connected system* (CCS) by assuming that component i, if working, is directly connected to the subsequent k_i components at probability p_{ij}. The system works if and only if there is a connection from the source to the sink through functioning components. A CCS can be either a linear system or a circular system depending on whether the components are arranged in a line or a circle. A CCS allows the modeling of systems in which components have different transmitting capabilities (different states). The algorithms for the linear CCS reliability evaluation has been developed by Hwang and Yao (1989) and Kossow and Preuss (1995). Zuo and Liang (1993) and Malinowski and Preuss (1995) provided the algorithms for the circular CCS reliability evaluation. Malinowski and Preuss (1996) also investigated a 2-way linear CSS system.

Sometimes, it is necessary to allow both the system and its components in a consecutive k-out-of-n:F system to experience more than two possible states. Hiam and Porat (1991) provides a Bayes reliability model of the consecutive k-out-of-n:F system, in which both the system and its components are assumed to have more than two possible states while k is assumed to be constant. When k is constant, the system has the same reliability structure at all system states. However, a multi-state system may have different structures at different system states or levels. The following example illustrates this point.

Example 5.1: A quality control problem

A batch of products may be labeled one of the following three classes based on the level of quality: Grade A, Grade B, and Rejected. The following sampling procedure is used to classify the product items: if consecutive 3-out-of-10 items of a sample do not meet the standard of Grade A, then a subsequent inspection is conducted under the standard of Grade B; otherwise, it is labeled Grade A. If consecutive 5-out-of-10 items of a sample are judged to be lower than Grade B, then this batch will be rejected; otherwise, it is labeled Grade B. For such a problem, we can define a multi-state consecutive k-out-of-n:F system with the label of the batch as system state and the sampled items as components. Both the system and the components have three possible states: State 2 (Grade A), state 1 (Grade B) and state 0 (rejected). At the system state level 2, it has a consecutive 3-out-of-10:F structure and at the system state level 1, it has a consecutive 5-out-of-10:F structure.

In this chapter, Section 5.2 proposes a definition of the multi-state consecutive k-out-of-n:F system. Under the proposed definition, a different number of consecutive components is needed to be below level j for the system to be below level j for different j values. The required number of consecutive component "failures" is dependent on the system state level under consideration. Two special cases are taken into consideration: the decreasing consecutive k-out-of-n:F system and the increasing consecutive k-out-of-n:F system. Section 5.3 discusses system performance evaluation algorithms for the proposed multi-state consecutive k-out-of-n:F system. In Case I, if the system is a decreasing system, then the binary algorithm can be applied on it directly. In Case II, if the system is an increasing system, then the binary algorithm can not be extended on it. Section 5.4 develops a bounding technique to establish the bounds of system state distribution for increasing consecutive k-out-of-n:F systems. Concluding remarks are provided in Section 5.5.

Notation and Assumptions

Notation:

k_j	minimum number of consecutive components to be in states below j
Q_{ij}	$\Pr(x_i < j)$.
Q_j	$Q_j = Q_{ij}$ when the components are i.i.d.
$F_j(n; k_j)$	probability that at least k_j consecutive components are in states below j for an n component system
$R_j(n; k_j)$	$1 - F_j(n; k_j)$
F_{sj}	$1 - R_{sj}$.

Other notions are the same as Chapter 4.

Assumptions:

1. The system is a multi-state monotone system (Griffith 1980):
   - $\phi(\mathbf{x})$ is non-decreasing in each argument.
   - $\phi(\mathbf{j}) = \phi(j, j, \ldots, j) = j$ for $j = 0, 1, \cdots, M$.
2. The x_i's are mutually s-independent.

5.2 The multi-state consecutive k-out-of-n:F system

We propose a definition of the multi-state consecutive k-out-of-n:F system as follows:

Definition: $\phi(\mathbf{x}) < j$ $(j = 1, 2, \cdots, M)$ *if at least* k_l *consecutive components are in states below* l *for all* l *such that* $j \leq l \leq M$*. An* n*-component system with such a property is called a multi-state consecutive* k*-out-of-*n*:F system.*

In this definition, k_j's do not have to be the same for different system states j ($1 \leq j \leq M$). This means that the structure of the multi-state system may be different for different system state levels. A few examples will be given to illustrate this definition. The following two special cases of this definition will be taken into consideration:

- When $k_1 \geq k_2 \geq \cdots \geq k_M$, the system is called a decreasing multi-state consecutive k-out-of-n:F system. In this case, for the system to be below a higher state level j, a smaller number of consecutive components must be below state j. In other words, as j increases, there is a *decreasing* requirement on the number of consecutive components that must be below state j for the system to be below state level j.

- When $k_1 \leq k_2 \leq \cdots \leq k_M$, the system is called an increasing multi-state consecutive k-out-of-n:F system. In this case, for the system to be below a higher state level j, a larger number of consecutive components must be below state j. In other words, as j increases, there is an *increasing* requirement on the number of consecutive components that must be below a state j for the system to be below state level j.

When k_j is constant, i.e., $k_1 = k_2 = \cdots = k_M = k$, the structure of the system is the same for all the system state levels. This reduces to the definition of the multi-state consecutive k-out-of-n:F system provided by Hiam and Porat (1991). We call such a system constant consecutive k-out-of-n:F system. We consider the constant consecutive-k-out-of-n:F system as a special case of the decreasing consecutive-k-out-of-n:F system.

Example 5.2: A decreasing consecutive k-out-of-n:F system

Consider a three-component system wherein both the system and the components may be in one of three possible states, 0, 1, and 2. The system state and the component states have the relationships as shown in the following table. In this example, we have $k_1 = 2$ and $k_2 = 1$.

$\phi(\mathbf{x})$:	0	1	2
$\mathbf{x}$:	$(0,0,0)$	$(0,1,0)$	$(2,2,2)$
	$(1,0,0)$	$(1,1,0)^+$	
	$(0,0,1)$	$(1,1,1)$	
	$(2,0,0)$	$(0,2,0)$	
	$(0,0,2)$	$(2,1,1)^+$	
		$(2,2,0)^+$	
		$(2,2,1)^+$	

Table 5.1: System state table for Example 5.2

In the table, the "$(\mathbf{x})^+$" sign represents all the permutations of the elements of the component state vector $\mathbf{x}$. For example, system state level 1 can result from component states (1, 1, 0) and their permutations, namely, (0, 1, 1) and (1, 0, 1).

In terms of the definition of a multi-state consecutive k-out-of-n:F system we have provided, the system in this example is below state 2 if and only if at least one (consecutive) component is below state 2 ($k_2 = 1$). The system is below state 1 if and only if at least two consecutive components are below state 1 ($k_1 = 2$). We can see that in this example, $k_1 > k_2$ indicating a strictly decreasing multi-state consecutive k-out-of-n:F system. We can say that the system in this example has a consecutive 1-out-of-3:F structure at system state level 2 and a consecutive 2-out-of-3:F structure at system state level 1.

Example 5.3: An increasing consecutive k-out-of-n:F system

Consider a three-component system with $k_1 = 1$, $k_2 = 2$, and $k_3 = 3$. Both the system and the components may be in one of four possible states, namely, 0, 1, 2, and 3. The following table illustrates the relationship between system state and component states.

$\phi(\mathbf{x})$:	0	1	2	3
	$(0,0,0)$	$(1,1,1)$	$(0,2,0)$	$(3,0,0)^+$
	$(1,0,0)^+$	$(2,1,1)$	$(1,2,0)$	$(3,1,0)^+$
	$(1,1,0)^+$	$(1,1,2)$	$(0,2,1)$	$(3,1,1)^+$
	$(2,0,0)$		$(2,2,0)^+$	$(3,2,0)^+$
$\mathbf{x}$:	$(0,0,2)$		$(2,2,1)^+$	$(3,2,1)^+$
	$(2,1,0)$		$(2,2,2)$	$(3,2,2)^+$
	$(2,0,1)$			$(3,3,0)^+$
	$(1,0,2)$			$(3,3,1)^+$
	$(0,1,2)$			$(3,3,2)^+$
				$(3,3,3)$

Table 5.2: System state table for Example 5.3

Again, the "$(\mathbf{x})^+$" sign in the table represents all permutations of the elements of the component state vector $\mathbf{x}$. The system is below state 3 if and only if at least three (consecutive) components are below state 3. The system is below state 2 if and only if at least two consecutive components are below state 2 *and* at least three (consecutive) components are below state 3. The system is below state 1 if and only if at least one (consecutive) component is below state 1, at least two consecutive components are below state 2, *and* at least three (consecutive) components are below state 3. The system in this example has a parallel structure at system state 3 (3-out-of-3:F), a consecutive 2-out-of-3:F structure at system state level 2, and a series structure at system state level 1 (1-out-of-3:F).

Example 5.4: A constant multi-state consecutive k-out-of-n:F system

Consider a 3-component system with $k = 2$ wherein both the system and the components may be in one of three possible states, 0, 1, and 2. It satisfies the following relationship between system state and the states of the components:

$\phi(\mathbf{x})$:	0	1	2
	$(0,0,0)$	$(0,1,0)$	$(0,2,0)$
	$(1,0,0)$	$(1,1,0)^+$	$(2,2,1)^+$
$\mathbf{x}$:	$(1,1,1)$	$(2,1,0)$	$(1,2,1)$
	$(2,0,0)$	$(2,1,1)$	$(2,2,0)^+$
	$(0,0,2)$	$(1,1,2)$	$(2,2,2)$

Table 5.3: System state table for Example 5.4

Again, the "$(\mathbf{x})^+$" sign represents the permutations of the elements of the component state vector $\mathbf{x}$. Because $k_1 = k_2 = k_3 = 2$, the system is below state j whenever at least 2 consecutive components are below state j for $j = 1, 2$. The system has a consecutive 2-out-of-3:F structure at each of the system states 1 and 2.

5.3 Evaluation of system state distribution

In this section, we discuss the algorithms for performance evaluation of the increasing systems and the decreasing systems separately.

Case I: Decreasing multi-state consecutive k-out-of-n:F systems, i.e., $k_1 \geq k_2 \geq \cdots \geq k_M$:

In this case, the definition of a multi-state consecutive k-out-of-n:F system is equivalent to that $\phi(\mathbf{x}) < j$ if and only if at least k_j consecutive components have

$x_i < j$ for $j = 1, 2, \ldots, M$. Those components with $x_i < j$ are considered "failed" with respect to state level j. The system is considered "failed" with respect to state level j if $\phi(\mathbf{x}) < j$. The corresponding binary algorithms can be used to find $\Pr(\phi < j)$ in a multi-state consecutive k-out-of-n:F system. Once $\Pr(\phi < j)$ is found for $j = 1, 2, \ldots, M$, we can easily find $\Pr(\phi = j)$ for $j = 0, 1, \ldots, M$. The following formula is similar to Equation (2.17) (Hwang, 1982):

$$F_j(n; k_j) = F_j(n-1; k_j) + (1 - F_j(n - k_j - 1; k_j))\, P_{(n-k_j)j} \prod_{m=n-k_j+1}^{n} Q_{mj}, \tag{5.1}$$

where $F_j(a; b)$ is the probability that at least b consecutive components are below state j in an a component system and j may take values from 1 to M. Equation (5.1) can be applied recursively with the following boundary conditions:

$$F_j(a; b) = 0, \quad \text{for} \quad b > a > 0, \quad j = 1, 2, \ldots, M \tag{5.2}$$

$$P_{0j} = 1, \quad j = 1, 2, \ldots, M \tag{5.3}$$

By recursively applying Equation (5.1), we can find the probabilities that the system is below state j for $j = 1, 2, \ldots, M$ and then find the probabilities that the system is in state j for $j = 0, 1, \ldots, M$ using the following equations:

$$\Pr(\phi < j) = F_j(n; k_j), \qquad \text{for } j = 1, 2, \ldots, M, \tag{5.4}$$

$$\Pr(\phi = 0) = \Pr(\phi < 1), \tag{5.5}$$

$$\Pr(\phi = M) = 1 - \Pr(\phi < M), \tag{5.6}$$

$$\Pr(\phi = j) = \Pr(\phi < j+1) - \Pr(\phi < j), \qquad \text{for } j = 1, \ldots, M-1. \tag{5.7}$$

Note: The constant consecutive k-out-of-n:F systems can be treated as a special case of the decreasing consecutive k-out-of-n:F systems with $k_1 = k_2 = \cdots = k_M$.

Example 5.5: Evaluation of performance distribution of a decreasing system

Consider Example 5.2 wherein $n = 3$ and $M = 2$. Assume that all the components are i.i.d. with the following common component state distribution:

$$p_0 = 0.1, \quad p_1 = 0.4, \quad p_2 = 0.5.$$

Calculate the system state distribution.

Solution:

$$Q_1 = p_0 = 0.1, \quad Q_2 = p_0 + p_1 = 0.5$$

$$P_1 = p_1 + p_2 = 0.9, \quad P_2 = p_2 = 0.5$$

For $j = 2$, we have $k_2 = 1$. Using Equation (5.1), we have:

$$\begin{aligned} F_2(3;1) &= F_2(2;1) + (1 - F_2(1;1))P_{22}Q_{32} = Q_{12} + P_{12}Q_{22} + P_{12}P_{22}Q_{32} \\ &= 0.5 + 0.5 \times 0.5 + 0.5 \times 0.5 \times 0.5 = 0.875 \end{aligned}$$

For $j = 1$, we have $k_1 = 2$. Again using Equation (5.1), we have:

$$\begin{aligned} F_1(3;2) &= F_1(2;2) + (1 - F_1(1;1))Q_{21}Q_{31} = Q_{11}Q_{21} + P_{11}Q_{21}Q_{31} \\ &= 0.1 \times 0.1 + 0.9 \times 0.1 \times 0.1 = 0.019 \end{aligned}$$

The system state distributions can be calculated as follows:

$$\Pr(\phi < 2) = F_2(3;1) = 0.875$$

$$\Pr(\phi < 1) = F_1(3;2) = 0.019$$

$$\Pr(\phi = 2) = 1 - \Pr(\phi < 2) = 1 - 0.875 = 0.125$$

$$\Pr(\phi = 1) = \Pr(\phi < 2) - \Pr(\phi < 1) = 0.875 - 0.019 = 0.856$$

$$\Pr(\phi = 0) = F_1(3;2) = 0.019$$

Case II: Increasing multi-state consecutive k-out-of-n:F systems, i.e., $k_1 \leq k_2 \leq \cdots \leq k_M$

We assume that at least one of the inequalities in this case is a strict inequality. Based on the definition of minimal paths of multi-state systems, the minimal path sets to level j in an increasing consecutive k-out-of-n:F system could also be a minimal path to level $j + 1$ or even higher levels. Consider the system structure given in Example 5.3. One of the minimal path sets to system level 1 is (0, 2, 0) because $\phi(0, 2, 0) \geq 1$ and $\phi(\mathbf{x}) < 1$ for all $\mathbf{x} < (0, 2, 0)$. At the same time, component state vector (0, 2, 0) is also a minimal path set for system state level 2. The same as the decreasing k-out-of-n:G system, the "domination phenomenon" does not exist in increasing consecutive k-out-of-n:F systems. As a result, we are unable to extend the binary formula for evaluation of system state distribution under Case II.

Alternatively, bounding techniques may be used to establish bounds for the probability that the system will be in each possible state. Given the minimal path sets or minimal cut sets of a multi-state system, either IE or SDP method can be used to establish the bounds (El-Neweihi et al. 1978). In the following, we present a new technique for bounding system state distribution in Case II.

5.4 Bounding system state distribution of the increasing systems

As we have discussed in Chapter2, there exists a duality relationship between consecutive k-out-of-n:F and consecutive k-out-of-n:G systems. The consecutive-k-out-of-n:F and G systems are duals of each other (Kuo et al. 1990). Equation (2.20) indicates that we can obtain the unreliability of a consecutive k-out-of-n:F system through the reliability of a consecutive k-out-of-n:G system or vice versa. Naturally, we are wondering if it is possible similarly to use the dual conversion method to find the system state distribution of the increasing consecutive k-out-of-n:F systems.

First, we investigate the relationship between the performance distributions of a multi-state system and its dual system. The dual definition of multi-state systems is as:

$$\phi^D(\mathbf{x}) = M - \phi(\mathbf{M} - \mathbf{x}), \tag{5.8}$$

where $\mathbf{M} = (M, M, \ldots, M)$. As a result, we have

$$\Pr(\phi^D(\mathbf{x}) = j) = \Pr(\phi(\mathbf{M} - \mathbf{x}) = M - j). \tag{5.9}$$

Based on Equation (5.9), if we let $p_{ij} = p^D_{i(M-j)}$ for $i = 1, 2, \ldots, n$ and $j = 0, 1, \ldots, M$, then the probability for the primal system to be in state j is equal to the probability for the dual system to be in state $M - j$ for $j = 0, 1, \ldots, M$. We call such a method a dual conversion method.

Example 5.6: Relationship between a primal system and its dual system

Consider the system structure given in Example 5.3 as the primal system, which is an increasing consecutive-k-out-of-n:F system. By applying Equation (5.8) to each component state vector in Example 5.3, we obtain the structure function of its dual system as shown in the following table:

Again, the "$(\mathbf{x})^+$" sign in the table represents all permutations of the elements of the component state vector $\mathbf{x}$. By examining the structure function in this table, we find that the dual system becomes a decreasing multi-state consecutive-k-out-of-n:G system. The system is at state 1 or above if at least three consecutive components are at state 1 or above. The system is at state 2 or above if at least two consecutive

$\phi^D(\mathbf{x})$:	0	1	2	3
	$(0,0,0)$	$(1,1,1)$	$(2,2,1)$	$(3,2,1)$
	$(1,0,0)^+$	$(2,1,1)^+$	$(1,2,2)$	$(2,3,1)$
	$(1,1,0)^+$	$(2,1,2)$	$(2,2,2)$	$(1,3,2)$
	$(2,0,0)^+$	$(3,1,1)^+$		$(1,2,3)$
	$(0,0,2)$	$(3,1,3)$		$(3,2,2)^+$
$\mathbf{x}$:	$(2,1,0)^+$	$(2,1,3)$		$(3,3,1)$
	$(2,2,0)^+$	$(3,1,2)$		$(1,3,3)$
	$(3,0,0)^+$			$(3,3,2)^+$
	$(3,1,0)^+$			$(3,3,3)$
	$(3,2,0)^+$			
	$(3,3,0)^+$			

Table 5.4: System state table for Example 5.6

components are at state 2 or above *and* at least three components are at state 1 or above. The system is at state 3 (or above) if at least one component is at state 3 (or above) *and* at least two consecutive components are at state 2 or above *and* at least three components are at state 1 or above. We notice that $k_1 = 3$, $k_2 = 2$, and $k_3 = 1$ and we have to use the *"and"* relationship to specify the conditions that should be satisfied for the system to be at a certain state or above.

We now formalize the definition of a multi-state consecutive-k-out-of-n:G system as follows.

Definition 1 $\phi(\mathbf{x}) \geq j$ $(j = 1, 2, \cdots, M)$ *if at least* k_l *consecutive components are in state* l *or above for all* l $(1 \leq l \leq j)$. *A system with such a structure function is called a multi-state consecutive-k-out-of-n:G system.*

The condition in this definition can also be phrased as follows: $\phi(\mathbf{x}) \geq j$ $(j =$

$1, 2, \cdots, M$) if at least k_j consecutive components are in state j or above; and at least k_{j-1} consecutive components are in state $j-1$ or above; and $\cdots$; and at least k_1 consecutive components are in state 1 or above.

As we have seen from Example 5.3 and 5.6, the dual of an increasing consecutive-k-out-of-n:F system becomes a decreasing consecutive-k-out-of-n:G system. In general, we have the following proposition:

Proposition 1: The dual of an increasing consecutive k-out-of-n:F system is a decreasing consecutive-k-out-of-n:G system.

Proof:

Assume that ϕ is the structure function of an increasing consecutive k-out-of-n:F system and $\mathbf{k} = (k_1, k_2, \cdots, k_M)$. We have: $k_1 \leq k_2 \leq \cdots \leq k_M$. Let A_j denote that at least k_j components are in states below j for $j = 1, 2, \cdots, M$. The system is equivalent to:

- $\phi(\mathbf{x}) < 1$ if $\mathbf{x} \in A_1 \cap A_2 \cap \cdots \cap A_M$;
- $\phi(\mathbf{x}) < 2$ if $\mathbf{x} \in A_2 \cap \cdots \cap A_M$;
- $\vdots$
- $\phi(\mathbf{x}) < M$ if $\mathbf{x} \in A_M$.

Let B_j denote that at least consecutive k_j^D components in the daul structure ϕ^D are in state j or above.

We examine system level $1 (j = 1)$ first. Consider a $\mathbf{x}$ with at least consecutive k_1 components in states below $1 (< 1)$, consecutive k_2 components in states below $2 (< 2), \cdots$, and consecutive k_M components in states below $M (< M)$, then in $\mathbf{y} = \mathbf{M} - \mathbf{x}$, there are at least consecutive k_M components in state 1 or above, consecutive k_{M-1} components in state 2 or above, $\cdots$, and consecutive k_1 components in state M. Letting

$\mathbf{k^D} = (k_1^D, k_2^D, \cdots, k_M^D) = (k_M, k_{M-1}, \cdots, k_1)$. We have: If a $\mathbf{x} \in A_1 \cap A_2 \cap \cdots \cap A_M$, then $\mathbf{y} \in B_1 \cap B_2 \cap \cdots \cap B_M$. By the definition $\phi^D(\mathbf{M} - \mathbf{x}) = M - \phi(\mathbf{x})$ and since $\phi(\mathbf{x}) < 1$, we have: $\phi^D(\mathbf{M} - \mathbf{x}) = \phi^D(\mathbf{y}) = M$.

Generally, at system level j, If a $\mathbf{x} \in A_j \cap A_{j+1} \cap \cdots \cap A_M$, then $\phi(\mathbf{x}) < j$. For $\mathbf{y} = \mathbf{M} - \mathbf{x}$, $\mathbf{y} \in B_1 \cap B_2 \cap \cdots \cap B_{M-j}$. And then $\phi^D(\mathbf{y}) = M - \phi(\mathbf{x}) \geq M - j$ for $j = 1, 2, \cdots, M$. Since $\mathbf{k^D} = (k_1^D, k_2^D, \cdots, k_M^D) = (k_M, k_{M-1}, \cdots, k_1)$. ϕ^D is a decreasing consecutive k-out-of-n:G system. We can interprete a decreasing consecutive k-out-of-n:G system as: $\phi^D(\mathbf{y}) \geq j$ if $\mathbf{y} \in B_1 \cap B_2 \cap \cdots \cap B_j$. (END)

Based on the definition, we know that a decreasing multi-state consecutive-k-out-of-n:G system has the following properties:

1. $n \equiv k_0 \geq k_1 \geq k_2 \geq \ldots \geq k_M$,

2. The minimal paths to system level j will cause the system to be exactly in level j,

3. One of the minimal paths to level j is in the following form:

$$\underbrace{\underbrace{\underbrace{\underbrace{(j, \ldots, j}_{k_j}, j-1, \ldots, j-1}_{k_{j-1}}, j-2, \ldots, 1, \ldots, 1}_{k_1}, 0, \ldots, 0)}_{n}, \tag{5.10}$$

where the number of elements taking the value of i is equal to $k_i - k_{i+1} \geq 0$ for $i = 1, \ldots, j-1$.

4. Every minimal path to level j can be obtained by permutating the elements of this minimal path shown in (5.10).

Because the set of minimal path sets to level j contains some permutations of the vector shown in (5.10), it is easy to find all these minimal path sets to level j. For example, suppose that a system has 5 components and a minimal path set to level 3 is (3, 3, 2, 1, 0). From this minimal path set, we know that for the system to be at level 3 or above, at least two consecutive components must be in level 3 or above, and at least 3 consecutive components have to be in state 2 or above, and at least 4 consecutive components have to be in state 1 or above, and all 5 components have to be in state 0 or above. We also note that the last condition that all 5 components have to be in state 0 or above may be ignored because it is automatically satisfied. The following vectors will also satisfy these requirements and thus, they are also minimal paths to system level 3:

$$(2,3,3,1,0), \quad (1,3,3,2,0), \quad (1,2,3,3,0), \quad (0,1,2,3,3)$$
$$(0,1,3,3,2), \quad (0,2,3,3,1), \quad (0,3,3,2,1).$$

All these vectors can be obtained by permutating the elements of vector (3, 3, 2, 1, 0). However, not all permutations of the elements of vector (3, 3, 2, 1, 0) are minimal paths to system level 3.

Observing these minimal path vectors, we see that each may include more than two different values as its elements. In the example of vector (3, 3, 2, 1, 0), there are four different values, namely, 3, 2, 1, and 0. If the number of such different values in a minimal path vector to system level j is less than or equal to 2, we can use binary reliability evaluation algorithms to find the exact probability that the system is in state j or above. However, if this number of different values in the minimal path vector is greater than two, we can not use binary algorithms to evaluate the exact probability that the system is in state j or above. In the following, we present a method to bound $\Pr(\phi \geq j)$ when this happens.

Suppose we have a minimal path vector for system state level j, denoted by $\mathbf{y}$, which is in the form shown in Equation (5.10). We will use $\mathbf{y}_j^*$ to represent all minimal path vectors to system state level j. Then, we have

$$\Pr(\phi \geq j) = \Pr(\mathbf{x} \geq \mathbf{y}_j^*), \tag{5.11}$$

where $\mathbf{x}$ represents all possible component state vectors. If $\mathbf{y}$ has no more than two different element values, we can evaluate $\Pr(\mathbf{x} \geq \mathbf{y}_j^*)$ directly, to be illustrated later. If $\mathbf{y}$ has more than two different element values, define

$$s = Min\{i | i \in \mathbf{y}, i < j\}, \tag{5.12}$$

$$t = Max\{i | i \in \mathbf{y}, i < j\}. \tag{5.13}$$

Note that we have $0 \leq s \leq t < j$. Vectors $\mathbf{L}$ and $\mathbf{U}$ both with dimension n are defined as

$$\mathbf{L} = \underbrace{(\underbrace{j, \cdots, j}_{k_j}, s, \cdots, s)}_{n}, \tag{5.14}$$

$$\mathbf{U} = \underbrace{(\underbrace{j, \cdots, j}_{k_j}, t, \cdots, t)}_{n}. \tag{5.15}$$

Obviously, we have

$$\mathbf{L} \leq \mathbf{y} \leq \mathbf{U}. \tag{5.16}$$

Let $\mathbf{L}_j^*$ represent all component state vectors in which exactly k_j consecutive elements have a value of j and all other elements have a value of s. Let $\mathbf{U}_j^*$ represent all component state vectors in which exactly k_j consecutive elements have a value of j and all other elements have a value of t. Then, we have

$$\Pr(\mathbf{x} \geq \mathbf{U}_j^*) \leq \Pr(\phi \geq j) = \Pr(\mathbf{x} \geq \mathbf{y}_j^*) \leq \Pr(\mathbf{x} \geq \mathbf{L}_j^*) \tag{5.17}$$

Inequality (5.17) can be used to find upper and lower bounds on the probability that the system is in state j or above. When $s = t$, the two bounds are the same and equal to the exact value of $\Pr(\phi \geq j)$. The question remaining to be answered is how to evaluate the two bounds in the inequality (5.17).

Based on the definition of a decreasing multi-state consecutive-k-out-of-n:G system, event $\{\mathbf{x} \geq \mathbf{U}_j^*\}$ represents that at least k_j consecutive components are in state j or above and at the same time, all components must be in state t or above in an n-component system. In other words, this event represents that at least k_j consecutive components are in state j or above and all other components must be in states t or above but below state j. Similarly, event $\{\mathbf{x} \geq \mathbf{L}_j^*\}$ represents that at least k_j components must be in state j or above and all other components must be in state s or above but below j. The question to be answered is then how to find the probability that at least k_j components are in state j or above and all other components are in a lower state s (or t) or above but below j. In the following, we discuss the two cases when the components are i.i.d. and non-i.i.d. separately.

Case I: The components are i.i.d.:

The number of possible ways to have exactly k consecutive components at a certain state or above in an n-component system is equal to $(n-k+1)$. When all components are i.i.d., the following formula can be used to calculate $\Pr(\mathbf{x} \geq \mathbf{U}_j^*)$ and $\Pr(\mathbf{x} \geq \mathbf{L}_j^*)$.

$$\text{Lower Bound} = \Pr(\mathbf{x} \geq \mathbf{U}_j^*) = \sum_{k=k_j}^{n} (n-k+1) P_j^k (1 - P_j - Q_t)^{n-k}, \qquad (5.18)$$

$$\text{Upper Bound} = \Pr(\mathbf{x} \geq \mathbf{L}_j^*) = \sum_{k=k_j}^{n} (n-k+1) P_j^k (1 - P_j - Q_s)^{n-k}, \qquad (5.19)$$

We now use the following example to illustrate how we find or bound $\Pr(\phi \geq j)$ for an increasing consecutive-k-out-of-n:F system.

<u>Example 5.7: Evaluating system state distribution of an increasing system</u>

Consider the system given in Example 5.3. Assume that all components are i.i.d.

$$n = 3,\quad M = 3,\quad k_1 = 1,\quad k_2 = 2,\quad k_3 = 3,$$

$$p_0 = 0.1,\quad p_1 = 0.2,\quad p_2 = 0.3,\quad p_3 = 0.4.$$

The exact system state distribution has been calculated with the enumeration method:

$$\Pr(\phi = 0) = 0.049,\quad \Pr(\phi = 1) = 0.032,\quad \Pr(\phi = 2) = 0.135,\quad \Pr(\phi = 3) = 0.784.$$

The dual system structure of this system is a decreasing multi-state consecutive-k-out-of-n:G system as shown in Example 5.6. We now illustrate the use of this dual structure in evaluating the state distribution of the primal system. We will use superscript D to indicate the parameters of the dual system.

$$n = 3,\quad M = 3,\quad k_1^D = k_3 = 3,\quad k_2^D = k_2 = 2,\quad k_3^D = k_1 = 1,$$

$$p_0^D = p_3 = 0.4,\quad p_1^D = p_2 = 0.3,\quad p_2^D = p_1 = 0.2,\quad p_3^D = p_0 = 0.1.$$

$$P_1^D = p_1^D + p_2^D + p_3^D = 0.6,\quad P_2^D = p_2^D + p_3^D = 0.3,\quad P_3^D = p_3^D = 0.1.$$

$$Q_1^D = 1 - P_1^D = 0.4,\quad Q_2^D = 1 - P_2^D = 0.7,\quad Q_3^D = 1 - P_3^D = 0.9.$$

For system state level 1, the only minimal path set is $(1, 1, 1)$. Thus,

$$R_{s1}^D = \Pr(\phi^D \geq 1) = P_1^D \times P_1^D \times P_1^D = 0.6^3 = 0.216.$$

For system state level 2, the minimal paths are $(2, 2, 1)$ and $(1, 2, 2)$. These minimal paths have only two different element values.

$$\begin{aligned} R_{s2}^D &= \sum_{k=k_2^D}^{n} (n - k + 1)(P_2^D)^k (1 - P_2^D - Q_1^D)^{n-k} \\ &= \sum_{k=2}^{3} (3 - k + 1) 0.3^k (1 - 0.3 - 0.4)^{3-k} = 2 \times 0.3^2 \times 0.3 + 0.3^3 = 0.081. \end{aligned}$$

For system state level 3, the minimal paths are (3,2,1), (2,3,1), (1,3,2), and (1,2,3). Each of these vectors have three different element values. By selecting $\mathbf{L} = (3, 1, 1)$ and $\mathbf{U} = (3, 2, 2)$, we have:

$$\begin{aligned} \text{Upper bound} &= \sum_{k=k_3^D}^{n} (n-k+1)(P_3^D)^k (p_1^D + p_2^D)^{n-k} \\ &= 3 \times 0.1 \times 0.5^2 + 2 \times 0.1^2 \times 0.5 + 0.1^3 = 0.086, \\ \text{Lower Bound} &= \sum_{k=k_3^D}^{n} (n-k+1)(P_3^D)^k (p_2^D)^{n-k} \\ &= 3 \times 0.1 \times 0.2^2 + 2 \times 0.1^2 \times 0.2 + 0.1^3 = 0.017 \end{aligned}$$

Now we have:

$$0.017 \leq R_{s3}^D \leq 0.086$$

while exactly $R_{s3}^D = 0.049$ calculated with an enumeration method. We can now express the probability for the dual system to be in each state as follows:

$$r_{s0}^D = 1 - R_{s1}^D = 1 - 0.216 = 0.784$$

$$r_{s1}^D = R_{s1}^D - R_{s2}^D = 0.216 - 0.081 = 0.135$$

$$r_{s2}^D = R_{s2}^D - R_{s3}^D = 0.081 - R_{s3}^D$$

$$-0.05 \leq r_{s2}^D \leq 0.064 \quad \text{or} \quad 0 \leq r_{s2}^D \leq 0.064$$

$$0.017 \leq r_{s3}^D = R_{s3}^D \leq 0.086$$

The primal system state distributions can be calculated as follows based on Equation (5.9):

$$r_{s3} = r_{s0}^D = 0.784$$

$$r_{s2} = r_{s1}^D = 0.135$$

$$0 \leq r_{s1} = r_{s2}^D \leq 0.064$$

$$0.017 \leq r_{s0} = r_{s3}^D \leq 0.086$$

Case II: The components are non-i.i.d:

When the components in a system are non-i.i.d., Equation (2.17) is used for reliability evaluation of binary consecutive k-out-of-n:F systems. To avoid notation confusion with what has been defined in this chapter, we will use u_i and v_i to represent the working and failure probabilities of component i in the binary system for $i = 1, 2, \ldots, n$. Equation(2.17) is for binary systems and require the following condition to be satisfied for every component i:

$$u_i + v_i = 1, \quad i = 1, 2, \ldots, n. \tag{5.20}$$

To enable our analysis of multi-state consecutive-k-out-of-n:G system, we now propose to relax the requirement shown in Equation (5.20) into the following:

$$0 \leq u_i + v_i \leq 1, \quad i = 1, 2, \ldots, n. \tag{5.21}$$

We will still call u_i the "working" probability of component i and q_i the "failure" probability of component i. Since $u_i + v_i$ may be less than 1, a component may be in a state other than "working" or "failed". Under this relaxed condition, we provide the following equation for calculating the probability that at least k consecutive components are "working" and all other components are "failed".

$$R(n;k) = v_n R(n-1;k) + u_n \left(\left(\prod_{j=n-k+1}^{n-1} u_j \right) R^*(n-k) + \sum_{i=n-k+1}^{n-1} v_i \left(\prod_{j=i+1}^{n-1} u_j \right) R(i-1;k) \right), \tag{5.22}$$

where $R(n;k)$ is the probability that at least k consecutive components in the system of n components are "working" while all other components are "failed" and $R^*(i) \equiv \prod_{j=1}^{i}(u_j + v_j)$ for $i \geq 1$.

The derivation of Equation (5.22) is based on the Bayes Theorem. The first term $v_n R(n-1;k)$ is the probability that component n is "failed" and at the same time at least k consecutive components out of the remaining $(n-1)$ components are "working". The second term $u_n(\bullet)$ represents the probability that component n is "working" and the system is "working". In the brackets, the first term represents the case that components $(n-1)$, $(n-2)$, $\cdots$, and $(n-k+1)$ are all "working" and the remaining $n-k$ components are either "working" or "failed" (but with a probability of $u_i + v_i < 1$). The second term in the bracket represents the probability that component i $(n-k+1 \le i \le n-1)$ is the first "failed" component counting from component $n-1$ downward and the $i-1$ component subsystem including components 1 through $i-1$ is "working".

Equation (5.22) is a recursive formula. The following boundary conditions are needed.

$$R(k;k) = u_1 u_2 \cdots u_k, \tag{5.23}$$

$$R(a;b) \;=\; 0, \quad \text{for} \quad b > a > 0 \tag{5.24}$$

Equation (5.22) with boundary conditions (5.23) and (5.24) can be used to find the upper bounds and lower bounds for $\Pr(\phi \ge j)$ for $j = 1, 2, \ldots, M$ in a multi-state consecutive-k-out-of-n:G system, as outlined below.

When one is considering system state level j, define the following:

$$u_i = P_{ij}, \quad i = 1, 2, \ldots, n.$$

For upper bound calculation, define

$$v_i = 1 - u_i - Q_{is}, \quad i = 1, 2, \ldots, n, \tag{5.25}$$

where s is given in equation (5.12). Equation (5.22) can then be used to derive the lower bound for $\Pr(\phi \ge j)$. For lower bound calculation, define

$$v_i = 1 - u_i - Q_{it}, \quad i = 1, 2, \ldots, n, \tag{5.26}$$

where t is given in equation (5.13). Equation (5.22) can then be used to calculate the upper bound for $\Pr(\phi \geq j)$. The following example illustrate this procedure.

Example 5.8: Evaluating system state distribution of an increasing system

Evaluate the state distribution of the system given in Example 5.3 assuming the component state distributions are non-i.i.d. as follows:

State	Component 1	Component 2	Component 3
	Its Probability in State		
0	0.1	0.2	0.1
1	0.2	0.2	0.3
2	0.3	0.3	0.1
3	0.4	0.3	0.5

The component state distribution in the dual system is:

$$p_{10}^D = p_{13} = 0.4, \quad p_{11}^D = p_{12} = 0.3, \quad p_{12}^D = p_{11} = 0.2, \quad p_{13}^D = p_{10} = 0.1.$$

$$p_{20}^D = p_{23} = 0.3, \quad p_{21}^D = p_{22} = 0.3, \quad p_{22}^D = p_{21} = 0.2, \quad p_{23}^D = p_{20} = 0.2.$$

$$p_{30}^D = p_{33} = 0.5, \quad p_{31}^D = p_{32} = 0.1, \quad p_{32}^D = p_{31} = 0.3, \quad p_{33}^D = p_{30} = 0.1.$$

$$P_{11}^D = 0.6, \quad P_{12}^D = 0.3, \quad P_{13}^D = 0.1.$$

$$P_{21}^D = 0.7, \quad P_{22}^D = 0.4, \quad P_{23}^D = 0.2.$$

$$P_{31}^D = 0.5, \quad P_{32}^D = 0.4, \quad P_{33}^D = 0.1.$$

The exact state distribution of the dual system has been calculated with the enumeration method:

$$\Pr(\phi^D = 0) = 0.79, \quad \Pr(\phi^D = 1) = 0.102, \quad \Pr(\phi^D = 2) = 0.034, \quad \Pr(\phi^D = 3) = 0.074.$$

For system state level 1, let $u_i = P_{i1}^D$ and $v_i = p_{i0}^D$. Using Equation (5.22):

$$R_{s1}^D = R(3;3) = P_{11}^D \times P_{21}^D \times P_{31}^D = 0.6 \times 0.7 \times 0.5 = 0.21.$$

For system state level 2, let $u_i = P_{i2}^D$ and $v_i = p_{i1}^D$. Using Equation (5.22):

$$\begin{aligned} R_{s2}^D &= R(3;2) = p_{31}^D R(2;2) + P_{32}^D(P_{22}^D R^*(1) + p_{21}^D R(1;2)) \\ &= p_{31}^D P_{22}^D P_{12}^D + P_{32}^D P_{22}^D (p_{11}^D + P_{12}^D) \\ &= 0.1 \times 0.4 \times 0.3 + 0.4 \times 0.4(0.3 + 0.3) = 0.108. \end{aligned}$$

For system state level 3, selecting $\mathbf{L} = (3,1,1)$ and $\mathbf{U} = (3,2,2)$ and $k = 2$. For $\mathbf{L} = (3,1,1)$, let $u_i = P_{i3}^D$ and $v_i = p_{i1}^D + p_{i2}^D$. For $\mathbf{U} = (3,2,2)$, let $u_i = P_{i3}^D$ and $v_i = p_{i2}^D$. Using Equation (5.22), we have:

$$\begin{aligned} \text{Upper bound} &= R(3;1) = (p_{31}^D + p_{32}^D)R(2;1) + P_{33}^D R^*(2) \\ &= (p_{31}^D + p_{32}^D)((p_{21}^D + p_{22}^D)R(1;1) + P_{23}^D R^*(1)) + P_{33}^D R^*(2) \\ &= 0.4 \times (0.5 \times 0.1 + 0.2 \times 0.6) + 0.1 \times 0.6 \times 0.7 = 0.11 \\ \text{Lower Bound} &= R(3;1) = p_{32}^D R(2;1) + P_{33}^D R^*(2) \\ &= p_{32}^D(p_{22}^D R(1;1) + P_{23}^D R^*(1)) + P_{33}^D R^*(2) \\ &= 0.3 \times (0.2 \times 0.1 + 0.2 \times 0.3) + 0.1 \times 0.3 \times 0.4 = 0.036 \end{aligned}$$

Now we have:

$$0.036 \leq R_{s3}^D \leq 0.11.$$

With the calculated R_{s1}^D and R_{s2}^D values and the bounds on R_{s3}^D, we can find the following for the primal system following the same approach as shown in Example 8:

$$0.036 \leq F_{s1} = R_{s3}^D \leq 0.11,$$

$$F_{s2} = R_{s2}^D = 0.108,$$

$$F_{s3} = R_{s1}^D = 0.21.$$

5.5 Concluding remarks

In this chapter, we propose a definition of the multi-state consecutive k-out-of-n:F system. This model contributes towards the analysis of multi-state k-out-of-n structures. Similar to the generalized multi-state k-out-of-n system, the binary algorithm can be extended to the decreasing consecutive k-out-of-n:F system for system performance distribution evaluation. For the increasing consecutive k-out-of-n:F system, no efficient algorithm has been available for system performance evaluation yet. Alternatively, we present a new technique for establishing the bounds of the system performance distribution. We also define the multi-state consecutive k-out-of-n:G system. The duality between multi-state consecutive k-out-of-n:F and G systems is investigated. The dual conversion method can be used for bounding the performance distribution of the increasing consecutive k-out-of-n:F system.

The issues that should be further investigated on multi-state consecutive k-out-of-n systems include:

1. Improve the bounds of the increasing consecutive k-out-of-n:F systems and develop an index to measure the bound precision.

2. Find an efficient algorithm for system performance evaluation of the increasing consecutive k-out-of-n:F systems or the decreasing consecutive k-out-of-n:G systems.

Chapter 6

Dominant Multi-state Systems

6.1 Introduction

In Chapter 4 and 5, we have found that the algorithms for system state distribution evaluation of multi-state increasing k-out-of-n:G systems and decreasing consecutive k-out-of-n:F systems are so simple. Both of them can be treated like the corresponding binary systems. We also noticed the "domination" phenomenon exists in these two types of the systems. However, the algorithms for multi-state decreasing k-out-of-n:G systems and increasing consecutive k-out-of-n:F systems are complicated or even difficult to develop. In both of them, the "domination" phenomenon does not exist. The systems can not be treated like binary systems. This raises a question: under what conditions, a multi-state system can be treated like a binary system? In the dichotomous opinion, a multi-state coherent system and its components could be divided into two groups: $\phi \geq j$ and $\phi < j$ or $x_i \geq j$ and $x_i < j$, at any level j, $j = 1, 2, \ldots, M$. However, because of different relevancy conditions that are used, it is not always possible to divide the states of the system and its components into two groups separated by the system state level j for system performance evaluation.

As we have reviewed in Chapter 3, the relationship between binary system structures and multi-state system structures has been explored by some researchers, for example, Block and Savits (1982) and Wood (1985). However, they mainly focused on binary decomposition of multi-state systems rather than the equivalent conditions between a multi-state system and a binary system.

In this chapter, we propose a definition of the dominant multi-state system. Without referred to component relevancy conditions, multi-state systems can be divided into two groups: dominant or non-dominant systems. In a dominant system, the system state and component state vectors can be divided into two separate groups at any given system level while a non-dominant system does not have such a property. Furthermore, dominant systems can be divided into two types: with binary image and without binary image. A dominant system with binary image can be treated as a binary system. The dominance property of multi-state systems possesses excellent potential for system performance evaluation and bound computation. Section 6.2 proposes a definition of the dominant system. In Section 6.3, we present a definition of the binary-imaged dominant system. In Section 6.4, the properties of dominant systems are discussed and an approach for the bound computation of dominant system state distribution is developed. Concluding remarks are provided in Section 6.5.

Notation:

x_{ij}	a binary indicator; $x_{ij} = \begin{cases} 1, & \text{if } x_i \geq j; \\ 0, & \text{otherwise.} \end{cases}$
$\mathbf{x}^j$	a binary indicator vector, $\mathbf{x}^j = (x_{1j}, x_{2j}, \cdots, x_{nj})$.
$\phi^j(\mathbf{x})$	a binary indicator; $\phi^j = \begin{cases} 1 & \text{if } \phi(\mathbf{x}) \geq j; \\ 0 & \text{otherwise.} \end{cases}$
ϕ^D	the dual structure function of ϕ
ψ	structure function of binary variables.
ψ^j	binary structure function for system state j.
P_{ij}	$\Pr(x_i \geq j)$.

P_{ij}^D	the corresponding P_{ij} of ϕ^D,$P_{ij}^D = P_{i(M-j)}$.
P_j	$P_j = P_{ij}$ when the components are i.i.d.
P_j^D	$P_j^D = P_{ij}^D$ when the components are i.i.d.
p_{ij}	$\Pr(x_i = j)$.
p_{ij}^D	the reliability of component i of ϕ^D, $p_{ij}^D = Pr(x_i = M - j)$.
p_j	$p_j = p_{ij}$ when the components are i.i.d.
$\mathbf{P}^j$	component state distribution vector at level j, $\mathbf{P}^j = (P_{1j}, P_{2j}, \cdots, P_{nj})$.
$h^j(\mathbf{P}^j)$	system reliability function at level j or above, $h = E(\phi^j(\mathbf{x}) = 1) = \Pr(\phi(\mathbf{x}) \geq j)$.
R_{sj}	$\Pr(\phi(\mathbf{x}) \geq j)$, $R_{sj} = h^j(\mathbf{P}^j)$.
r_{sj}	$\Pr(\phi(\mathbf{x}) = j)$

Other notations are the same as Chapter 4 and 5.

6.2 The definition of dominant multi-state systems

Much research work has been reported on the relevancy conditions of multi-state systems. However, the relevancy conditions are complicated and many different versions are proposed (Andrzejezak, 1992). Another approach by researchers for multi-state system reliability analysis is to investigate the mathematical properties of the structure functions rather than the underlying relevancy conditions. Generally speaking, we can divide multi-state systems into dominant systems and non-dominant systems depending on their system structure functions. Such a classification is independent of what relevancy conditions are used.

Definition 2.1: Two component state vectors $\mathbf{x}$ and $\mathbf{y}$ are said to be equivalent if and only if there exists a j such that $\phi(\mathbf{x}) = \phi(\mathbf{y}) = j$, $j = 0, 1, \cdots, M$. We use notation $\mathbf{x} \leftrightarrow \mathbf{y}$ to indicate that these two vectors are equivalent (Hudson and Kapur, 1983).

Property 2.1: If $\mathbf{x} \leftrightarrow \mathbf{y}$ and $\mathbf{y} \leftrightarrow \mathbf{z}$, then $\mathbf{x} \leftrightarrow \mathbf{z}$.

Equivalent state vectors have the transferable property. Obviously, $\mathbf{x} \leftrightarrow \mathbf{x}$. All component state vectors that result in the same system state level are equivalent to one another. We can say that these equivalent vectors are in the same class called equivalent class.

Definition 2.2: A multi-state coherent system is called a dominant system if and only if its structure function ϕ satisfies: $\phi(\mathbf{y}) > \phi(\mathbf{x})$ implies either (1) $\mathbf{y} > \mathbf{x}$ or (2) $\mathbf{y} > \mathbf{z}$ and $\mathbf{x} \leftrightarrow \mathbf{z}$.

The dominance condition states that if $\phi(\mathbf{y}) > \phi(\mathbf{x})$, then vector $\mathbf{y}$ must be bigger than a vector that is in the same equivalent class as vector $\mathbf{x}$. A vector is bigger than another vector if every element of the first vector is at least as big as the corresponding element in the second vector and at least one component in the first vector is bigger than the corresponding one in the second vector. For example, vector $(2, 2, 0)$ is bigger than $(2, 1, 0)$, but not bigger than $(0, 1, 1)$. We use the word "dominant" to indicate that vector $\mathbf{y}$ dominates vector $\mathbf{x}$ even though we may not necessarily have $\mathbf{y} > \mathbf{x}$. If a multi-state system does not satisfy Definition 2.2, we call it a non-dominant system.

Example 6.1:

Consider a two component coherent multi-state system. Both the system and its components may be in three possible states: 0, 1, and 2. Table 6.1 shows one structure function of such a system (called system A), which has a series structure at level 2 because the system is at level 2 if and only if both components are at level 2; and the system has a parallel structure at level 1 because the system is in state 1 or above if and only if at least one of the components is in state 1 or above. Based on Definition 2.2, we conclude that system A is a dominant system. Table 6.2 gives a different system structure of a two-component system (called system B). System B has a parallel structure at level 2 and a series one at level 1. Since vector (2,0) resulting in system state 2 is not bigger or equal to any vector resulting in system state 1 so system B is a non-dominant system.

$\phi(\mathbf{x})$:	0	1	2
	$(0,0)$	$(1,0)$	$(2,2)$
		$(0,1)$	
		$(1,1)$	
$\mathbf{x}$:		$(2,0)$	
		$(0,2)$	
		$(2,1)$	
		$(1,2)$	

Table 6.1: System A: a dominant system

Let ϕ and ϕ' be the structure functions of two different binary coherent systems with the same number of components. If $\phi(\mathbf{x}) \geq \phi'(\mathbf{x})$ holds for all $\mathbf{x}$, then we say that structure ϕ is stronger than structure ϕ'. Among binary systems with n components, the series structure is the weakest structure while the parallel structure is the strongest structure because $\prod_{i=1}^{n} x_i \leq \phi(\mathbf{x}) \leq 1 - \prod_{i=1}^{n}(1 - x_i)$ (Barlow and Proschan, 1975). In a dominant system, let ϕ^l and ϕ^j denote the binary structure functions for system levels l and j respectively, where $l > j$. For any component state

$\phi(\mathbf{x})$:	0	1	2
	(0,0)	(1,1)	(2,0)
$\mathbf{x}$:	(1,0)		(0,2)
	(0,1)		(2,1)
			(1,2)
			(2,2)

Table 6.2: System B: a non-dominant system

vector $\mathbf{x}$, it is always true that $\phi^l(\mathbf{x}) \leq \phi^j(\mathbf{x})$ based on Definition 2.2. Otherwise, if $\phi^l(\mathbf{x}) > \phi^j(\mathbf{x})$, then we will have: $\mathbf{x} > \mathbf{x}$, this is not true. As a result, we say that the structure of a dominant system changes from strong to weak as its system level increases.

With respect to any given system level j, the states of a multi-state system can be divided into two separate groups: "functioning" if $\phi(\mathbf{x}) \geq j$ and "failed" otherwise. Similarly, component i is said "functioning" when $x_i \geq j$ and "failed" otherwise. Since j may take different values, "functioning" and "failure" have different meanings for different j values. We can say that the meanings of "functioning" and "failure" are dynamic or context dependent. The dichotomy of the system and the components can be done at all state levels for a dominant system while a non-dominant system does not have this property. For instance, in Example 6.1, the minimal paths to level 1 of system A are $(1,0)$ and $(0,1)$, which result exactly in system level 1. The only minimal path to level is $(2,2)$, which also results exactly in system level 2. The system state and component state vectors can be divided into two groups at both levels, i.e., $\phi(\mathbf{x}) \geq 2$ if and only if $\mathbf{x} \geq (2,2)$, $\phi(\mathbf{x}) \geq 1$ if and only if $\mathbf{x} \geq (0,1)$ or $(1,0)$. However, the minimal paths to level 1 of system B are $(1,1)$, $(2,0)$ and $(0,2)$. As $\phi(2,0) = \phi(0,2) = 2$, the system state and component state vectors can not be divided into two separate groups at system level 1.

A key problem in the multi-state context is how to find the system performance probability distribution $\Pr(\phi(\mathbf{x}) \geq j)$ for all j. Although the component state vectors of a dominant system can be divided into two separate groups at any system state level j, the binary algorithm may not be able to applied on it sometimes. In the following, we will propose the definition of binary-imaged systems.

6.3 The definition of binary-imaged multi-state systems

In general, dominant systems can be divided into two types: with binary image and without binary image, where the meanings of "binary image" implies that the system can be treated as a binary system and thus binary reliability evaluation algorithm can be applied on it. As illustrated in the previous paragraph, a non-dominant system is not a binary-imaged system because its component state vectors can not be divided into two separate groups. For instance, system B in Example 6.1 is not a binary-imaged system since it is a non-dominant system, even it has a parallel structure at level 2 and a series structure at level 1. In other words, the dominance property has to be satisfied first for the system to have a binary image. Ansell and Bendell (1987) defined a multi-state coherent system with a binary image by:

$$\mathbf{x}^j = \mathbf{y}^j \Longrightarrow \phi^j(\mathbf{x}) = \phi^j(\mathbf{y}), \tag{6.1}$$

for any j. If a system satisfies (6.1), then it can be treated as a binary system. However, this condition is incomplete. Some systems can also be binary-imaged even they do not satisfy Equation (6.1). We propose a definition of the binary-imaged dominant system as follows:

Definition 2.3: A multi-state system is a binary-imaged system if and only if its structure indicator functions satisfies either: (1) $\mathbf{x}^j = \mathbf{y}^j \Longrightarrow \phi^j(\mathbf{x}) = \phi^j(\mathbf{y})$; or (2) $\mathbf{x}^j = \mathbf{z}^j$ and $\mathbf{z} \leftrightarrow \mathbf{y} \Longrightarrow \phi^j(\mathbf{x}) = \phi^j(\mathbf{y})$.

This definition implies that a binary-imaged system must be a dominant system. Consider a non-dominant system and assume level $j+1$ does not dominant level j. It means that at level $j+1$, there exists at least one component state vector $\mathbf{y}$ such that $\mathbf{y}$ is not bigger than any component state vectors at level j. Simply, we assume that $\mathbf{y}$ is one of the minimal paths to level $j+1$ such that $\mathbf{y} = (\underbrace{j+1, \cdots, j+1}_{n-m}, r_1, \cdots, r_m)$, where element $r_i \in \{0, 1, \cdots, j\}$ for $i = 1, \cdots, m$. Now construct a $\mathbf{x}$ such that $\mathbf{x} = (\underbrace{j+1, \cdots, j+1}_{n-m-1}, r_1, \cdots, r_m, j)$, i.e., replacing one element $j+1$ in $\mathbf{y}$ with j to obtain $\mathbf{x}$. We have: $\phi(\mathbf{x}) \neq j$ since ϕ is a non-dominant system. Regarding to level j, $\mathbf{x}^j = \mathbf{y}^j$ and $\phi^j(\mathbf{y}) = 1$, but $\phi^j(\mathbf{x}) = 0$. $\phi^j(\mathbf{y}) \neq \phi^j(\mathbf{x})$. This contradicts with Definition 2.3.

In Example 6.1, system A is a binary-imaged system based on Definition 2.3. However, system B is not a binary-imaged system. In system B, let $\mathbf{x} = (1,0)$ and $\mathbf{y} = (2,0)$, with regard to level $j = 1$, we have: $\mathbf{x}^j = \mathbf{y}^j = (1,0)$, but $\phi^j(\mathbf{x}) = 0 \neq \phi^j(\mathbf{y}) = 1$. Usually, a dominant system is not binary-imaged since the structure property of a multi-state is very complex. For example, a binary system with two-components have only two possible structures: either series or parallel. The two systems A and B in Example 6.1 contain only two structures out of 2 components in the multi-state context. There are many other structures that can be constructed out of 2 components in the multi-state context. Example 6.2 is used to illustrate this point.

Example 6.2:

Consider a two component coherent system. Both the system and its components may be in three possible states: 0, 1, and 2. Table 6.3 shows another structure function of such a system (called system C), which is neither series nor parallel at each level, however it is a dominant system. System D in Table 6.4 belongs to dominant system too, but its structures are neither series nor parallel. In this example, both systems C and D are dominant, but they are not binary-imaged systems.

$\phi(\mathbf{x}):$	0	1	2
	(0, 0)	(1, 0)	(2, 0)
	(0, 1)	(1, 1)	(2, 1)
$\mathbf{x}:$		(0, 2)	(2, 2)
		(1, 2)	

Table 6.3: System C: a dominant system

$\phi(\mathbf{x}):$	0	1	2
	(0, 0)	(1, 0)	(2, 1)
		(0, 1)	(1, 2)
$\mathbf{x}:$		(1, 1)	(2, 2)
		(2, 0)	
		(0, 2)	

Table 6.4: System D: a dominant system

An equivalent definition of the dominant system using the expression of binary structure function is as follows:

Definition 2.4: A multi-state system is a dominant if and only if its structure indicator functions satisfies: $\phi^j(\mathbf{x}) = \phi^j(\mathbf{y})$ implies either (1) $\mathbf{x}^j = \mathbf{y}^j$ or (2) $\mathbf{x}^j = \mathbf{z}^j$ and $\mathbf{z} \leftrightarrow \mathbf{y}$ for $j = 1, 2, \cdots, M$.

Usually, The structure function ϕ of a multi-state system can be expressed in terms of $n \times M$ binary variable as follows:

$$\phi(\mathbf{x}) = \psi(x_{10}, x_{11}, \cdots, x_{1M}, \cdots, x_{n1}, \cdots, x_{nM}) \tag{6.2}$$

For a binary-imaged system, we can express its structure function ϕ as the following:

$$\phi^j(\mathbf{x}) = \psi^j(x_{1j}, x_{2j}, \cdots, x_{nj}) \tag{6.3}$$

Equation (6.3) indicates that the structure function of a binary-imaged multi-state system is only related to n binary variables. Existing binary algorithms can be used to evaluate the probability that a binary-imaged system is at state j or above for any level j. For instance, in Example 6.1, the minimal paths with respect to level 1 are $(1, 0)$ and $(0, 1)$. The minimal paths with respect to level 2 is $(2, 2)$. We can write: $\phi^1(\mathbf{x}) = \psi^1(x_{11}, x_{21})$ and $\phi^2(\mathbf{x}) = \psi^j(x_{12}, x_{22})$. In a binary system, a system reliability function of n components can be expressed in terms of system reliability function of $(n-1)$ components as:

$$h(\mathbf{p}) = p_{i1} h(1_i; \mathbf{p}) + (1 - p_{i1}) h(0_i; \mathbf{p}) \tag{6.4}$$

For a binary-imaged multi-state system, similar principle can be used as follows:

$$h^j(\mathbf{P}^j) = P_{ij} h(1_i; \mathbf{P}^j) + (1 - P_{ij}) h(0_i; \mathbf{P}^j) \tag{6.5}$$

Once $h^j(\mathbf{P}^j) = \Pr(\phi \geq j)$ is calculated for all j values, we can find $\Pr(\phi = j)$ as follows.

$$\Pr(\phi = j) = h^j(\mathbf{P}^j) - h^{j+1}(\mathbf{P}^{j+1}) \tag{6.6}$$

In the following, we use an example to illustrate how to apply binary algorithm for the system performance evaluation of a binary-imaged multi-state system.

Example 6.3:

A multi-state coherent system consists of 3 i.i.d. components. Both the system and the components are allowed to have four possible states: 0, 1, 2, 3. Assume $p_0 = 0.1$, $p_1 = 0.3$, $p_2 = 0.4$, $p_3 = 0.2$. The structures of the system are shown as Fig. 6.1.

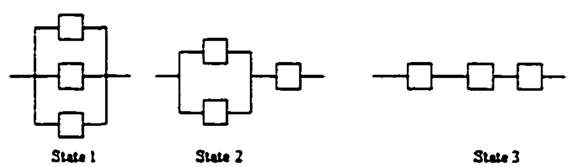

Figure 6.1: A dominant system with binary image for Example 6.3

Solution:

Obviously, the system has a parallel structure at level 1, a mixed parallel-series structure at level 2 and a series structure at level 3. The structures change from strong to weak as the level of the system increases. The minimal paths to level 1 are $(1,0,0)$ and its permutations; to level 2 are $(2,0,2)$ and $(0,2,2)$; and to level 3 is $(3,3,3)$. The system is a binary-imaged system. The following table illustrates the relationship between the system state and the component states.

In the table, the "+" sign represents all permutations of the states of the components listed above the "+" sign. To calculate the system reliabilities at all levels, we can extend binary algorithms:

$P_1 = 0.9, P_2 = 0.6, P_3 = 0.2.$

$R_{s3} = P_{13} \times P_{23} \times P_{33} = P_3^3 = 0.008.$

$R_{s2} = (1 - (1 - P_{12})(1 - P_{22})) \times P_{32} = 2 \times P_2^2 - P_2^3 = 0.504.$

$R_{s1} = 1 - (1 - P_{11})(1 - P_{21})(1 - P_{31}) = 1 - (1 - P_1)^3 = 0.999.$

$r_{s3} = R_{s3} = 0.008.$

$r_{s2} = R_{s2} - R_{s3} = 0.496.$

$r_{s1} = R_{s1} - R_{s2} = 0.495.$

$r_{s0} = 1 - R_{s1} = 0.001.$

As shown in Example 6.3, a commonly-used approach for multi-state system reliability evaluation is to extend the results from binary system reliability evaluation. The minimal path sets and minimal cut sets play important roles for the reliability evaluation of binary systems. In the following section, we discuss the properties of minimal path sets and minimal cut sets of dominant systems.

6.4 Properties of dominant systems

Usually, finding all minimal path sets or minimal cut sets of a multi-state system is difficult because some minimal path sets to level j are hidden in some higher system levels. This often happens in a non-dominant system. But for a dominant system, this will not happen.

Property 3.1: Let $P_1^j, P_2^j, \cdots, P_r^j$ be the minimal path sets to level j of a dominant systems, then $\phi(P_i^j) = j$ for $i = 1, 2, \cdots, r$.

Proof:

$\phi(\mathbf{x})$:	0	1	2	3
	(0,0,0)	(3,1,1)	(3,3,2)	(3,3,3)
		(3,1,0)	(3,2,2)	
		(3,0,0)	(2,2,2)	
		(2,1,1)	+	
		(2,1,0)	(3,1,3)	
		(2,0,0)	(1,3,3)	
		(1,1,1)	(3,0,3)	
		(1,1,0)	(0,3,3)	
$\mathbf{x}$:		(1,0,0)	(3,1,2)	
		+	(1,3,2)	
		(3,3,1)	(1,2,3)	
		(3,3,0)	(2,1,3)	
		(3,2,1)	(0,2,3)	
		(2,3,1)	(3,0,2)	
		(3,2,0)	(0,3,2)	
		(2,3,0)	(2,0,3)	
		(2,2,1)	(1,2,2)	
		(2,2,0)	(2,1,2)	
			(2,0,2)	
			(0,2,2)	

Table 6.5: System state table for Example 6.3

Assume that there exists an m ($1 \le m \le r$) such that $\phi(P^j_m) = l > j$. By the definition of the dominant system, we have:

If $\phi(P^j_m) = l > \phi(P^j_i) = j$ for $i = 1, \cdots, r$ and $i \neq m$, then either we have $P^j_m > P^j_i$ or there exists some $\mathbf{z} \leftrightarrow P^j_i$ such that $P^j_m > \mathbf{z}$. This contradicts that P^j_m is a minimal path set to level j. (END)

Property 3.2: Let $K^j_1, K^j_2, \cdots, K^j_s$ be the minimal cut sets to level j of a dominant system, then $\phi(K^j_i) = j - 1$ for $i = 1, 2, \cdots, s$.

Proof: Similar to that for Property 3.1.

For a non-dominant system, there exists at least one level j such that at least one of the minimal path sets $P^j_1, P^j_2, \cdots, P^j_r$ satisfies $\phi(P^j_i) > j, i = 1, 2, \cdots, r$. And there exists at least one level j such that its minimal cut sets $K^j_1, K^j_2, \cdots, K^j_k$ satisfy $\phi(K^j_i) < j - 1, i = 1, 2, \cdots, k$.

Since all minimal path sets to level j of a dominant system result in system level j, we can dichotomize the system. But for a non-dominant system, some minimal path sets may have $\phi(P^j_i) > j$. A minimal path set to level j could also be the minimal path set to level $j+1$, $j+2$, and so on, this is the reason why we can not dichotomize a non-dominant system.

Property 3.3: The minimal path sets to level j of a binary-imaged multi-state system are of the form $(j, \cdots, j, 0, \cdots, 0)$ for $j = 1, 2, \cdots, M$ or one of its permutations.

Proof:

With respect to level j, assume that there exists one minimal path $\mathbf{y}$, which is not of form $(\underbrace{j, \cdots, j}_{m}, \underbrace{0, \cdots, 0}_{n-m})$, where the form $(\underbrace{j, \cdots, j}_{m}, \underbrace{0, \cdots, 0}_{n-m})$ means that the vector contains m elements equal to j and other $n - m$ elements equal to 0, but the positions of these n elements are exchangeable. Without loss of generality, assume $\mathbf{y} = (\underbrace{j, \cdots, j}_{m}, \underbrace{r, \cdots, r}_{k}, 0, \cdots, 0)$ and $1 \le r < j$, i.e., there are not only m elements equal to j, but also k non-zero elements (their states are at 1 or above, but below j) in $\mathbf{y}$. Now select a component state vector $\mathbf{x} = (\underbrace{j, \cdots, j}_{m}, 0, \cdots, 0)$. Since $\mathbf{y}$ is a minimal path to level j and $\mathbf{x} < \mathbf{y}$, we have $\phi(\mathbf{x}) < j$. Thus, for level j, we have: $\mathbf{x}^j = \mathbf{y}^j$, but $\phi^j(\mathbf{x}) \neq \phi^j(\mathbf{y})$. This contradicts Definition 2.3. Thus, if $\mathbf{y}$ is a minimal path to level j of a multi-state system, then the system is not a binary-imaged system. (END)

If a binary-imaged dominant system has n minimal path sets at level j, it is a parallel structure; if it has one minimal path set at level j, then it is a series structure. Property 3.3 is a strong restriction on multi-state systems resulting from Definition 2.3.

Usually, a dominant system may have minimal path of a form different from that stated in Property 3.3, for example, $(j, \cdots, j, s, \cdots, s, t, \cdots, t)$ for $j > s > t$. In this case, we can not say the system is binary-imaged. In the following, we will present a method to deal with minimal paths of a dominant system with a form of $(j, \cdots, j, s, \cdots, s)$ or one of its permutations. Here, we require that each minimal path of a dominant system exactly includes two distinct values as its elements, say $(j, \cdots, j, s, \cdots, s)$ and its permutations, where $j > s$ and s may not be zero.

With regard to level j, to calculate $\Pr(\phi(\mathbf{x}) \geq j)$, what we concern is that component state probability P_{ij} ("functioning") and $\Pr(s \leq x_i < j)$ ("failed"). The component state probability $\Pr(x_i < s)$ can be ignored for calculating $\Pr(\phi(\mathbf{x}) \geq j)$. Assume $p_i = P_{ij}$ and $q_i = Q_{ij} - Q_{is}$, we can extend the corresponding binary algorithm to such a multi-state system if its structure is analogous to some binary structure. Note that the condition $p_i + q_i = 1$ always satisfied in the binary context is not necessarily satisfied here. Thus the binary algorithms need to be modified. This method has been applied to establish the bounds of system performance distribution for multi-state consecutive k-out-of-n systems (Huang, Zuo and Fang, 2001). In the following, we use an example to illustrate this point.

Example 6.4:

Consider system D in Example 6.2. The system has a parallel structure at system level 1 because the system is at level 1 or above if at least one component is at state 1 or above. the system has neither a series nor a parallel structure at level 2. But we notice that the system is at level 2 if at least one component is at state 2 with the restriction that no component is allowed to be below state 1. It is analogous a "parallel" structure. Let us see how the binary algorithm for parallel systems can be extended to such a system. Recall a binary parallel system with n components. If the reliability of component i is p_i and the unreliability is q_i, then the system reliability can be calculated:

$$R_s = 1 - \prod_{i=1}^{n}(1 - p_i) = 1 - \prod_{i=1}^{n} q_i \tag{6.7}$$

Now if the minimal paths to level j of a multi-state system is $(j, \cdots, j, s, \cdots, s)$ and its permutations, where s may be non-zero, we can treated it as a parallel structure. With regard to system level j, no components are allowed to have a state below s,. i.e., those states below s can be ignored when calculating R_{sj}. The maximum system reliability R_{sj} of a "multi-state system" when ignoring states below s is $\prod_{i=1}^{n} P_{is}$ rather than 1. Thus the following formula can be used to calculate R_{sj}:

$$R_{sj} = \prod_{i=1}^{n} P_{is} - \prod_{i=1}^{n}(Q_{ij} - Q_{is}), \tag{6.8}$$

Now in Example 6.2, we assume $p_{i0} = 0.1$, $p_{i1} = 0.2$ and $p_{i2} = 0.7$ in system D. The system reliabilities are calculated:

$R_{s1} = 1 - \prod_{i=1}^{2} Q_{i1} = 1 - 0.1 \times 0.1 = 0.99.$

$R_{s2} = \prod_{i=1}^{2} P_{i1} - \prod_{i=1}^{2}(Q_{i2} - Q_{i1}) = 0.9 \times 0.9 - 0.2 \times 0.2 = 0.77.$

$r_{s2} = R_{s2} = 0.77.$

$r_{s1} = R_{s1} - R_{s2} = 0.99 - 0.77 = 0.22.$

$r_{s0} = 1 - r_{s1} - r_{s2} = 1 - 0.22 - 0.77 = 0.01.$

In general, there are three fundamental elements in a formula used for binary system reliability evaluation: maximum system reliability, it is always equal to 1; reliability p_i and unreliability q_i of component i ($i = 1, 2, \cdots, n$). In a multi-state system, if its minimal paths to every state level j are of form $(j, \cdots, j, s, \cdots, s)$, then we use the maximum system reliability $\prod_{i=1}^{n} P_{is}$ to replace 1, P_{is} to replace p_i and $Q_{ij} - Q_{is}$ to replace q_i. Thus the binary algorithm can be extended to the multi-state system.

Usually, the number of different values in each minimal path vector of a dominant system may be greater than two. We can not use binary reliability evaluation algorithms to evaluate the exact probability that the system is in state j or above. In Chapter 5, we have used a bounding technique to deal with the decreasing multi-state consecutive k-out-of-n:G system when this happens. This method can be used for any multi-state systems.

Suppose we have a minimal path vector for system state level j, denoted by $\mathbf{y}$, which is in the form shown below or one of its permutations:

$$\mathbf{y} = (\underbrace{\underbrace{j, \cdots, j}_{k}, r_1, \cdots, r_1, r_2, \cdots, r_2, \cdots, r_m, \cdots, r_m}_{n})$$

where $r_i, i = 1, \cdots, m$ is a set of numbers taking from $\{0, 1, \cdots, j-1\}$ and $r_i > r_{i+1}$.

We will use $\mathbf{y}_j^*$ to represent all minimal path vectors to system state level j. Then, we have

$$\Pr(\phi \geq j) = \Pr(\mathbf{x} \geq \mathbf{y}_j^*), \tag{6.9}$$

where $\mathbf{x}$ represents all possible component state vectors. Now define:

$$s = Min\{i|i \in \mathbf{y}, i < j\}, \quad (6.10)$$

$$t = Max\{i|i \in \mathbf{y}, i < j\}. \quad (6.11)$$

Note that we have $0 \leq s \leq t < j$. Vectors $\mathbf{L}$ and $\mathbf{U}$ of dimension n are defined as

$$\mathbf{L} = \underbrace{(\underbrace{j, \cdots, j}_{k}, s, \cdots, s)}_{n}, \quad (6.12)$$

$$\mathbf{U} = \underbrace{(\underbrace{j, \cdots, j}_{k}, t, \cdots, t)}_{n}. \quad (6.13)$$

Obviously, we have

$$\mathbf{L} \leq \mathbf{y} \leq \mathbf{U}. \quad (6.14)$$

Then, we have

$$\Pr(\mathbf{x} \geq \mathbf{U}_j^*) \leq \Pr(\phi \geq j) = \Pr(\mathbf{x} \geq \mathbf{y}_j^*) \leq \Pr(\mathbf{x} \geq \mathbf{L}_j^*) \quad (6.15)$$

Inequality (6.15) can be used to find upper and lower bounds on the probability that the system is in state j or above. When $s = t$, the two bounds are the same and equal to the exact value of $\Pr(\phi \geq j)$. We use the following examples to illustrate this method.

Example 6.5:

Consider a three-component system wherein both the system and the components may be in one of four possible states, 0, 1, 2, and 3. The relationship between system state and component states are shown in the following table.

In the table, the "+" sign represents the permutations of the states of the components listed above the "+" sign. Assume all the components are i.i.d. and let

$p_0 = 0.1$, $p_1 = 0.2$, $p_2 = 0.3$, $p_3 = 0.4$. The minimal path to level 1 is $(1, 1, 1)$. We can use the binary formula directly.

$\Pr(\phi \geq 1) = P_{11}P_{21}P_{31} = 0.729.$

The minimal paths to level 2 are $(2, 2, 1)$ or its permutations and they are the two-element vectors. The system structure is analogous to a k-out-of-n system (a 2-out-of-3:G structure) because the system is at state 2 or above if at least two components are at in state 2 or above, but no components are allowed to be below state 1. Recall that the following formula is used for calculating the reliability of binary k-out-of-n:G system with i.i.d. components:

$$R_s = \sum_{i=k}^{n} \binom{n}{i} p^i q^{n-i}$$

$\phi(\mathbf{x})$:	0	1	2	3
	(0, 0, 0)	(1, 1, 1)	(2, 2, 1)	(3, 2, 1)
	(1, 0, 0)	(2, 1, 1)	(2, 2, 2)	(3, 2, 2)
	(2, 0, 0)	(3, 1, 1)	+	(3, 3, 1)
	(1, 1, 0)	+		(3, 3, 2)
	(2, 1, 0)			(3, 3, 3)
$\mathbf{x}$:	(2, 2, 0)			+
	(3, 0, 0)			
	(3, 1, 0)			
	(3, 2, 0)			
	(3, 3, 0)			
	+			

Table 6.6: System state table for Example 6.5

When extending this formula for our system, $p_i + q_i \neq 1$ here so we need modifying the formula.

$\Pr(\phi \geq 2) = \sum_{i=2}^{3} \binom{3}{i} (P_2)^i (1 - P_2 - p_0)^{3-i} = \sum_{i=2}^{3} \binom{3}{i} \times 0.7^i \times 0.2^{3-i} = 0.637.$

As the minimal paths to level 3 are $(3,2,1)$ and its permutations, which include more than two values, we select $\mathbf{L} = (3,2,2)$ and $\mathbf{U} = (3,1,1)$. Assume $\mathbf{L}$ and $\mathbf{U}$ and their permutations to be the minimal paths of two structures. The structures identified by $\mathbf{L}$ and $\mathbf{U}$ are denoted by ϕ_1 and ϕ_2 respectively. Since $(3,1,1) < (3,2,1) < (3,2,2)$, then $\Pr(\phi_2 \geq j) \leq \Pr(\phi \geq j) \leq \Pr(\phi_1 \geq j)$. Using Equation (6.8):

$\Pr(\phi_1 \geq 3) = \prod_{i=1}^{3} P_{i2} - \prod_{i=1}^{3} (Q_{i3} - Q_{i2}) = 0.7^3 - 0.3^3 = 0.316.$

$\Pr(\phi_2 \geq 3) = \prod_{i=1}^{3} P_{i1} - \prod_{i=1}^{3} (Q_{i3} - Q_{i1}) = 0.9^3 - 0.5^3 = 0.604.$

The bounds are: $0.218 < \Pr(\phi \geq 3) < 0.316$ while the exact value of R_{s3} is 0.556.

$r_{s0} = 1 - P(\phi \geq 1) = 1 - 0.729 = 0.271.$

$r_{s1} = \Pr(\phi \geq 1) - \Pr(\phi \geq 2) = 0.729 - 0.637 = 0.092.$

$r_{s2} = \Pr(\phi \geq 2) - \Pr(\phi \geq 3) = 0.637 - \Pr(\phi \geq 3).$

$0.033 < r_{s2} < 0.321$. while the exact value of r_{s2} is 0.081.

$0.316 < r_{s3} < 0.604.$

In the following, we provide an example (a k-out-of-n:G system model) worked out for the illustration of the properties reported in this chapter on multi-state systems.

<u>Example 6.6:</u>

Consider a pumping system which consists of four pumps wherein each pump is a component of the system. First, we consider the binary case: both the system and the components have only two states, either working or failed. The system is working if at least three pumps are working. Such a system can be described using a binary 3-out-of-4:G system model. Assume that all the components are i.i.d. and $p = 0.95$ and $q = 0.05$. Equation (2.13) can be used to calculate system reliability.

$$R_s = \sum_{i=k}^{n} \binom{n}{i} p^i q^{(n-i)} = \sum_{i=3}^{4} \binom{4}{i} 0.95^i \times 0.05^{(4-i)} = 0.986$$

Now we extend the pumping system to the multi-state case by allowing both the system and the components to have three states: state 2 (perfect working state), state 1 (marginal working state) and state 0 (failed). Assume that all the components are i.i.d. and $p_2 = 0.8, p_1 = 0.15$ and $p_0 = 0.05$. To determine the system state in terms of the component states, customers may choose different reliability models in accordance to their individual concerns. As an example, we consider the following three scenarios:

Scenario I: The system is in state j or above if at least three components are in state j or above for $j = 1, 2$. Such a system can be described using a constant multi-state 3-out-of-4:G system model where $k = 3$. The system is binary-imaged and can be treated like a binary k-out-of-n:G system. The relationship between system state and component states are shown in Table 6.7. In this table, the "+" sign represents the permutations of the states of the components listed above the "+" sign. Equation (4.1) can be used to calculate the system performance.

$$R_{s2} = \sum_{i=k}^{n} \binom{n}{i} P_2^i Q_2^{(n-i)} = \sum_{i=3}^{4} \binom{4}{i} 0.8^i \times 0.2^{(4-i)} = 0.819$$

$$R_{s1} = \sum_{i=k}^{n} \binom{n}{i} P_1^i Q_1^{(n-i)} = \sum_{i=3}^{4} \binom{4}{i} 0.95^i \times 0.05^{(4-i)} = 0.986$$

$$r_{s2} = R_{s2} = 0.819$$

$$r_{s1} = R_{s1} - R_{s2} = 0.986 - 0.819 = 0.167$$

$$r_{s0} = 1 - r_{s1} - r_{s2} = 0.014$$

Scenario II: The system is in state 2 if at least four components are in state 2 and

$\phi(\mathbf{x})$:	0	1	2
	(0, 0, 0, 0)	(1, 1, 1, 0)	(2, 2, 2, 0)
	(1, 0, 0, 0)	(1, 1, 1, 1)	(2, 2, 2, 1)
	(1, 1, 0, 0)	(2, 1, 1, 0)	(2, 2, 2, 2)
$\mathbf{x}$:	(2, 0, 0, 0)	(2, 1, 1, 1)	+
	(2, 1, 0, 0)	(2, 2, 1, 0)	
	(2, 2, 0, 0)	(2, 2, 1, 1)	
	+	+	

Table 6.7: System state table for Example 6.6: Scenario I

is in state 1 or above if at least three components are in state 1 or above. Such a system can be described using an increasing multi-state k-out-of-n:G system model with $k_1 = 3$ and $k_2 = 4$. The relationship between system state and component states are shown in Table 6.8. In this table, the "+" sign represents the permutations of the states of the components listed above the "+" sign. The system has dominant property since all the state vectors resulting in state 2 are bigger than at least one vector resulting in state 1. Compared with Definition 2.3, we find that the system is a binary-imaged system. Equation (4.1) can be extended to the evaluation of the system performance.

$$R_{s2} = \sum_{i=k_2}^{n} \binom{n}{i} P_2^i Q_2^{(n-i)} = \sum_{i=4}^{4} \binom{4}{i} 0.8^i \times 0.2^{(4-i)} = 0.410$$
$$R_{s1} = \sum_{i=k_1}^{n} \binom{n}{i} P_1^i Q_1^{(n-i)} = \sum_{i=3}^{4} \binom{4}{i} 0.95^i \times 0.05^{(4-i)} = 0.986$$
$$r_{s2} = R_{s2} = 0.410$$
$$r_{s1} = R_{s1} - R_{s2} = 0.986 - 0.410 = 0.576$$
$$r_{s0} = 1 - r_{s1} - r_{s2} = 0.014$$

$\phi(\mathbf{x})$:	0	1	2
	(0,0,0,0)	(1,1,1,0)	(2,2,2,2)
	(1,0,0,0)	(1,1,1,1)	+
	(1,1,0,0)	(2,1,1,0)	
$\mathbf{x}$:	(2,0,0,0)	(2,1,1,1)	
	(2,1,0,0)	(2,2,1,0)	
	(2,2,0,0)	(2,2,1,1)	
	+	(2,2,2,0)	
		(2,2,2,1)	
		+	

Table 6.8: System state table for Example 6.6: Scenario II

Scenario III: The system is in state 2 if at least two components are in state 2 and is in state 1 or above if at least three components are in state 1 or above. Such a system can be described using a decreasing multi-state k-out-of-n:G system model with $k_1 = 3$ and $k_2 = 2$. The relationship between system state and component states are shown in Table 6.9. In this table, the "+" sign represents the permutations of the states of the components listed above the "+" sign. The system is not a dominant system since some state vectors resulting in state 2 are not bigger than any vector resulting in state 1. For example, $\phi(2,2,0,0) = 2$, but $\mathbf{x} = (2,2,0,0)$ is not bigger than any vector resulting in state 1. Vector $(2,2,0,0)$ is not only a minimal path to level 2, but also a minimal path to level 1. The binary formula can not be applied on such a system. Equations (4.6) and (4.7) can be used for the evaluation of the system performance.

$$r_{s2} = \sum_{i=k_2}^{n} \binom{n}{i} P_2^i Q_2^{(n-i)} = \sum_{i=2}^{4} \binom{4}{i} 0.8^i \times 0.2^{(4-i)} = 0.9728$$

$$r_{s1} = \sum_{k=k_1}^{n} \binom{n}{k} p_0^{(n-k)} (p_1^k + \beta_2(k))$$

$$\beta_2(k) = \sum_{i_1=1}^{(k_2-1)} \binom{k}{i_1} p_2^{i_1} p_1^{(k-i_1)} = k \times p_2 \times p_1^{k-1}$$

$$r_{s1} = \sum_{k=k_1}^{n} \binom{n}{k} p_0^{(n-k)} (p_1^k + k \times p_2 \times p_1^{k-1}) = 0.0228$$

$$r_{s0} = 1 - r_{s1} - r_{s2} = 0.0044$$

$\phi(\mathbf{x})$:	0	1	2
	(0,0,0,0)	(1,1,1,0)	(2,2,0,0)
	(1,0,0,0)	(1,1,1,1)	(2,2,1,0)
$\mathbf{x}$:	(1,1,0,0)	(2,1,1,0)	(2,2,1,1)
	(2,0,0,0)	(2,1,1,1)	(2,2,2,0)
	(2,1,0,0)	(2,2,1,1)	(2,2,2,1)
	+	+	(2,2,2,2)
			+

Table 6.9: System state table for Example 6.6: Scenario III

6.5 Concluding remarks

In this chapter, we define the dominant multi-state system. This definition reveals the inherent relationship between binary systems and multi-state systems. In general, multi-state systems can be divided into two groups: dominant systems and non-dominant systems. A dominant system is a binary-prone system. If a dominant system is binary-imaged, then it can treated completely like a binary system. Existing binary algorithms can still be applied on some dominant system with modifications even these systems may not be binary-imaged. A new bounding technique is developed for dominant systems. In summary, dominant systems have the following advantages than non-dominant systems:

1. Easy to find all the minimal path sets or cut sets;

2. Usually, the number of the minimal path sets are less;

3. Convenient for system performance evaluation or bound computation.

The performed research also answers some questions in the previous chapters, for example, the dual relationship between multi-state consecutive k-out-of-n:F system and consecutive k-out-of-n:G system; the properties of minimal paths and cuts of multi-state systems. "Dominant system" is a newly-proposed topic in this thesis. There are many interesting issues left. For example, we have discovered that for some system states, the dominance condition may become satisfied in its dual system even though it is not satisfied in the primal system. This means that a dual transformation, which has been applied in Chapter 5, may be used to transform a non-dominant system into a dominant system. Further investigation on this subject may produce very interesting and useful research results for multi-state system performance evaluation.

Chapter 7

Conclusions

In this thesis, a systematic review of the multi-state system reliability theory is provided. Over the past two decades, extensive research has been conducted on defining component relevancy conditions and exploring the relationship between binary systems and multi-state systems. In a multi-state system, the relevancy condition represents the relation between the state of the system and the states of the components. Under different relevancy conditions, the structure of a multi-state system may be very complicated. A multi-state system may have different structures at different system state levels. However, other researchers assume that the system has the same structure at all state levels when extending the definitions of series system, parallel system and k-out-of-n system in the binary context to the multi-state context. This assumption only represents the simplest extension because it hides the flexibility of multi-state system structure for reflecting practical problems and then limits the applications of the multi-state systems. Multi-state system reliability extended many concepts from binary systems. However, a multi-state system is much more complex than a binary system. It is necessary to investigate the relationship between binary systems and multi-state systems. Although some research has been performed in this direction, not many satisfactory results are available.

This research contributes to multi-state k-out-of-n structures considering systems with different structures at different state levels. The proposed multi-state k-out-of-n systems and consecutive k-out-of-n systems not only extend the ideas of the corresponding binary systems, but also expand the applications of k-out-of-n structures in practice. The algorithms and bounding techniques developed for these systems incorporate many concepts of the multi-state reliability theory, for examples, minimal paths and minimal cuts, dual and binary decomposition. Based on observation on different multi-state k-out-of-n systems and consecutive k-out-of-n systems, we propose a definition of the dominant system. This definition reveals the inherent relation between binary systems and multi-state systems.

In Chapter 4, we propose a definition of the generalized multi-state k-out-of-n:G system by releasing the assumption that a multi-state k-out-of-n system must have the same structure imposed by other researchers. Based on practical examples, we concentrate on two special cases: the increasing system and the decreasing system. An increasing k-out-of-n system is totally like a binary system such that the binary formula can be used. A new algorithm for the decreasing k-out-of-n system is developed in this thesis.

In Chapter 5, we define the multi-state consecutive k-out-of-n:F system. Similar to the multi-state k-out-of-n system, two special cases are taken into consideration. For a decreasing consecutive k-out-of-n:F system, we can treat it like a binary system. However, we find that it is very difficult to develop an efficient formula for calculating the performance distribution of an increasing consecutive k-out-of-n:F system. Alternatively, we provide a new bounding technique for the increasing consecutive k-out-of-n:F system. This technique can be applied to general multi-state systems. The work performed in Chapter 4 and 5 not only deals with multi-state k-out-of-n structures, but also attempts to provide answers to some questions encountered in the literature review. For example, the duality relation between consecutive k-out-of-n:F system and consecutive k-out-of-n:G system is investigated. Based on the definition of the dual of multi-state systems, some special duality relations in binary k-out-of-n structures are lost in the multi-state case.

Chapter 6 generalizes the "domination phenomenon" existing in the increasing k-out-of-n:G system and the decreasing consecutive k-out-of-n:F system and then proposes a definition of the dominant multi-state system. In general, we can divide multi-state systems into two groups: dominant systems and non-dominant systems. This classification provides a tool for judging whether a multi-state system is dichotomous at all system state levels or not. If a multi-state system is a dominant system, we can extend binary algorithm to evaluate its performance distribution or use the bounding technique presented in this work to establish the bounds of the system state distribution. Furthermore, if a dominant system is binary-imaged, we can treat it like a binary system. For a non-dominant system, we could use the dual conversion method to transform it into a dominant system. These results can be applied on any multi-state systems systems without referred to relevancy conditions.

Along the line of this thesis, further studies can be done in the following areas:

- Finding efficient algorithms for multi-state k-out-of-n structures;
- Investigating the duality relation between multi-state k-out-of-n:G and F structures;
- Studying on improving the bounds of the performance distribution of the dominant system;
- Investigating the relation between dominant systems and non-dominant systems.

In summary, this research introduces two multi-state k-out-of-n structures, the generalized multi-state k-out-of-n:G system and multi-state consecutive k-out-of-n:F systems, and then proposes a definition of the dominant multi-state system. It makes significant contributions to multi-state system modeling, reliability evaluation and analysis.

Bibliography

[1] Abouammoh, A. M. and Al-Kadi, M. A. (1991), "Component relevancy in multi-state reliability models" , *IEEE Tran. On Reliability*, Vol. R-40, No. 3, pp: 370-374.

[2] Abouammoh, A. M. and Al-Kadi, M. A. (1995), "Multi-state coherent systems of order k", *Microelectronic Reliability*, Vol. 35, No.11, pp: 1415-1421.

[3] Amari, S. V. and Misra, R. B. (1997), "Comments on: dynamic reliability analysis of coherent multi-state systems", *IEEE Tran. On Reliability*, Vol. R-46, No. 4, pp: 460-461.

[4] Andrzejczak, K. (1992), "Structure analysis of multi-state coherent systems", *Optimization*, Vol. 25, pp: 301-316.

[5] Ansell, J. I. and Bendell, A. (1987), " On alternative definitions of multi-state coherent systems", *Optimization*, Vol. 18, No.1, pp: 119-136.

[6] Aven, T. (1985), "Reliability evaluation of multi-state systems with multi-state components", *IEEE Tran. On Reliability*, Vol. R-34, No. 5, pp: 473-479.

[7] Aven, T. (1993), "On performance measure for multi-state monotone systems", *Reliability Engineering and System Safety*, Vol. 41, pp: 259-266.

[8] Bai, T. and Zhang, R. (1997), "Building of a multi-state reliability model of large generating units", *Journal of Shanghai Jiaotong University*, Vol. 31, No. 6, pp: 33-37.

[9] Barlow, R. and Proschan, F. (1975), " Statistical Theory of Reliability and Life Testing (Probability Models)", *Holt, Rinehart and Winston Inc.*.

[10] Barlow, R. E. and Heidtmann, K. D. (1984), "Computing k-out-of-n system reliability", *IEEE Tran. On Reliability*, Vol. 33, No. 4, pp: 322-323.

[11] Barlow, R. E. and Wu A. S. (1978), "Coherent systems with multi-state components", *Mathematics of Operations Research*, Vol. 3, No. 4, pp: 275–281.

[12] Bhuiyan, M. R. and Allan, R. N. (1994), "Modeling multi-state problems in sequential simulation of power system reliability studies", *IEE Proc-Gener Transm. Distrib.*, Vol. 142, No. 4, pp: 343-349.

[13] Block, H. W., Griffith, W. S. and Savits, T. H. (1989), "L-superadditive structure functions", *Applied Probability*, Vol. 21, pp: 919-929.

[14] Block, H. W. and Savits, T. H. (1982), "A decomposition for multi-state monotone systems", *Journal Applied Probability*, Vol. 19, pp: 391-402.

[15] Block, H. W. and Savits, T. H. (1984), "Continuous multistate structure functions", *Operations Research*, Vol. 32, No. 3, pp: 703-714.

[16] Boedigheimer, R. A. and Kapur, K.C. (1994), "Customer-driven reliability models for multi-state coherent systems", *IEEE Tran. On Reliability*, Vol. 43, No.1, pp: 46-50.

[17] Borges, W. D. S. and Rodrigues, F. W. (1983), "An axiomatic characterization of multistate coherent structure", *Mathematics of Operations Research*, Vol. 8, No. 3, pp: 435-438.

[18] Brunell, R. D. and Kapur, K. C. (1997), "Customer-centered reliability methodology", *Proceedings Annual R&M Symposium*, pp: 286-292.

[19] Butler, D. A. (1982), "Bounding the reliability of multi-state systems", *Operations Research*, Vol. 30, No. 3, pp: 530-544.

[20] Cafaro, G., Corsi, F. and Vacca, F. (1986), "Multi-state Markov models and structural properties of the transition-rate matrix", *IEEE Tran. On Reliability*, Vol. R-35, No. 2, pp: 192-200.

[21] Cao, J. and Wu, Y. (1988), "Reliability analysis of a multi-state system with a replaceable repair facility", *ACTA Mathematical Application SINICA*, Vol. 4, No. 3, pp:113-121.

[22] Chao, M. T., Fu, J. C. and Koutras, M. V. (1995), "Survey of reliability studies of consecutive-k-out-of-n:F & related Systems." *IEEE Tran. On Reliability*, Vol. 44, No. 1, pp: 120-125.

[23] Chiang, D. T. and Niu, S. (1981), "Reliability of consecutive-k-out-of-n:F system", *IEEE Tran. On Reliability*, Vol. 30, No. 1, pp: 87-89.

[24] Chung, K. L. (1979), "Elementary Probability Theory With Stochastic Processes", *Springer-Verlag*, New York.

[25] El-Neweihi, E., Proschan, F. and Sethuraman, J. (1978), "Multi-state coherent system", *Journal of Applied Probability*, Vol. 15, pp: 675-688.

[26] Ebrahimi, N. (1984), "Multi-state reliability models", *Naval Research Logistics Quarterly*, Vol. 31, pp: 671-680.

[27] Griffith, W. S. (1980), "Multi-state reliability models", *Journal Applied Probability*, Vol. 17, pp: 735-744.

[28] Hirsch, W. M. and Meisner, M. (1968), "Cannibalization in multi-component systems and the theory of reliability", *Naval Research Logist. Quart.*, No. 15, pp: 331-360.

[29] Hiam, M. and Porat, Z. (1991), "Bayes reliability modeling of a multistate consecutive k-out-of-n:F system", *Proceeding Annual Reliability and Maintainability Symposium*, pp: 582-586.

[30] Huang, J. and Wu, L. (1990), "A new approach to make a policy decision on mining equipment maintenance", *Quarterly of CIMR*, Vol. 10, No. 1, pp: 29-34.

[31] Huang, J. and Zuo, M. (1997), "Estimating furnace remaining life using Monte Carlo simulation", *Proceedings of the IASTED Conference, Applied Modeling and Simulation*, Banff, pp: 443-446.

[32] Huang, J., Zuo, M. and Wu, Y. (2000), "Generalized multi-state k-out-of-n:G system", *IEEE Tran. On Reliability*, Vol. 49, No. 1, pp: 105-111.

[33] Huang, J., Zuo, M. and Fang, Z. (2000), "Multi-state consecutive k-out-of-n systems", submitted to *IIE Tran. On Reliability and Quality.*

[34] Huang, J. and Zuo, M. (2001), "Dominant multi-state system and its properties", submitted to *IEEE Tran. On Reliability.*

[35] Huang, J. and Zuo, M. (2000), "Multi-state k-out-of-n system model and its applications", *Proceedings of Annual RAM Symposium*, pp: 264-268.

[36] Hudson, J. C. and Kapur, K. C. (1983), "Reliability analysis for multi-state systems with multi-state components", *IIE Transaction*, Vol. R-32, No. 2, pp: 127-135.

[37] Hudson, J. C. and Kapur, K. C. (1983), "Modules in coherent multi-state systems", *IEEE Tran. On Reliability*, Vol. 15, No. 2, pp: 183-185.

[38] Hudson, J. C. and Kapur, K. C. (1985), "Reliability bounds for multistate systems with multistate components", *Operations Research*, Vol. 33, No. 1, pp: 153-160.

[39] Hwang, F. K. (1982), "Fast solutions for consecutive-k-out-of-n:F system", *IEEE Tran. On Reliability*, Vol. 31, No. 5, pp: 447-448.

[40] Hwang, F. K. and Yao, Y. C. (1989), "Multi-state consecutively-connected systems", *IEEE Tran. On Reliability*, Vol. 38, No. 4, pp: 472-474.

[41] Karpinski, J. (1986), "A multi-state system under an inspection and repair policy", *IEEE Tran. On Reliability*, Vol. R-35, No. 1, pp: 76-77.

[42] Korczak, E. (1993), "Reliability analysis of multistate monotone systems", *Safety and Reliability Assessment*, Vol. 2, pp: 671-682.

[43] Kossow, A. and Preuss, W. (1995), "Reliability of linear consecutively-connected systems with multistate Components", *IEEE Tran. On Reliability*, Vol. 44, No. 3, pp: 518-522.

[44] Koval, D. (1996), "Reliability Engineering (EE 514 class notes)", *Department of Electrical Engineering*, University of Alberta.

[45] Kuhnert, I. (1995), "Comments on "Component relevancy in multi-state reliability models" , *IEEE Tran. On Reliability*, Vol. R-44, No. 1, pp: 95-96.

[46] Kuo, W., Zhang, W. and Zuo, M. (1990), "A consecutive-k-out-of-n:G system: the mirror image of a consecutive-k-out-of-n:F system", *IEEE Tran. On Reliability*, Vol. 39, No. 2, pp: 244-253.

[47] Kuo, W., Fu, J. C. and Lou, W. Y. (1994), "Opinions on consecutive-k-out-of-n:F system", *IEEE Tran. On Reliability*, Vol. 43, No. 4, pp: 659-662.

[48] Lesanovsky, A. (1988), "Multi-state Markov models for system with dependent units", *IEEE Tran. On Reliability*, Vol. R-37, No. 5, pp: 505-511.

[49] Levitin, G. and Lisnianski, A. (2000), "Optimization of imperfect preventive maintenance for multi-state systems", *Reliability Engineering and System Safety*, Vol. 67, pp: 193-203.

[50] Levitin, G., Lisnianski, A., Ben-Haim, H. and Elmarkis, D. (1998), "Redundancy optimization for series-parallel multi-state systems", *IEEE Tran. On Reliability*, Vol. 47, No. 2, pp: 165-172.

[51] Lisnianski, A., Levitin, G. and Ben-Haim, H. (2000), "Structure optimization of multi-state system with time redundancy", *Reliability Engineering and System Safety*, Vol. 67, pp: 103-122.

[52] Mak, K. (1989), " A note on Barlow-Wu structure functions", *Operations Research Letter*, Vol. 8, pp: 43-44.

[53] Malinowski, J. and Preuss, W. (1995), "Reliability of circular consecutively-connected systems with multi-state components", *IEEE Tran. On Reliability*, Vol. 44, No. 3, pp: 532-534.

[54] Malinowski, J. and Preuss, W. (1996), "Reliability of reverse-tree-structured systems with multi-state components", *Microelectronic Reliability*, Vol. 36, No. 1, pp: 1-7.

[55] Meng, F. C. (1989), " Contributions to Multi-state Reliability Theory", *Ph.D. Dissertation, Department of Industrial Engineering*, University of Illinois.

[56] Meng, F. C. (1993), "Component-relevancy and characterization results in multi-state systems", *IEEE Tran. On Reliability*, Vol. R-42, No. 3, pp: 478-483.

[57] Meng, F. C. (1994), "Characterizing the Barlow-Wu structure functions", *IEEE Tran. On Reliability*, Vol. 41, pp: 661-668.

[58] Natvig, B. (1982), "Two suggestions of how to define a multi-state coherent system", *Applied Probability*, Vol. 14, pp: 434-455.

[59] Ohi, F. and Nishida, T. (1984), "On multi-state coherent systems", *IEEE Tran. On Reliability*, Vol. R-33, No. 4, pp: 284-288.

[60] Ross, S. M. (1979), "Multivalued state component systems", *The Annals of Probability*, Vol. 7, No. 2, pp: 379-383.

[61] Rushdi, A. M. (1986), "Utilization of symmetric switching functions in the computation of k-out-of-n system reliability", *Microelectronics and Reliability*, Vol. 26, No. 5, pp: 973-987.

[62] Utkin, L. V. (1993), "Uncertainty importance of multi-state system components", *Microelectronic Reliability*, Vol. 33, No. 13, pp: 2021-2029.

[63] Wood, P.A. (1985), "Multi-state block diagrams and fault trees", *IEEE Tran. On Reliability*, Vol. R-34, No. 3, pp: 236-240.

[64] Xue, J. (1985), "On multi-state system analysis", *IEEE Tran. On Reliability*, Vol. R-34, No. 4, pp: 329-337.

[65] Xue, J. and Yang, K. (1995), "Dynamic reliability analysis of coherent multi-state systems", *IEEE Tran. On Reliability*, Vol. R-44, No. 4, pp: 683-688.

[66] Xue, J. and Yang, K. (1995), "Symmetric relations in multi-state systems", *IEEE Tran. On Reliability*, Vol. R-44, No. 4, pp: 689-693.

[67] Yeh, R. H. (1996), "Optimal inspection and replacement policies for multi-state deteriorating systems", *European Journal of Operational Research*, Vol. 96, pp: 248-259.

[68] Yu, K., Koren, I. and Guo, Y. (1994), "Generalized multi-state monotone coherent systems", *IEEE Tran. On Reliability*, Vol. R-43, No. 2, pp: 242-250.

[69] Zuo, M. (1993), "Reliability and component importance of a consecutive-k-out-of-n system", *Microelectronic Reliability*, Vol. 33, No. 2, pp: 243-258.

[70] Zuo, M., Chiovelli, S. and Huang, J. (1998), "Reliability evaluation of furnace systems", *Reliability Engineering and System Safety*, No. 65, pp: 283-287.

[71] Zuo, M. J. and Liang, M. (1994), "Reliability of multistate consecutively-connected systems", *Reliability Engineering and System Safety*, Vol. 44, pp: 173-176.

[72] Zuo, M., Jiang, R. and Yam, R. (1999), "Approaches for reliability modeling of continuous-state devices", *IEEE Tran. On Reliability*, Vol. 48, No.1, pp: 9-18.