Study on Reliability Evaluation and Optimal Design of Connected-X-out-of-(m, n):F Lattice Systems

Connected-X-out-of-(m, n):F システムの信頼度評価と最適設計に関する研究

Taishin Nakamura

Graduate School of System Design,
Tokyo Metropolitan University

Abstract

Modern society has been increasingly depending on various systems that have consistently enriched our lives. However, an absolute guarantee cannot be made that such systems will perform their specific functions satisfactorily throughout their intended life spans. System failure is an unavoidable event, and it can occur under various circumstances. The consequences of such failures could significantly impact our lives, as in the cases of nuclear explosions, airplane crashes, and electrical network shutdowns. Reliability, which is defined as the probability that a component or system will perform its required function without failures under given conditions for a stated time interval, is a critical metric of system performance. The appropriate evaluation and enhancement of the reliability of such systems are critical to ensuring that they can meet their design requirements.

In reliability theory, one of the key problems is to accurately determine the reliability of a system from the knowledge of its component reliabilities. System reliability could be used as a decision-making factor when choosing between design alternatives. Thus, this thesis considers the reliability evaluation. Furthermore, system reliability plays a major role in determining system performance, wherein systems are expected to be reliable. It is necessary to design systems with high reliability, leading to the study of reliability optimization, in which enhancing system reliability is the main objective. Therefore, this thesis also focuses on the optimal design.

In practice, a system exists such that a cluster of failed components causes system failure, which can be modeled as a connected-**X**-out-of-(m, n):F lattice system. Because of theoretical development and practical applications in the research field of reliability, much effort has been devoted to studying the reliability of this system. The main objective of this thesis is to propose methods and algorithms for the reliability evaluation and the optimal design of connected-**X**-out-of-(m, n):F lattice systems.

This thesis consists of five chapters. Chapter 1 briefly explains the background and

introduces the mathematical concepts useful for understanding subsequent chapters. In addition, the literature reviews related to this thesis are detailed and systematically classified.

Chapter 2 focuses on system reliability evaluation. Most of the research has been devoted to studying the reliability of linear-type systems, whereas no study has focused on toroidal-type systems. The applications, however, emphasize the necessity of studying toroidal-type systems; therefore, we present algorithms for efficiently computing the reliability of a toroidal-type system. The numerical experiments have shown the efficiency of the algorithms. Furthermore, as the size of a system becomes large, obtaining the exact system reliability becomes time-consuming. Accordingly, it would be beneficial to use appropriate upper and lower bounds if the exact system reliability is not necessarily required. Thus, this thesis provides the upper and lower bounds for the system reliability in such a case. From the results of the numerical experiments, it can be concluded that the obtained bounds are tighter at the expense of the computational effort compared with the existing bounds.

Chapter 3 considers the system signature. A stochastic comparison, which compares the lifetimes of systems, can determine the system that works properly for a longer time. The system signature is essential for establishing the stochastic comparison. However, the computation of a system signature is known to be challenging, especially when a system has a large number of components. Consequently, its practical applications have generally been limited to relatively small systems. Therefore, methods for efficiently computing the system signature are proposed for a connected-**X**-out-of-(m, n):F lattice system. Numerical experiments are performed for comparing the efficiency of the proposed and existing methods. Moreover, the obtained system signature enables us to establish the stochastic comparison of these systems.

Chapter 4 addresses the component assignment problem for a linear connected-(r, s)-out-of-(m, n):F lattice system, which is one type of connected-**X**-out-of-(m, n):F lattice system. This problem aims to find a component arrangement that maximizes system reliability, otherwise known as the optimal arrangement. For the sake of enhancing system reliability, this problem is of great interest in the research field of reliability. Although an enumeration method can theoretically find the optimal arrangement, it is time-consuming and applicable only for small systems. This thesis thus develops an efficient algorithm for finding the optimal arrangement. In addition, to improve the efficiency, an algorithm specific to the case in which $r = m - 1$ and $s = n - 1$

is also proposed. The comparison with the existing algorithm demonstrates that the proposed algorithms outperform the existing one in terms of computation time. In particular, the result shows that the algorithm specific to the case performs well for a connected-$(m-1, n-1)$-out-of-(m, n):F lattice system.

The contributions of this thesis are summarized in Chapter 5, and various future perspectives are discussed. The methods and algorithms resulting from the research in this thesis will be useful for the reliability evaluation and the optimal design of practical systems that can be expressed as connected-$\mathbf{X}$-out-of-(m, n):F lattice systems.

Contents

Abstract **i**

List of Figures **vii**

List of Tables **ix**

Acronyms and Notation **xi**

1 Introduction **1**

1.1 Background 1

1.2 System Reliability Modeling 4

1.2.1 Consecutive-k System 4

1.2.2 Consecutive-k-out-of-n:F System 6

1.2.3 Connected-**X**-out-of-(m, n):F Lattice System 9

1.2.4 Other Connected-**X**-out-of-(m, n):F Lattice Systems 19

1.3 General Problems for Consecutive-k Systems 20

1.4 Research Scope and Objective 32

1.5 Organization of the Thesis 35

2 Reliability Evaluation **39**

2.1 Reliability of a Tor/(r, s)/(m, n):F System 39

2.1.1 Proposal of a Computational Method 40

2.1.2 Algorithm and Illustrative Example 46

2.1.3 Efficiency Investigation 50

2.1.4 Birnbaum Importance for a Tor/(r, s)/(m, n):F System 54

2.2 Reliability of a Connected-$(1, 2)$-or-$(2, 1)$-out-of-(m, n):F Lattice System 56

2.2.1 Notation 56

2.2.2 Computational Method for a Lin/$(1,2)$-or-$(2,1)$/(m,n):F System 57
2.2.3 Computational Method for a Cir/$(1,2)$-or-$(2,1)$/(m,n):F System 60
2.2.4 Computational Method for a Tor/$(1,2)$-or-$(2,1)$/(m,n):F System 64
2.2.5 Efficiency Investigation 66
2.3 Bounds for the Reliability of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F System . . . 72
2.3.1 Derivation of Upper and Lower Bounds 73
2.3.2 Numerical Experiment 77
2.4 Summary 80

3 Computing the System Signature **83**
3.1 Existing Methods for Computing the System Signature 83
3.2 System Signature of a Lin/(r,s)/(m,n):F System 87
3.2.1 Notation 87
3.2.2 Proposal of a Computational Method 88
3.2.3 Algorithm and Illustrative Example 92
3.2.4 Efficiency Investigation 98
3.3 System Signature of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F System 100
3.3.1 Proposal of a Computational Method 100
3.3.2 Algorithm and Illustrative Example 102
3.3.3 Efficiency Investigation 107
3.4 Stochastic Comparisons of the Connected-**X**-out-of-(m,n):F Lattice Systems 109
3.5 Summary 112

4 Component Assignment Problem **113**
4.1 CAP in a Lin/(r,s)/(m,n):F System 113
4.1.1 Formulation of the CAP 113
4.1.2 Properties of the CAP in the Lin/(r,s)/(m,n):F System 114
4.2 Algorithm for Solving the CAP in the General Case 119
4.2.1 Conditions for the Pruning 119
4.2.2 Algorithm and Illustrative Example 124
4.3 Algorithm for Solving the CAP in the Case of $r = m-1$ and $s = n-1$ 130
4.3.1 Conditions for the Pruning 130
4.3.2 Algorithm and Illustrative Example 139

4.4 Efficiency Investigation 142
4.5 Summary 148

5 Conclusions **149**
5.1 Summary of the Thesis 149
5.2 Future Work 152

Appendix **156**

References **168**

Publications **182**

Acknowledgments **185**

List of Figures

1.1 Family of the consecutive-k systems. 5
1.2 Lin/Con/k/n:F system. 6
1.3 Cir/Con/k/n:F system. 7
1.4 Lin/(r,s)/(m,n):F system. 10
1.5 Cir/(r,s)/(m,n):F system. 11
1.6 Failures of the Cir/$(2,3)$/$(5,6)$:F systems. 12
1.7 Tor/(r,s)/(m,n):F system. 13
1.8 Failure of the Tor/$(2,3)$/$(5,6)$:F system. 14
1.9 Working and failed Lin/$(1,2)$-or-$(2,1)$/(m,n):F systems. 15
1.10 Example of a supervision system. 17
1.11 Example of a toroidal alarm system. 19
1.12 Organization of this thesis (Chapters 1 and 2). 37
1.13 Organization of this thesis (Chapters 3, 4, and 5). 38

2.1 Event decomposition of a Tor/(r,s)/(m,n):F system. 40
2.2 Definition of $A(i,l;h_l;g_l)$. 42
2.3 Lin/$(1,2)$-or-$(2,1)$/(m,j):F systems with a condition. 58
2.4 Cir/$(1,2)$-or-$(2,1)$/(i,n):F systems with a condition. 63
2.5 Lin/$(2,2)$/$(3,j)$:F system when $\boldsymbol{g}=(2,1)$. 70
2.6 Lin/$(1,2)$-or-$(2,1)$/$(b-a+1,n)$:F subsystem. 73
2.7 Definition of $S^{(\gamma)}((b-a+1,n),P_{[a,b]})$. 74
2.8 Definition of $D_b([l_1,l_2],n)$. 75

3.1 Definition of $A_j(\boldsymbol{g})$. 88
3.2 Lin/$(2,2)$/$(3,j)$:F system with exactly z failed components and the additional condition given by $\boldsymbol{g}=(1,0)$. 90

3.3 Lin/$(1,2)$-or-$(2,1)/(3,j)$:F system with exactly z failed components and states of the m components in the jth column are given by $\boldsymbol{y} = (1,0,1)$. 101

4.1 Explanation of Lemma 4.1. 116
4.2 Explanation of Lemma 4.2. 117
4.3 Explanation of Proposition 4.1. 118
4.4 Example of assignment order of components in the proposed algorithm. 125
4.5 Flow chart of DFS(k). 126
4.6 Tree for enumerating arrangements of the Lin/$(2,3)/(3,4)$:F system. . . 128
4.7 System divided into nine areas. 131
4.8 Order for assigning components. 131
4.9 Circular sub-system. 135
4.10 Idea of computing the system reliability effectively. 137
4.11 Flow chart of DFS(k). 138
4.12 Tree for enumerating arrangements of the Lin/$(2,3)/(3,4)$:F system. . . 141
4.13 Optimal arrangements of Lin/$(3,4)/(4,5)$:F system when the component reliabilities are generated from a uniform distribution on $[0.10, 0.50]$. . . 143
4.14 Optimal arrangements of the Lin/$(2,7)/(3,8)$:F systems when the component reliabilities are generated from a uniform distribution on $[0.25, 0.35]$ and $[0.10, 0.50]$. 143

A.1 Relationship among $\boldsymbol{g}$, $\boldsymbol{g}'$, and $\boldsymbol{y}$. 162

List of Tables

1.1 Summary of the studies for the consecutive-k-out-of-n:F systems and the connected-$\mathbf{X}$-out-of-(m,n):F lattice systems. 31

1.2 Approaches for computing the reliability of connected-(r,s)-out-of-(m,n):F lattice systems and connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F lattice systems. 32

2.1 Time and space complexities of Algorithms A and B. 51

2.2 Comparison of computation times of Algorithms A and B ($(r,s)=(2,2)$). 52

2.3 Comparison of computation times of Algorithms A and B ($(m,n)=(20,6)$). 53

2.4 Birnbaum importance for a Tor/$(2,2)$/$(7,7)$:F system with $p_{ij}=0.7$ for any i,j. 55

2.5 Birnbaum importance for a Tor/$(2,2)$/$(7,7)$:F system with $p_{4,4}=0.3$ and $p_{ij}=0.7$ for the others. 55

2.6 Approaches for computing the reliability of connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F lattice systems. 66

2.7 Comparison of computation times of the recursive equation approach and the FMCIA for a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system in the INID case (sec.). 67

2.8 Comparison of computation times of the recursive equation approach and the FMCIA for a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system in the IID case (sec.). 67

2.9 Comparison of computation times of the recursive equation approach and the FMCIA for a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system in the INID case (sec.). 68

2.10 Comparison of computation times of the recursive equation approach and the FMCIA for a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system in the IID case (sec.). 68

2.11 Comparison of computation times of (A) Recursive equation approach and (B) Matrix-based method ($(r,s)=(4,4), p=0.05$). 71
2.12 Comparison of computation times of (A) Recursive equation approach and (B) Matrix-based method ($(m,n)=(5,1000), p=0.05$). 71
2.13 Bounds for the reliability of the Lin/$(1,2)$-or-$(2,1)$/$(36,100)$:F system and their computation times when $(\alpha,\beta)=(0.995,0.999)$ for various b. 78
2.14 Bounds for the reliability of the Lin/$(1,2)$-or-$(2,1)$/$(36,100)$:F system and their computation times when $(\alpha,\beta)=(0.995,0.999)$ for various l_1 and l_2. 79
2.15 Comparison of the derived and existing lower and upper bounds for a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system. 80

3.1 $\alpha(1;z;\boldsymbol{g})$ of the Lin/$(2,2)$/$(3,3)$:F system. 93
3.2 $\alpha(2;z;\boldsymbol{g})$ of the Lin/$(2,2)$/$(3,3)$:F system. 94
3.3 $\alpha(3;z;\boldsymbol{g})$ of the Lin/$(2,2)$/$(3,3)$:F system. 95
3.4 System signatures of the Lin/(r,s)/(m,n):F systems. 97
3.5 Comparison of computation times for system signatures of the Lin/(r,s)/(m,n):F systems (sec.). 99
3.6 $\beta(1;z;\boldsymbol{y})$ of the Lin/$(1,2)$-or-$(2,1)$/$(3,3)$:F system. 103
3.7 $\beta(2;z;\boldsymbol{y})$ of the Lin/$(1,2)$-or-$(2,1)$/$(3,3)$:F system. 104
3.8 $\beta(3;z;\boldsymbol{y})$ of the Lin/$(1,2)$-or-$(2,1)$/$(3,3)$:F system. 104
3.9 System signatures of the Lin/$(1,2)$-or-$(2,1)$/(m,n):F systems. 106
3.10 Comparison of computation times for system signatures of Lin/$(1,2)$-or-$(2,1)$/(m,n):F systems (sec.). 108
3.11 Orderings in the usual stochastic order among some Lin/(r,s)/(m,n):F systems. 111

4.1 Comparison of the proposed algorithms and the existing algorithm (sec.). 146
4.2 Efficiency of preprocessing step based on Proposition 4.1, Conditions 1 and 2 in the proposed algorithm (sec.). 146
4.3 Computation time and the number of enumerated arrangements for finding the optimal arrangement in the case of $(r,s)=(2,n-1)$ (sec.). . . 147

Acronyms and Notation

The following list the acronyms and notation, which are used in this thesis. Additional notations are defined if necessary when they are introduced for the first time.

Acronyms

- Lin/Con/k/n:F system: linear consecutive-k-out-of-n:F system
- Cir/Con/k/n:F system: circular consecutive-k-out-of-n:F system
- Lin/$(r,s)/(m,n)$:F system:
  linear connected-(r,s)-out-of-(m,n):F lattice system
- Cir/$(r,s)/(m,n)$:F system:
  circular connected-(r,s)-out-of-(m,n):F lattice system
- Tor/$(r,s)/(m,n)$:F system:
  toroidal connected-(r,s)-out-of-(m,n):F lattice system
- Lin/$(1,2)$-or-$(2,1)/(m,n)$:F system:
  linear connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F lattice system
- Cir/$(1,2)$-or-$(2,1)/(m,n)$:F system:
  circular connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F lattice system
- Tor/$(1,2)$-or-$(2,1)/(m,n)$:F system:
  toroidal connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F lattice system
- B&B: branch-and-bound
- B-importance: Birnbaum importance

- CAP: component assignment problem
- DFS: depth-first search
- FMCIA: finite Markov chain imbedding approach
- IID: independent and identically distributed
- INID: independent but non-identically distributed
- MTTF: mean time to failure

Notation

- $|\Omega|$: the number of elements of the set Ω
- $\emptyset$: empty set
- $\setminus$: set exclusive
- E^c: complement of the event E
- $\lceil a \rceil$: the smallest integer greater than or equal to a
- $\lfloor a \rfloor$: the greatest integer less than or equal to a
- $\mathcal{X}(G)$: indicator function that takes 1 if argument G is true; 0 otherwise
- $\mathrm{Tr}(A)$: trace of matrix A
- $\boldsymbol{a}^{\mathsf{T}}$: transpose of row vector $\boldsymbol{a}$
- Z_j: random variable representing the state of the jth component ($j = 1, 2, \ldots, n$)

$$Z_j = \begin{cases} 0, & \text{if the } j\text{th component works,} \\ 1, & \text{if the } j\text{th component fails.} \end{cases}$$

- p_j: reliability of the jth component, namely, $p_j = \Pr\{Z_j = 0\}$
- $\boldsymbol{p} = (p_1, p_2, \ldots, p_n)$: vector of reliabilities of n components
- $R^L(k, n, \boldsymbol{p})$: reliability of the Lin/Con/k/n:F system with component reliabilities $\boldsymbol{p}$

- $R^C(k, n, \boldsymbol{p})$: reliability of the Cir/Con/k/n:F system with component reliabilities $\boldsymbol{p}$

- Z_{ij}: random variable representing the state of component (i, j) ($i = 1, 2, \ldots, m$, $j = 1, 2, \ldots, n$)

$$Z_{ij} = \begin{cases} 0, & \text{if component } (i, j) \text{ works,} \\ 1, & \text{if component } (i, j) \text{ fails.} \end{cases}$$

- p_{ij}: reliability of component (i, j), namely, $p_{ij} = \Pr\{Z_{ij} = 0\}$

- $P = (p_{ij})_{1 \le i \le m, 1 \le j \le n}$: matrix of reliabilities of mn components

- $R^L((r, s), (m, n), P)$: reliability of the Lin/(r, s)/(m, n):F system with component reliabilities P

- $R^C((r, s), (m, n), P)$: reliability of the Cir/(r, s)/(m, n):F system with component reliabilities P

- $R^T((r, s), (m, n), P)$: reliability of the Tor/(r, s)/(m, n):F system with component reliabilities P

- $Y_{ij}^{(1,2)}$: random variable that is defined, for $i = 1, 2, \ldots, m$ and $j = 2, 3, \ldots, n$, as

$$Y_{ij}^{(1,2)} = Z_{i,j-1} \times Z_{ij}$$

- $Y_{ij}^{(2,1)}$: random variable that is defined, for $i = 2, 3, \ldots, m$ and $j = 1, 2, \ldots, n$, as

$$Y_{ij}^{(2,1)} = Z_{i-1,j} \times Z_{ij}$$

- $R^L((m, n), P)$: reliability of the Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system with component reliabilities P

- $R^C((m, n), P)$: reliability of the Cir/$(1, 2)$-or-$(2, 1)$/(m, n):F system with component reliabilities P

- $R^T((m, n), P)$: reliability of the Tor/$(1, 2)$-or-$(2, 1)$/(m, n):F system with component reliabilities P

- $I^B(i)$: B-importance of component i
- $I^B(i,j)$: B-importance of component (i,j)
- T: lifetime of the system
- $T_1, T_2, \ldots, T_N$: lifetimes of the N components
- $T_{i:N}$: the ith order statistic (that is, the ith smallest value) among $T_1, T_2, \ldots, T_N$ for $i = 1, 2, \ldots, N$
- $\boldsymbol{s}(N) = (s_1(N), s_2(N), \ldots, s_N(N))$: system signature
- $r_i(N)$: the number of path sets of a system with exactly i failed components $(i = 0, 1, \ldots, N)$
- x_{ij}: indicator variable representing the state of component (i,j) $(i = 1, 2, \ldots, m,$ $j = 1, 2, \ldots, n)$

$$x_{ij} = \begin{cases} 0, & \text{if component } (i,j) \text{ works,} \\ 1, & \text{if component } (i,j) \text{ fails.} \end{cases}$$

- $\boldsymbol{x}_j = (x_{1j}, x_{2j}, \ldots, x_{mj})$: state vector representing the states of the m components in the jth column $(j = 1, 2, \ldots, n)$
- $(\boldsymbol{x}_1^{\mathsf{T}}, \boldsymbol{x}_2^{\mathsf{T}}, \ldots, \boldsymbol{x}_n^{\mathsf{T}})$: state matrix representing the states of the mn components in a system
- $N(\boldsymbol{x}_j)$: the number of the failed components in the jth column, that is,

$$N(\boldsymbol{x}_j) = \sum_{a=1}^{m} x_{aj}$$

- $\phi^{RS}(\boldsymbol{x}_1^{\mathsf{T}}, \boldsymbol{x}_2^{\mathsf{T}}, \ldots, \boldsymbol{x}_n^{\mathsf{T}})$: structure function of a Lin/(r,s)/(m,n):F system
- $\alpha_z((r,s),(m,n))$: the number of path sets of a Lin/(r,s)/(m,n):F system with exactly z failed components
- $\phi^{OR}(\boldsymbol{x}_1^{\mathsf{T}}, \boldsymbol{x}_2^{\mathsf{T}}, \ldots, \boldsymbol{x}_n^{\mathsf{T}})$: structure function of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system

- $\beta_z(m,n)$: the number of path sets of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system with exactly z failed components
- $\leq_{st}$: usual stochastic ordering
- $\leq_{sp}$: stochastic precedence ordering
- $\mathbb{E}(Z)$: expected value of random variable Z
- $\pi(i,j)$: index of a component assigned to position (i,j) ($i = 1,2,\ldots,m$ and $j = 1,2,\ldots,n$)
- $\Pi = (\pi(i,j))_{1\leq i\leq m, 1\leq j\leq n}$: arrangement of mn components that assigned component $\pi(i,j)$ to position (i,j)
- $\boldsymbol{p} = (p_1, p_2, \ldots, p_{mn})$: vector of reliabilities of mn components (in CAP), where $p_1 < p_2 < \cdots < p_{mn}$ without loss of generality
- $R^L((r,s),(m,n),\boldsymbol{p};\Pi)$: reliability of the Lin/(r,s)/(m,n):F system under an arrangement Π with component reliabilities $\boldsymbol{p}$
- Ω: set of all arrangements
- Π^*: optimal arrangement, which is defined as

$$\Pi^* \equiv \arg\max_{\Pi\in\Omega} R^L((r,s),(m,n),\boldsymbol{p};\Pi)$$

Chapter 1

Introduction

1.1 Background

Nowadays, there are many sophisticated systems, such as aircraft, space shuttles, telecommunication networks, robots, and manufacturing facilities [1]. Reliability is a critical metric of system performance and is defined as the probability that a component or system will perform its required function under given conditions for a stated time interval [2]. This definition contains three elements. The first element is "function," and this should be clearly defined to quantify reliability. A failure means that a component or system cannot perform its function satisfactorily. The second element is "conditions," and reliability depends on operating conditions. In other words, a system or component might be reliable under certain conditions but become unreliable under more severe conditions. Finally, reliability usually varies with "time"; thus, the time to failure is a random variable.

An overview of historical developments in reliability engineering is presented as follows. The theory of reliability engineering has its roots in research on performance evaluation of various military electronic systems during the Second World War. In such an unusual situation, the systems were apt to be less reliable because they were used in severe environments. The tendency was accelerated due to the increased complexity of the systems. For these reasons, many electric systems could not perform their functions due to frequent failures, which had a risk to cause serious accidents. To overcome this problem, several new approaches that use probability theory and statistics was introduced. Today, this is called *reliability engineering*. Over the years, many

developments in reliability engineering have been obtained. The reliability engineering is summarized was Birolini [2], and the basic concepts used in the reliability engineering were introduced in Barlow and Proschan [3].

Modern society has been increasingly depending on various systems that have consistently enriched our lives. However, an absolute guarantee cannot be made that such systems will perform their specific functions satisfactorily throughout their intended life spans. System failure is an unavoidable event, and it can occur under various circumstances. The consequences of such failures could significantly impact our lives, as in the cases of nuclear explosions, airplane crashes, and electrical network shutdowns [4]. To prevent accidents and reduce the causes of failure, these systems should be reliable. The appropriate evaluation and enhancement of the reliability of such systems are critical to ensuring that they can meet their design requirements. A reliability study of practical systems provides helpful insights into industrial manufacturing and contributes to the stable operation of practical systems.

Reliability theory focuses on evaluating the reliability of a component or system and enhancing the system reliability during the design and operation phases. These objectives are generally accomplished according to the following phases:

(1) collecting lifetime data on components using a reliability test;

(2) estimating the reliability of individual components based on the collected lifetime data;

(3) computing the reliability of a system from the component reliabilities;

(4) optimizing reliability in the design phase;

(5) optimizing reliability in the operation phase.

In the reliability theory, one of the key problems is to accurately determine the reliability of a system from the knowledge of its component reliabilities. System reliability may be used to solve some optimization problems in the design or operation phase. In addition, system reliability could be used as a decision-making factor when choosing between design alternatives. Thus, this thesis considers Phase (3), the reliability evaluation. In this thesis, it is assumed that the mission time of the system and its components are implicitly specified, which means that we deal primarily with system and component reliabilities instead of their reliability functions of time.

A system should satisfy the performance requirements desired by the customer. System reliability plays a major role in determining system performance, wherein systems are expected to be reliable. As described in Lad *et al.* [5], reliability has become a mandatory requirement for customer satisfaction and is playing an increasing role in determining the competitiveness of products (systems). Hence, it is necessary to design systems with high reliability, leading to the study of reliability optimization, in which enhancing system reliability is the main objective. Therefore, this thesis also focuses on Phase (4), optimal design.

The primary objective of reliability optimization is to increase system reliability. To realize that, there are five general options [6]:

(i) increasing the reliability of each component in the system;

(ii) providing redundant components in parallel;

(iii) using a combination of enhanced component reliability and redundant components provisioned in parallel;

(iv) adjusting the system parameters;

(v) reassigning the exchangeable components.

The first option, increasing the component reliability is achieved by the intrinsic technology of each field and cannot be conducted by reliability engineering. The second and third options are called the redundancy allocation problem (RAP) and the reliability-redundancy allocation problem (RRAP), respectively. The RAP aims to determine a system configuration by either maximizing system reliability under budget constraints or by minimizing system cost under constraints on the system reliability. The RRAP is the problem of maximizing system reliability through component reliability choices and component redundancy. Relevant work could be seen in Kuo and Prasad [7], Kuo and Wan [8], and Coit and Zio [9]. However, these options do not always yield satisfactory results [10]. Specifically, the RAP not only increases costs but also adds undesirable extra volume and weight to the system. Also, the RRAP may incur large costs to select component reliability, for example, because of the difficulty in production. Therefore, the other two approaches are considered in this thesis.

The fourth option is a problem of adjusting the system parameters, including a parameter for determining the system size. Finding the optimal parameters would be

helpful for designing a system with high reliability. The fifth option is called the component assignment problem (CAP). The functionally exchangeable components can have different reliabilities because of distinct brands, quality, ages, and/or extent of degradations [11]. Under the assumption that the components are functionally exchangeable, different arrangements of components result in different system reliabilities. Hence, the system reliability may be improved by optimally assigning components to positions in the system.

1.2 System Reliability Modeling

Many modern systems are large and complicated, yet they often have characteristic features and structures. The study of practical systems often utilizes simplified system models based on these characteristic features and structures.

1.2.1 Consecutive-k System

In the reliability theory, literature has focused on different types of system models. Typical system models are a series system and a parallel system; the series system fails if and only if at least one component fails, whereas the parallel system fails if and only if all the components fail. Besides, k-out-of-n:F system, which generalizes the series and parallel systems, fails if and only if at least k of the n components fail. These systems are relatively simple and quite general but can be applied to a variety of problems.

The state of k-out-of-n:F systems, including series and parallel systems, depends only on the number of failed components, but not on the positions of failed components. However, it is essential to consider the positions of failed components. In a practical situation, a system exists such that a cluster of failed components causes a system failure, and we should consider the positions of failed components in such a system. Practical examples of such systems will be given later. In 1980, Kontoleon [12] first studied such a system, and subsequently, Chiang and Niu [13] formally named it "consecutive-k-out-of-n:F system." Over the past four decades, because of theoretical development and practical applications in the research field of reliability, much effort has been devoted to studying the reliability of this system. Here, a consecutive-k system is defined as a general term for systems such that a cluster of failed components causes a system failure. This thesis uses the term "consecutive-k system," but it is also

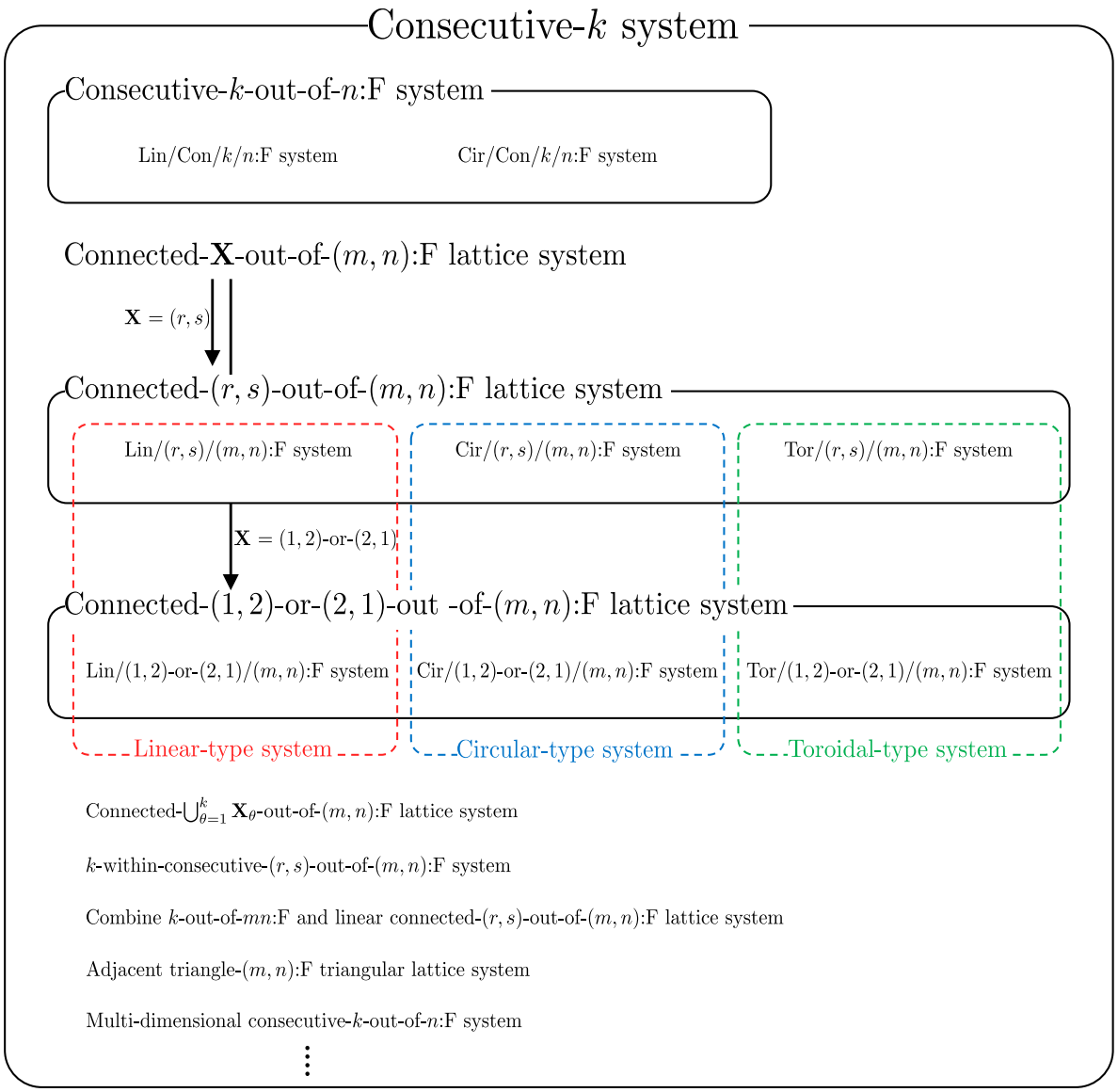


Figure 1.1: Family of the consecutive-k systems.

called a consecutive-type system [14] or a consecutive multi-unit system [15]. Figure 1.1 displays the family of the consecutive-k systems (each system in this figure is explained later).

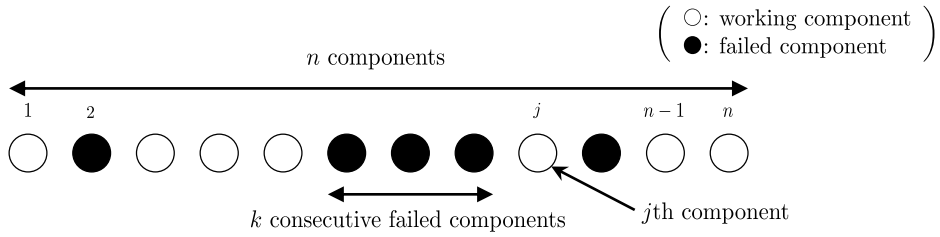


Figure 1.2: Lin/Con/k/n:F system.

1.2.2 Consecutive-k-out-of-n:F System

This subsection introduces a mathematical model to describe the consecutive-k-out-of-n:F system since this system is the basis of the consecutive-k system. This system can be classified into a linear-type and circular-type systems. Throughout this thesis, we use the following nomenclature for the consecutive-k-out-of-n:F systems:

- Lin/Con/k/n:F system: linear consecutive-k-out-of-n:F system
- Cir/Con/k/n:F system: circular consecutive-k-out-of-n:F system

Also, Lin and Cir/Con/k/n:F systems are collectively called a consecutive-k-out-of-n:F system in this thesis.

Before describing the above systems, we define some common notations. For $j = 1, 2, \ldots, n$, we define the random variable Z_j by

$$Z_j = \begin{cases} 0, & \text{if the } j\text{th component works,} \\ 1, & \text{if the } j\text{th component fails.} \end{cases} \tag{1.1}$$

Letting p_j denote the reliability of the jth component from the left-hand side of a Lin/Con/k/n:F system, namely, $p_j = \Pr\{Z_j = 0\}$ for $j = 1, 2, \ldots, n$, then we have

$$\boldsymbol{p} = (p_1, p_2, \ldots, p_n). \tag{1.2}$$

Lin/Con/k/n:F system

This system consists of n linearly ordered components and fails if and only if at least k consecutive components fail as shown in Fig. 1.2. The reliability of the Lin/Con/k/n:F

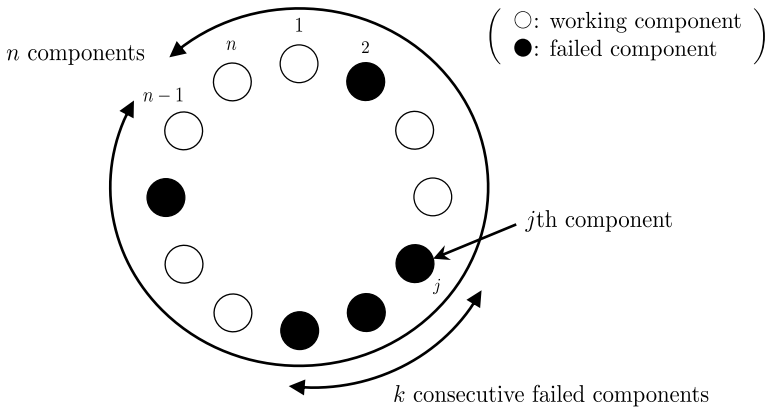


Figure 1.3: Cir/Con/k/n:F system.

system with component reliabilities $\boldsymbol{p}$, denoted by $R^L(k, n, \boldsymbol{p})$, is defined by

$$R^L(k, n, \boldsymbol{p}) = \Pr\left\{ \bigcap_{l=1}^{n-k+1} \left\{ \prod_{v=l}^{l+k-1} Z_v = 0 \right\} \right\}. \tag{1.3}$$

Cir/Con/k/n:F system

This system consists of n circularly ordered components and fails if and only if at least k consecutive components fail as shown in Fig. 1.3. The reliability of the Cir/Con/k/n:F system with component reliabilities $\boldsymbol{p}$, denoted by $R^C(k, n, \boldsymbol{p})$, is defined by

$$R^C(k, n, \boldsymbol{p}) = \Pr\left\{ \bigcap_{l=1}^{n} \left\{ \prod_{v=l}^{l+k-1} Z_v = 0 \right\} \right\}, \tag{1.4}$$

where for $j = 1, 2, \ldots, n$, Z_j was defined in Eq. (1.1), and for $j = n{+}1, n{+}2, \ldots, n{+}k{-}1$, $Z_j = Z_{j-n}$ for convenience.

For the applications of these systems, we give some practical examples.

Example 1. Oil Pipeline System (Chiang and Niu [13])

Let us consider an oil pipeline system consisting of n pump stations, and it transports oil from point A to point B. The pump stations are equally spaced between points A and B. Each pump station can transport the oil to the k next stations. If one pump station fails, the flow of the oil would not be interrupted because the next stations could

carry the load. However, when at least k consecutive pump stations fail, the flow is interrupted, and the system fails. Thus, such an oil pipeline system can be modeled as a Lin/Con/k/n:F system.

Example 2. Microwave Stations of a Telecom Network (Chiang and Niu [13])

Consider a sequence of n microwave stations designed to transmit information from place A to place B. Assume that the microwave stations are equidistantly spaced between places A and B and each station is able to transmit information a distance up to k microwave stations. Whenever there exist no k consecutive failed microwave stations, the signal flow is not interrupted, and the signal reaches the receiver. However, it can be readily verified that if at least k consecutive microwave stations fail, the communication between places A and B will be interrupted, and hence, the whole system fails in this situation. Thus, such a telecommunication system can be modeled as a Lin/Con/k/n:F system.

Example 3. Road Lights in a Highway (Dafnis *et al.* [16], Peng *et al.* [17])

Let us consider a highway, which needs lighting during night-time. That is why road lights are installed at regular intervals, e.g., 20 meters. When k consecutive lights are not working, sufficient light does not exist in a particular area, which may affect the traffic. Thus, such road lights can be modeled as a Lin/Con/k/n:F system.

Example 4. Production Monitoring System (Zhao *et al.* [18])

Let us consider a production monitoring system, which has n monitors. Each monitor can observe k units, and they are equally spaced along the production line. If one monitor fails, the neighboring working monitors can observe the units that the failed monitor cannot observe. However, if k consecutive monitors fail, the monitoring system will have a blind area, which means that the whole system is in failure. Thus, such a production monitoring system can be modeled as a Lin/Con/k/n:F system.

Example 5. Closed Recurring Water-Cooling System (Shen and Cui [19])

Consider a closed recurring water-cooling system with n water pumps in a thermoelectric plant. The water and steam expelled from a turbine are pumped to a cooling tower through n water pumps. Because of the disparity of the water level, the cooled water can be sent back to the boiler to produce steam for the turbine again. For such a recurring water-cooling system, each pump must be powerful enough to pump water and steam to at least the next k consecutive pumps. Failure of at least k consecutive

pumps will interrupt the flow, which leads to the failure of the whole system. Thus, such a closed recurring water-cooling system can be modeled as a Cir/Con/k/n:F system.

For a comprehensive survey of reliability studies of a consecutive-k-out-of-n:F system and its generalized systems, see [14, 20–25].

1.2.3 Connected-X-out-of-(m, n):F Lattice System

A consecutive-k-out-of-n:F system can be regarded as a one-dimensional system, and this system can be extended to two- or d-dimensional versions ($d \geq 3$). The two-dimension consecutive-k-out-of-n:F system was introduced first by Salvia and Lasher [26] in 1990. After then, Boehme *et al.* [27] defined a more general two-dimension consecutive-k-out-of-n:F system, i.e., connected-**X**-out-of-(m, n):F lattice system. This system consists of mn components arranged into a rectangular pattern with m rows and n columns. Here, **X** denotes the form of failed components that triggers system failure and is called "failure pattern" throughout this thesis. To put it differently, the system fails if and only if the system has at least one subset represented by failure pattern **X** in which all components fail. This system belongs to the consecutive-k system because this system has the feature that the system state is determined by the states of consecutive components. As typical examples of the connected-**X**-out-of-(m, n):F lattice system, we consider the following two systems:

- connected-(r, s)-out-of-(m, n):F lattice system
- connected-$(1, 2)$-or-$(2, 1)$-out-of-(m, n):F lattice system

Studying the above systems can provide a basis for research on the general connected-**X**-out-of-(m, n):F lattice system.

First, we define a connected-(r, s)-out-of-(m, n):F lattice system. If **X** represents (r, s), the connected-**X**-out-of-(m, n):F lattice system becomes a connected-(r, s)-out-of-(m, n):F lattice system. The linear-type and circular-type systems have been studied. As a natural extension of these systems, a toroidal-type system can be considered, but this system has not been studied so far. Consequently, the connected-(r, s)-out-of-(m, n):F lattice system can be classified into a linear-type, circular-type, and toroidal-type systems. Throughout this thesis, we use the following nomenclature for the connected-(r, s)-out-of-(m, n):F lattice systems:

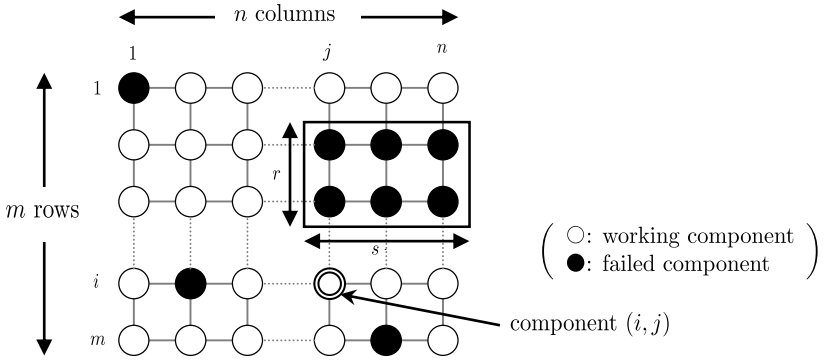


Figure 1.4: Lin/(r, s)/(m, n):F system.

- Lin/(r, s)/(m, n):F system [27]:
  linear connected-(r, s)-out-of-(m, n):F lattice system
- Cir/(r, s)/(m, n):F system [27]:
  circular connected-(r, s)-out-of-(m, n):F lattice system
- Tor/(r, s)/(m, n):F system:
  toroidal connected-(r, s)-out-of-(m, n):F lattice system

Also, Lin, Cir, and Tor/(r, s)/(m, n):F systems are collectively called a connected-(r, s)-out-of-(m, n):F lattice system in this thesis.

Before describing the above systems, we define some common notations. The component located at the ith row and the jth column is denoted by component (i, j) as shown in Fig. 1.4 in a linear-type system. For $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, we define the random variable Z_{ij} by

$$Z_{ij} = \begin{cases} 0, & \text{if component } (i, j) \text{ works,} \\ 1, & \text{if component } (i, j) \text{ fails.} \end{cases} \tag{1.5}$$

Letting p_{ij} denote the reliability of component (i, j), namely, $p_{ij} = \Pr\{Z_{ij} = 0\}$, for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, then we have

$$P = (p_{ij})_{1 \le i \le m, 1 \le j \le n}\,. \tag{1.6}$$

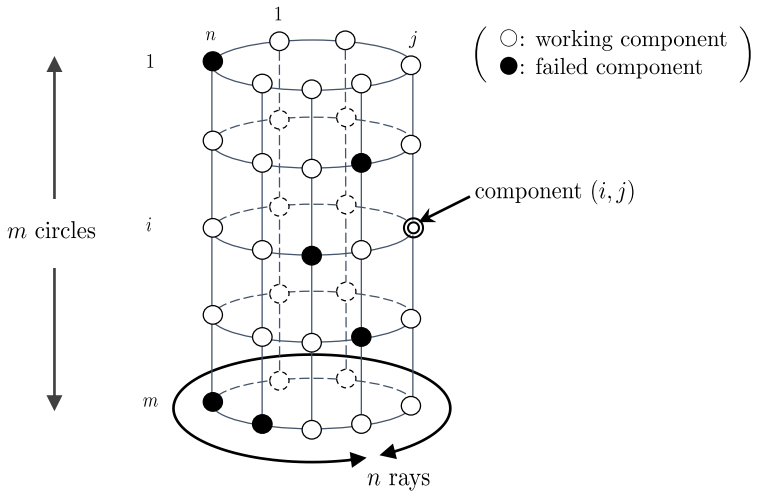


Figure 1.5: Cir/(r, s)/(m, n):F system.

Lin/(r, s)/(m, n)**:F system**

This system consists of mn components arranged as an (m, n) matrix and fails if and only if the system has an (r, s) sub-matrix where all components fail. In a linear-type system, (r, s) sub-matrix represents a matrix that consists of r rows and s columns. Figure 1.4 depicts the Lin/(r, s)/(m, n):F system. The reliability of the Lin/(r, s)/(m, n):F system with component reliabilities P, denoted by $R^L((r, s), (m, n), P)$, is defined by

$$R^L((r, s), (m, n), P) = \Pr\left\{ \bigcap_{k=1}^{m-r+1} \bigcap_{l=1}^{n-s+1} \left\{ \prod_{u=k}^{k+r-1} \prod_{v=l}^{l+s-1} Z_{uv} = 0 \right\} \right\}. \tag{1.7}$$

Cir/(r, s)/(m, n)**:F system**

This system forms a cylinder in which both ends of a Lin/(r, s)/(m, n):F system are connected as shown in Fig. 1.5. For a circular-type system, the component located at the ith circle and the jth ray is denoted by component (i, j). This system fails if and only if the system has an (r, s) sub-matrix where all components fail. In a circular-type system, (r, s) sub-matrix represents a matrix that consists of r circles and s rays. Figure 1.6 depicts the failures of the Cir/$(2, 3)$/$(5, 6)$:F systems as two-dimensional grids. In Fig. 1.6 (b), since the 1st and 6th rays are connected, this system has a $(2, 3)$ sub-matrix where all components fail. The reliability of the Cir/(r, s)/(m, n):F system

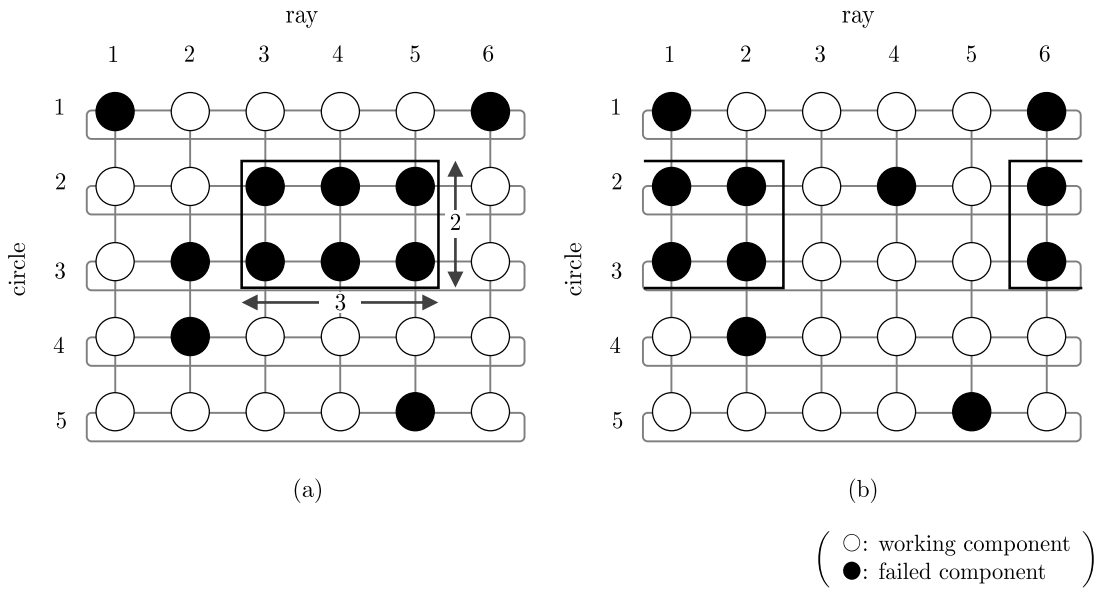


Figure 1.6: Failures of the Cir/(2, 3)/(5, 6):F systems.

with component reliabilities P, denoted by $R^C((r,s),(m,n),P)$, is defined by

$$R^C((r,s),(m,n),P) = \Pr\left\{\bigcap_{k=1}^{m-r+1}\bigcap_{l=1}^{n}\left\{\prod_{u=k}^{k+r-1}\prod_{v=l}^{l+s-1} Z_{uv} = 0\right\}\right\}, \tag{1.8}$$

where, for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, Z_{ij} was defined in Eq. (1.5) and for $i = 1, 2, \ldots, m$ and $j = n+1, n+2, \ldots, n+s-1$, $Z_{ij} = Z_{i,j-n}$ for convenience.

Tor$/(r,s)/(m,n)$:F system

This system has mn components, which are deployed at the intersections of m circles and n rings as shown in Fig. 1.7. For a toroidal-type system, the component located at the ith circle and the jth ring is denoted by component (i,j). This system fails if and only if the system has an (r,s) sub-matrix where all components fail. In a toroidal-type system, (r,s) sub-matrix represents a matrix that consists of r circles and s rings. The failures of the Tor/(2, 3)/(5, 6):F systems are pictorially illustrated in Fig. 1.8 as two-dimensional grids. The system in Fig. 1.8 has a $(2,3)$ sub-matrix where all components fail because the 1st and 5th circles are connected. The reliability of the Tor$/(r,s)/(m,n)$:F system with component reliabilities P, denoted by

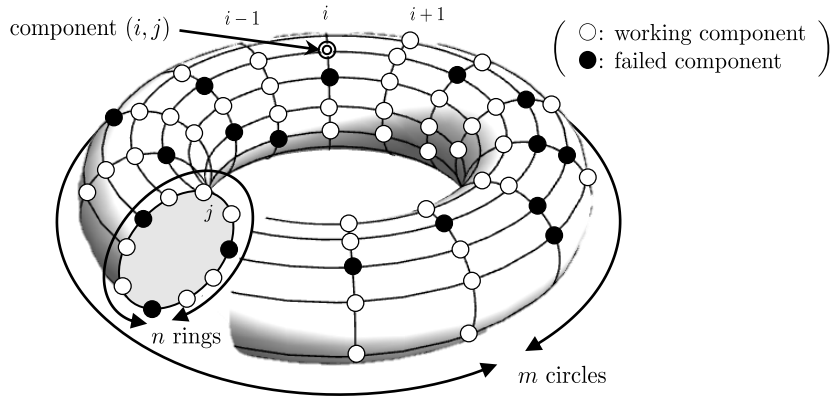


Figure 1.7: Tor/(r,s)/(m,n):F system.

$R^T((r,s),(m,n),P)$, is defined by

$$R^T((r,s),(m,n),P) = \Pr\left\{\bigcap_{k=1}^{m}\bigcap_{l=1}^{n}\left\{\prod_{u=k}^{k+r-1}\prod_{v=l}^{l+s-1} Z_{uv} = 0\right\}\right\}, \tag{1.9}$$

where, for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, Z_{ij} was defined in Eq. (1.5) and for the other cases,

$$Z_{ij} = \begin{cases} Z_{i,j-n}, & \text{if } 1 \le i \le m \text{ and } n+1 \le j \le n+s-1, \\ Z_{i-m,j}, & \text{if } m+1 \le i \le m+r-1 \text{ and } 1 \le j \le n, \\ Z_{i-m,j-n}, & \text{if } m+1 \le i \le m+r-1 \text{ and } n+1 \le j \le n+s-1, \end{cases}$$

for convenience.

Second, we define a connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F lattice system. If $\mathbf{X}$ represents $(1,2)$-or-$(2,1)$, the connected-$\mathbf{X}$-out-of-(m,n):F lattice system becomes a connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F lattice system. This system can also be classified into a linear-type, circular-type, and toroidal-type systems similarly to the connected-(r,s)-out-of-(m,n):F lattice system. Throughout this thesis, we use the following nomenclature for the connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F lattice system:

- Lin/$(1,2)$-or-$(2,1)$/(m,n):F system [27]:
  linear connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F lattice system
- Cir/$(1,2)$-or-$(2,1)$/(m,n):F system [27]:

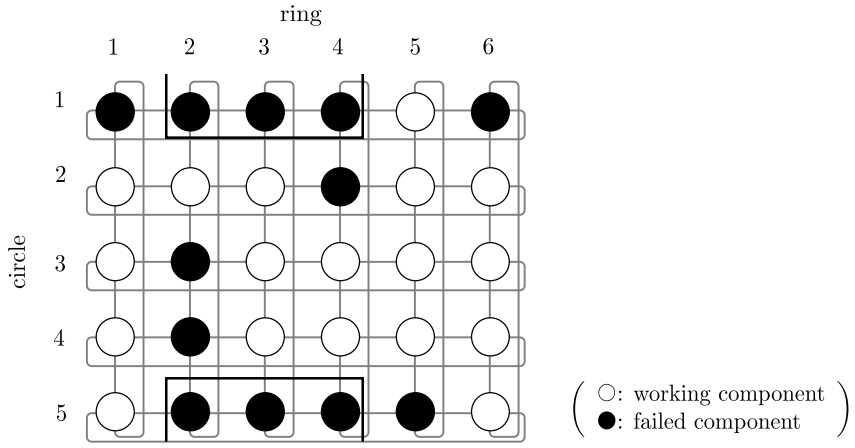


Figure 1.8: Failure of the Tor/$(2,3)$/$(5,6)$:F system.

circular connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F lattice system

- Tor/$(1,2)$-or-$(2,1)$/(m,n):F system:
  toroidal connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F lattice system

Also, Lin, Cir, and Tor/$(1,2)$-or-$(2,1)$/(m,n):F systems are collectively called a connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F lattice systems in this thesis.

Before describing the above systems, we define some common notations. The random variables $Y_{ij}^{(1,2)}$ and $Y_{ij}^{(2,1)}$ are defined, for $i = 1,2,\ldots,m$ and $j = 2,3,\ldots,n$, as

$$Y_{ij}^{(1,2)} = Z_{i,j-1} \times Z_{ij}, \tag{1.10}$$

and, for $i = 2,3,\ldots,m$ and $j = 1,2,\ldots,n$,

$$Y_{ij}^{(2,1)} = Z_{i-1,j} \times Z_{ij}. \tag{1.11}$$

Lin/$(1,2)$-or-$(2,1)$/(m,n):F system

This system consists of mn components arranged as an (m,n) matrix. It fails if and only if the system has the two adjacent failed components in a row or a column. Note that $Y_{ij}^{(1,2)}$ (or $Y_{ij}^{(2,1)}$) takes one if there exists at least one failure pattern $(1,2)$ (or $(2,1)$), and otherwise zero. The failure criterion of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system is pictorially explained in Fig. 1.9. Fig. 1.9 (a) shows that the failed compo-

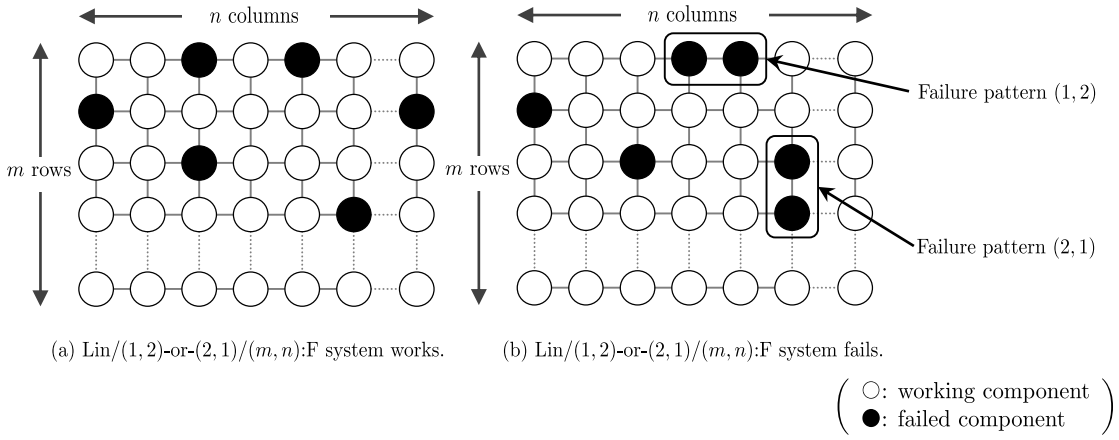


Figure 1.9: Working and failed Lin/$(1,2)$-or-$(2,1)/(m,n)$:F systems.

nents are located sparsely. Because the working components adjacent to each failed component can complement the failed one, the system works normally. In contrast, Fig. 1.9 (b) has two adjacent failed components, and thus, the system fails. The reliability of the Lin/$(1,2)$-or-$(2,1)/(m,n)$:F system with component reliabilities P, denoted by $R^L((m,n),P)$, is defined by

$$R^L((m,n),P) = \Pr\left\{\prod_{i=1}^{m}\prod_{j=2}^{n}\left(1-Y_{ij}^{(1,2)}\right) \times \prod_{i=2}^{m}\prod_{j=1}^{n}\left(1-Y_{ij}^{(2,1)}\right) = 1\right\}. \tag{1.12}$$

Cir/$(1,2)$**-or-**$(2,1)/(m,n)$**:F system**

This system consists of mn components, which are arranged in a lattice on the surface of a cylinder. It fails if and only if the system has the two adjacent failed components in a circle or ray. The reliability of the Cir/$(1,2)$-or-$(2,1)/(m,n)$:F system with component reliabilities P, denoted by $R^C((m,n),P)$, is defined by

$$R^C((m,n),P) = \Pr\left\{\prod_{i=1}^{m}\prod_{j=1}^{n}\left(1-Y_{ij}^{(1,2)}\right) \times \prod_{i=2}^{m}\prod_{j=1}^{n}\left(1-Y_{ij}^{(2,1)}\right) = 1\right\}, \tag{1.13}$$

where, for $i = 1,2,\ldots,m$ and $j = 1,2,\ldots,n$, Z_{ij} was defined in Eq. (1.5), and for $i = 1,2,\ldots,m$, $Z_{i0} = Z_{in}$ for convenience.

Tor$/(1,2)$**-or-**$(2,1)/(m,n)$**:F system**

This system consists of mn components, which are arranged in a lattice on the surface of a torus. It fails if and only if the system has the two adjacent failed components in a circle or ring. The reliability of the Tor$/(1,2)$-or-$(2,1)/(m,n)$:F system with component reliabilities P, denoted by $R^T((m,n),P)$, is defined by

$$R^T((m,n),P) = \Pr\left\{\prod_{i=1}^{m}\prod_{j=1}^{n}\left(1-Y_{ij}^{(1,2)}\right)\times\prod_{i=1}^{m}\prod_{j=1}^{n}\left(1-Y_{ij}^{(2,1)}\right)=1\right\}, \tag{1.14}$$

where, for $i = 1,2,\ldots,m$ and $j = 1,2,\ldots,n$, Z_{ij} was defined in Eq. (1.5), for $i = 1,2,\ldots,m$, $Z_{i0} = Z_{in}$, and for $j = 1,2,\ldots,n$, $Z_{0j} = Z_{mj}$ for convenience.

The following illustrative examples for connected-**X**-out-of-(m,n):F lattice systems are presented as follows:

Example 6. Wireless Sensor Network System (Lin *et al.* [28])

Let us consider a wireless sensor network (WSN). Current and potential applications of sensor networks include: military sensing, physical security, air traffic control, traffic surveillance, industrial and manufacturing automation, distributed robotics, environment monitoring, and building and structures monitoring [29]. Their wide range of applications is based on the possible use of various sensor types (i.e. thermal, visual, seismic, acoustic, radar, magnetic) in order to monitor a wide variety of conditions (e.g. temperature, object presence and movement, humidity, pressure, noise levels) [30]. For simplicity, we suppose that the sensors in this system are deployed in a lattice structure. These sensors can detect the information of interest with limited coverage range, and the system has overlapped coverage ranges among different sensors. If one sensor fails, we cannot obtain the information within its coverage range; however, the neighboring working sensors can provide the necessary information. On the contrary, if a cluster of failed sensors exists, it may cause a blind spot in the system; consequently, the entire system will be unable to function normally. Thus, such a WSN can be modeled as a Lin$/(r,s)/(m,n)$:F system. Depending on the coverage range of each sensor, a Lin$/(1,2)$-or-$(2,1)/(m,n)$:F system could be used to evaluate this system. Liu [31] discussed this application in detail.

Example 7. Phased Array Radar (Yuge *et al.* [32], Lin *et al.* [33])

Consider a phased array radar. The T/R modules are placed at the coordinate point

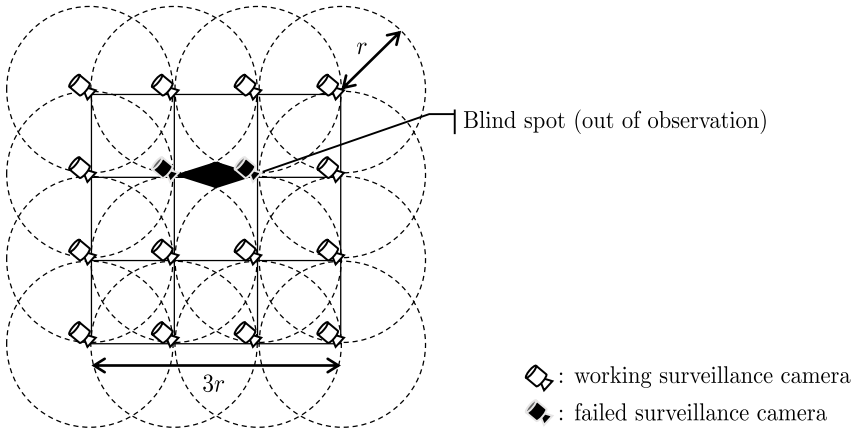


Figure 1.10: Example of a supervision system.

with m rows and n columns. Because of overlapping coverage of the T/R modules, the radar can continue working when some of the modules are failed sparsely. However, if an (r, s) sub-matrix of failed components occurs, a blind spot exists, which would lead the malfunction of the radar. As a result, the whole system is regarded as a failure. Thus, such a phased array radar can be modeled as a Lin/(r, s)/(m, n):F system.

Example 8. Pattern Detection (Salvia and Lasher [26])

The presence of cancer is diagnosed by reading an X-ray picture represented by an (m, n) matrix of cells. Unless the local density of cancer cells is sufficiently large, i.e., there are (r, s) sub-matrix of cells, the radiologist might not detect their presence. A Lin/(r, s)/(m, n):F system can be used in such a pattern detection.

Example 9. Supervision System (Boehme *et al.* [27])

Let us consider a supervision system with mn surveillance cameras arranged into m rows and n columns to monitor a specified area. Each surveillance camera supervises an area with radius r, and all surveillance cameras are the same. Even if the failed surveillance cameras are located sparsely, the system can work normally. This is because the working surveillance cameras adjacent to each failed surveillance camera can complement the failed one, and the whole area can be monitored. However, two adjacent failed surveillance cameras will cause a blind spot, which is out of observation. Consequently, the whole system fails. Thus, such a supervision system can be modeled as a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system. Figure 1.10 graphically illustrates an example of a supervision system, which is modeled as a Lin/$(1, 2)$-or-$(2, 1)$/$(4, 4)$:F system.

Noguchi *et al.* [34] discussed this application in detail.

Likewise, a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system can be used to model a lighting (sprinkler) system with lights (sprinklers) arranged into m rows and n columns.

Example 10. Liquid Crystal Screen (Akiba and Yamamoto [35])

Let us consider a liquid crystal screen, which, for example, is used in a laptop computer. This screen consists of a number of liquid crystal cells arranged in a matrix configuration. Even if some liquid crystal cells fail sparsely, we can get specific information from the screen. However, if there exists a cluster of the failed liquid crystal cells, information cannot accurately be obtained. Accordingly, the whole system is regarded as a failure. Such a liquid crystal screen can be modeled as a Lin/(r,s)/(m,n):F system.

Besides, when considering a liquid crystal screen with 360-degree wide-area visual angle, we can use a Cir/(r,s)/(m,n):F system to model this screen.

Example 11. Alarm System (Boehme *et al.* [27], Makri and Psillakis [36])

Consider an alarm system consisting of sensors arranged in a lattice on the surface of a cylindrical object, e.g., a reactor, where this system has m circles parallel to the reactor's base and each circle has n sensors. It is supposed that each sensor can measure the temperature. This system fails whenever there exists an (r,s) sub-matrix where all sensors fail. Thus, such a cylindrical alarm system can be modeled as a Cir/(r,s)/(m,n):F system. Depending on the measuring range of each sensor, a Cir/$(1,2)$-or-$(2,1)$/(m,n):F system could be used to evaluate this alarm system.

Also, let us consider an alarm system consisting of sensors arranged in a lattice on the surface of a toroidal object, e.g., toroidal storage tank and neutron accelerator. Here, a toroidal object has m vertical circles and n horizontal circles, and the sensors are deployed at their intersections (Fig. 1.11). This system fails whenever there exists an (r,s) sub-matrix where all sensors fail, and hence, such a toroidal alarm system can be modeled as a Tor/(r,s)/(m,n):F system. Depending on the measuring range of each sensor, a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system could be used to evaluate this alarm system.

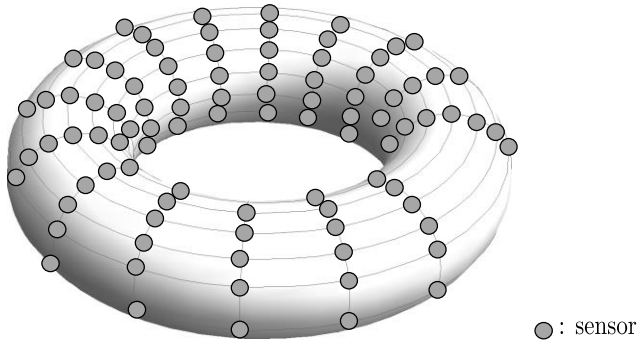


Figure 1.11: Example of a toroidal alarm system.

1.2.4 Other Connected-X-out-of-(m, n):F Lattice Systems

Various types of consecutive-k systems have been developed. Some of the representative systems are listed as follows:

Connected-$\bigcup_{\theta=1}^{k} \mathbf{X}_\theta$-out-of-$(m, n)$:F lattice system [37]

This system consists of mn components arranged as an (m, n) matrix and has multiple failure criteria. Specifically, this system fails if and only if the system has one of the $\mathbf{X}_\theta$s where all components fail for $\theta = 1, 2, \ldots, k$, where $\mathbf{X}_\theta$ represents a failure pattern. If $k = 1$ and $\mathbf{X}_1 = (r, s)$, then it is equivalent to a Lin/(r, s)/(m, n):F system; if $k = 2$, $\mathbf{X}_1 = (1, 2)$, and $\mathbf{X}_2 = (2, 1)$, then it is equivalent to a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system.

k-within-consecutive-(r, s)-out-of-(m, n):F system [38]

This system consists of mn components arranged as an (m, n) matrix and fails if and only if there is at least one cluster of size $r \times s$ such that the number of the failed components within this cluster is at least k. If $k = rs$, then it is equivalent to a Lin/(r, s)/(m, n):F system.

Combine k-out-of-mn:F and linear connected-(r, s)-out-of-(m, n):F lattice system [39]

This system consists of mn components arranged as an (m, n) matrix and fails if and only if at least k components fail in the system, or there is an (r, s) sub-matrix where all components fail. If $k = mn$, then it is equivalent to a Lin/(r, s)/(m, n):F system.

Adjacent triangle-(m, n):F triangular lattice system [40]

This system consists of mn components arranged on a triangular lattice and fails if the system has three adjacent failed components that are the vertices of a triangle.

In addition, as a natural generalization, the concept of the connected-**X**-out-of-(m, n):F lattice system can be extended to space, that is to say, this system can be extended to multi-dimensional cases. These systems are listed below:

- Three-dimensional consecutive-k-out-of-n:F system [41]
- Three-dimensional consecutive-(r_1, r_2, r_3)-out-of-(n_1, n_2, n_3):F system [42, 43]
- Conditional three-dimensional consecutive-(r, r, r)-out-of-(n, r, r):F system [44]
- d-dimensional consecutive-k-out-of-n:F system [45, 46]
- Three-dimensional adjacent triangle:F triangular lattice system [47]

For a comprehensive survey of reliability studies of connected-**X**-out-of-(m, n):F lattice systems and its generalized systems, see Kuo and Zuo [22] and Yamamoto and Akiba [48]. Recent developments can be found in Akiba *et al.* [49] and Cui and Dong [25].

1.3 General Problems for Consecutive-k Systems

In this section, we briefly take an overview of the general problems for consecutive-k systems, which are addressed in Chapters 2, 3, and 4 of this thesis. Recall that this thesis is devoted to Phases (3) and (4) in page 2.

(3) computing the reliability of a system from the component reliabilities;

(4) optimizing reliability in the design phase.

(3) Computing the reliability of a system from the component reliabilities

One of the traditional problems in the research field of reliability is known as the reliability evaluation, that is, to compute the system reliability when the component reliabilities are given. It is noteworthy that system reliability provides a useful tool

to solve some optimization problems in the design or operation phase. Therefore, the system reliability evaluation is a fundamental step in all reliability studies.

(3-1) Reliability Evaluation (Exact method)

In many situations, the exact system reliabilities are necessary, such as reliability optimization, component importance analysis [50]. For a small system, generally, the system reliability can be obtained by enumeration method, which is the simplest of all methods for computing system reliability. For example, let us consider a system with N components. This method enumerates 2^N possible system states that depend on the states of N components. These 2^N system states are identified for being a working or failed state of the system, and eventually, the system reliability can be obtained as a summation of the probabilities associated with the working states. Since this method needs to identify the 2^N possible system states, it is inefficient and time-consuming, and as a result, it may fail to obtain the system reliability within a reasonable time. Consequently, the practical application of this method is limited to relatively small systems. Therefore, many valuable methods for the reliability of consecutive-k systems have been developed.

Fundamental techniques for system reliability evaluation are pivotal decomposition, the inclusion-exclusion method, and the sum-of-disjoint-products method (see [22, Chapter 5]). In the case where the components are independent and identically distributed, most researchers have used the combinatorial approach for reliability evaluation. The combinatorial approach can find an explicit expression of system reliability as a function of common component reliability. In the case where the components are independent but non-identically distributed, two types of methods are common and popular for efficiently computing the reliability of the consecutive-k systems: the recursive equation approach and the finite Markov chain imbedding approach (FMCIA). For deriving a recursive equation, we condition some components in a system, and then, "the event that the system works" is decomposed into "several disjoint events that systems with the conditions work." In this manner, a recursive equation for computing the system reliability can be derived. This approach is called the event decomposition approach [22]. Many recursive equation approaches for computing the reliability of consecutive-k-out-of-n:F systems have been reported in the literature. For a Lin/Con/k/n:F system, the recursive equation was first used by Chiang and Niu [13], and subsequently, Hwang [51] derived the well-known recursive equation. Lin [52] derived a recursive equation for the

reliability of a Lin and Cir/Con/k/n:F systems consisting of the components with the same reliability.

The FMCIA has also been one of the most successful approaches for computing the reliability of a consecutive-k system. The basic idea of the FMCIA is to construct a finite state Markov chain to fit the discussed problems and then use the property of the Markov chain to give the solutions [53]. It was first employed by Fu [54] and Fu and Hu [55], and subsequently, this method was named "finite Markov chain imbedding" by Fu and Koutras [56]. In addition, Zhao and Cui [57] introduced the accelerated scan FMCIA, which can reduce the number of matrices required for computing the system reliability. As a result, the complexity of computation can be reduced, and the efficiency of computation can be enhanced.

The recursive equation approach and the FMCIA are also used to compute the reliability of connected-**X**-out-of-(m, n):F lattice systems. For a Lin/(r, s)/(m, n):F system, Yamamoto and Miyakawa [58] and Noguchi *et al.* [59] derived the recursive equations. Zhao *et al.* [60] employed the FMCIA to compute the reliability of the Lin/(r, s)/(m, n):F system. For a Cir/(r, s)/(m, n):F system, Yamamoto and Miyakawa [61] and Yamamoto and Akiba [62] derived the recursive equations. For a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system, Higashiyama [63], and Yamamoto *et al.* [64] derived the recursive equations. For a Cir/$(1, 2)$-or-$(2, 1)$/(m, n):F system, Higashiyama [65] derived the recursive equations.

(3-2) Reliability Evaluation (Bounds)

The above methods enable us to obtain the reliability of consecutive-k systems efficiently. For example, the method of Lin [52] can compute the reliability of a Lin/Con/k/n:F system with $k = 50$ and $n = 16384000$ in about 5 seconds. However, when we compute the reliability of a large connected-**X**-out-of-(m, n):F lattice system, the conventional methods require significant computation time and may fail to obtain the system reliability. To mitigate this difficulty, the upper and lower bounds for system reliability have also been used since they can be computed in a shorter computation time. A lower bound is the minimum system reliability, whereas an upper bound is an optimistic estimation of the system reliability.

Many upper and lower bounds for system reliability have been reported. First, we introduce the existing lower bounds for system reliability. Let us consider a coherent

system[1] with n components, and this system has N minimal cut sets as follows:

$$\mathbb{C} = \{C_1, C_2, \ldots, C_N\}. \tag{1.15}$$

where a set $C \subseteq \{1, 2, \ldots, n\}$ is called a cut set of a coherent system if the system fails when all the components in C fail. In particular, a cut set that does not contain other cut sets is called a minimal cut set. Let p_i denote the reliability of component i for $i = 1, 2, \ldots, n$, and Esary and Proschan [66] provided a well-known lower bound as follows:

$$LB_{EP} = \prod_{C \in \mathbb{C}} \left(1 - \prod_{i \in C} (1 - p_i) \right). \tag{1.16}$$

The lower bound LB_{EP} is obtained by considering a series system with $|\mathbb{C}|$ independent subsystems, and each subsystem has a parallel structure whose components are contained in each minimal cut. As argued in Boutsikas and Koutras [67], a combination of several minimal cut sets results in reducing the number of subsystems, and as a result, we can obtain a tighter lower bound. An arbitrary partition of $\mathbb{C}$ in Eq. (1.15) is denoted by $\{\mathbb{C}_1, \mathbb{C}_2, \ldots, \mathbb{C}_v\}$ for $v = 2, 3, \ldots, N$, such that $\mathbb{C}_s \subseteq \mathbb{C}$, for $s = 1, 2, \ldots, v$, $\mathbb{C}_s \cap \mathbb{C}_t = \emptyset$ when $s \neq t$, and $\bigcup_{s=1}^{v} \mathbb{C}_s = \mathbb{C}$. Let $R(\mathbb{C}_s)$ be the reliability of a system with the minimal cuts $\mathbb{C}_s$ for $s = 1, 2, \ldots, v$, and then the lower bound is

$$LB_{EP}^{(G)} = \prod_{s=1}^{v} R(\mathbb{C}_s). \tag{1.17}$$

Next, we introduce the existing upper bounds for system reliability. Yamamoto and Miyakawa [58] established an upper bound for the reliability of a Lin/(r, s)/(m, n):F system. This upper bound can be extended to any coherent system. An arbitrary partition of $\mathbb{C}$ in Eq. (1.15) is denoted by $\{\mathbb{K}_0, \mathbb{K}_1, \mathbb{K}_2, \ldots, \mathbb{K}_u\}$ for $u = 2, 3, \ldots, N$, such that $\mathbb{K}_s \subseteq \mathbb{C}$ for $s = 0, 1, \ldots, v$, $\mathbb{K}_s \cap \mathbb{K}_t = \emptyset$ when $s \neq t$, and $\bigcup_{s=0}^{u} \mathbb{K}_s = \mathbb{C}$. Besides, for any $A \in \mathbb{K}_s$ and $B \in \mathbb{K}_t$ ($s = 1, 2, \ldots, v$, $t = 1, 2, \ldots, v$, and $s \neq t$), $A \cap B = \emptyset$ holds. Let $R(\mathbb{K}_s)$ be the reliability of a system with the minimal cuts $\mathbb{K}_s$ for $s = 1, 2, \ldots, v$,

[1] A general system that satisfies the following conditions: (a) the state of the system depends on the states of all of its components; (b) the improvement of any component usually does not degrade the performance of the system.

and then an upper bound $UB_{YM}^{(G)}$ is given by

$$UB_{YM}^{(G)} = \prod_{s=1}^{u} R\left(\mathbb{K}_s\right). \tag{1.18}$$

Fu and Koutras [68] provided an upper bound for system reliability, and subsequently, Boutsikas and Koutras [67] established an upper bound ($UB_{FK}^{(G)}$) for the reliability of any coherent system, which can be obtained by generalizing the upper bound derived by Fu and Koutras [68].

Boutsikas and Koutras [67] derived some bounds that extended the existing bounds and comprehensively compared the derived and existing bounds. The result showed that the lower bound $LB_{EP}^{(G)}$ and the upper bound $UB_{FK}^{(G)}$ exhibit the highest performance among the bounds available in the literature so far. In addition, the upper and lower bounds for the reliability of the connected-**X**-out-of-(m, n):F lattice systems were reported by many papers (e.g., Yamamoto and Miyakawa [58], Yamamoto [69], Godbole [46], Boutsikas and Koutras [67], Hsieh and Chen [70], and Lin *et al.* [33].) Moreover, Beiu and Daus [71] reviewed several bounds for the reliability of the Lin/(r, s)/(m, n):F systems.

(3-3) Component Importance Analysis

According to Si *et al.* [72], the importance measure is defined as a criterion to evaluate the degree of the system performance changes due to the state changes of certain component. In reliability engineering, importance measures are generally used to rank the components in a system according to their contributions to the proper functioning of the entire system. For a review of the literature on importance measures, the reader is referred to Ohi [73] and Amrutkar and Kamalja [74]. Among all the importance measures, the Birnbaum importance (B-importance) is still one of the most widely studied importance measures in the context of consecutive-k systems [19]. The B-importance was introduced by Birnbaum [75], and it means the probability of the occurrence of the circumstances where the state of a component critically contributes to the system state [73]. The B-importance of component i, denoted by $I^B(i)$, can be computed by

$$I^B(i) = R(1_i, \boldsymbol{p}^i) - R(0_i, \boldsymbol{p}^i), \tag{1.19}$$

where $R(1_i, \boldsymbol{p}^i)$ ($R(0_i, \boldsymbol{p}^i)$) is the reliability of the system in which the reliability of

component i is 1 (0). In real applications, it is important to identify the most critical component of the system in designing and operating a reliable system. For instance, the B-importance can be used to determine which component should be preferentially maintained.

As for Lin/Con/k/n:F systems, Zhu *et al.* [50] and Kamalja [76] investigated B-importance patterns, which become a guide for designing and operating a reliable system [18]. Furthermore, Zhao *et al.* [60] provided the B-importance of the components in a Lin/(r, s)/(m, n):F system as a numerical example. Recently, Dui and Si [77] addressed reliability optimization based on the B-importance.

(4) Optimizing reliability in the design phase

Here, we describe optimizing reliability in the design phase. Recall that two approaches to enhancing system reliability are considered in this thesis.

(iv) adjusting the system's parameters;

(v) reassigning the exchangeable components.

(4-1) Stochastic Comparison

A stochastic comparison compares the lifetimes[2] of systems and can be used to appropriately determine the system parameters. The system signature is essential for establishing the stochastic comparison. The concept of system signature, which is a useful tool for analyzing coherent systems [78], was introduced by Samaniego [79]. Assume a coherent system consisting of N components, whose lifetimes are independent and identically distributed. Samaniego [78] defined the system signature as follows:

Definition 1.1 (Samaniego [78])**.** The system signature, denoted by $\boldsymbol{s}(N)$, is an N-dimensional probability vector whose ith element $s_i(N)$ is equal to the probability that the ith component failure causes the system to fail. In brief,

$$s_i(N) = \Pr\{T = T_{i:N}\}, \tag{1.20}$$

where T is the lifetime of the system and $T_{i:N}$ is the ith order statistic of the N component failure times, that is, the time of the ith component failure.

[2]A lifetime is defined as the time when a component or system fails.

Note that $\sum_{i=1}^{N} s_i(N) = 1$. By means of the system signature, the reliability function can be represented as

$$\Pr\{T > t\} = \sum_{i=1}^{N} s_i(N) \Pr\{T_{i:N} > t\}, \tag{1.21}$$

which shows the probability that the system's lifetime is larger than t, the probability that the system will survive beyond time t, or the probability that the system will fail after time t [22]. Samaniego [78] summarized the studies on the system signature.

In reliability engineering, the system signature has various theoretical applications, a few of which are described here. As noted above, the system signatures are used primarily to establish stochastic comparisons of some systems. The essential feature of this approach is that it is based on system structural invariant, which depends only on a structure function and does not depend on probabilistic properties (such as lifetime distribution) of the components [80]. The stochastic comparisons of the consecutive-k-out-of-n:F systems were reported by the existing studies [81–83].

The system signature can also be used to derive the mean time to failure (MTTF for short) and the mean number of failed components that are failed at the time of system failure [84], which can be easily derived once a system signature is obtained. This is advantageous when repeated analysis of a system is required, for example, in the analysis of real-time systems, importance measure analysis, and optimization problems [85]. Now, the system signature has become an indispensable tool in the reliability studies.

(4-2) Component Assignment Problem (Exact method)

Another reliability optimization problem that has been studied in the research field of reliability is the component assignment problem (CAP). This problem aims to find a component arrangement that maximizes system reliability, otherwise known as the optimal arrangement. A practical system is expected to be reliable and work satisfactorily for a long time. For the sake of enhancing the system reliability in the design phase, the problem of how and where to assign the components to the system is of great interest in the research field of reliability. Considerable work has been done on the CAP for coherent systems, particularly on Con/k/n:F systems, Kuo and Zhu [86] summarized the results of the CAP.

Optimal arrangements are classified into two types: invariant and variant opti-

mal arrangements. An invariant optimal arrangement is an optimal arrangement that depends only on the ordering of the values of component reliabilities, and a variant optimal arrangement is an optimal arrangement that depends on the numerical values of component reliabilities.

An attractive characteristic of an invariant optimal arrangement is that if we know only the ordering of the component reliabilities, it is possible to determine the optimal arrangement without any computations of the system reliability. Even if it is hard to obtain the exact values of component reliabilities, it may be relatively easy to get their orderings. For example, the reliability of components can be ordered according to their ages. Besides, in a practical situation, the component reliabilities are statistically estimated (Phase (2)). If the estimated values have an error or vary within a range, but the ordering of the estimated values is the same as the ordering of the actual component reliabilities, then we can obtain the invariant optimal arrangement. Zuo and Kuo [87] summarized the results available for the invariant optimal arrangement of Lin/Con/k/n:F systems and Cir/Con/k/n:F systems.

Most of the optimal arrangements depend on the values of component reliabilities, that is to say, they are variant optimal arrangements. Thus, we need to solve a combinatorial optimization problem for finding the optimal arrangement. An exhaustive enumeration method is a classic method for exactly solving the CAP. This method generates all possible component arrangements, evaluates system reliability under each component arrangement, and then finds the optimal arrangement. However, when the number of components is not small, this method encounters the combinatorial explosion problem, and thereby, it may fail to find the optimal arrangement within a reasonable time. Therefore, methods for solving the CAP are required and can be broadly classified into two approaches: the exact method and the approximate method.

In general, exact methods are used to find the optimal arrangement of small or middle-sized systems. A branch-and-bound (B&B) framework is a fundamental and widely used methodology for efficiently finding an optimal solution to the combinatorial optimization problem. The method systematically enumerates all possible candidates for the solution and finds the solution effectively by discarding candidates that will not be optimal. The operation term where non-optimal candidates are discarded is called "pruning" in this thesis. To enhance the speed of the B&B method, we should perform the pruning as many times as possible. In the CAP, basically, there are three pruning types as follows [88]:

(a) pruning based on the necessary condition that the optimal arrangement must satisfy;

(b) pruning for eliminating the equivalent arrangement;

(c) pruning based on system reliability.

Here, equivalent arrangements mean component arrangements with the same system reliability for any component reliabilities. The mathematical definition of the equivalent arrangements is given in Chapter 4. Besides, combining the B&B method with a particular recursive equation reduces the complexity of computation, which is explained in Chapter 4.

As for consecutive-k-out-of-n:F systems, Hanafusa and Yamamoto [88] developed a B&B-based algorithm for finding the optimal arrangement of a Lin/Con/k/n:F system. This algorithm incorporates (a) pruning based on the necessary condition, (b) pruning for eliminating the equivalent arrangement, and (c) pruning based on system reliability. It employs the recursive equation derived by Hwang [51] to compute the system reliability. Shingyochi and Yamamoto [89] developed a B&B-based algorithm for finding the optimal arrangement of a Cir/Con/k/n:F system. This algorithm also incorporates (a), (b), and (c) and employs the recursive equation [51].

As for connected-**X**-out-of-(m, n):F lattice systems, especially Lin/$(r, s)/(m, n)$:F system, Koutras *et al.* [90] established the necessary condition that the optimal arrangement must satisfy. Omura *et al.* [91] proposed an algorithm for finding the optimal arrangement, which is available for the case of $r = m$ and $s = 2$. It utilizes the invariant optimal arrangement of a Lin/Con/$2/n$:F system [92] to reduce the number of candidates for the optimal arrangement. For the case of $r = m - 1$ and $2s > n$, Omura *et al.* [93] developed a B&B-based algorithm for finding the optimal arrangement. This algorithm incorporates (a) pruning based on the necessary condition and (b) pruning for eliminating the equivalent arrangement.

(4-3) Component Assignment Problem (Approximate method)

Even if we use this efficient algorithm, it is time-consuming and sometimes impossible to find the optimal arrangement for a large system. On the other hand, the heuristic and meta-heuristic methods can solve large-scale problems more efficiently. Therefore, these are not able to certify the optimality of the solutions that they find. Generally,

they are classified into two groups: heuristics methods and meta-heuristics methods[3]. In solving the CAP for Lin/Con/k/n:F systems or Cir/Con/k/n:F systems, various heuristic and meta-heuristic methods have been reported over the past thirty years.

(1) Heuristic method [87, 94, 95].

(2) Meta-heuristic method, such as genetic algorithm [11, 96–99], simulated annealing [100, 101], and ant colony optimization [102, 103].

Some heuristic and meta-heuristic methods are based on the B-importance. The basic idea behind these methods is that the higher reliable components should be assigned to the position with a higher B-importance.

Heuristic methods have been reported. Zuo and Kuo [87] developed two ZK-type heuristics, namely, ZKA and ZKB heuristics. In the ZKA heuristic, starting from the least reliable component, if the B-importance of this component is larger than that of the next more reliable component, we exchange these two components. If this exchange improves the system reliability, we keep the arrangement in which these two components have been exchanged. Conversely, in the ZKB heuristic, starting from the least reliable component, if its B-importance is not the least important one among all the components with higher reliabilities, this component is exchanged with that of the least important. If this exchange improves the system reliability, we keep the arrangement in which these two components have been exchanged. Also, Lin and Kuo [94] proposed two LK-type heuristics, which are named as LKA and LKB heuristics. The LKA heuristic assigns the least reliable component into all positions and then assigns the unassigned most reliable component into the position with the highest B-importance. In contrast, the LKB heuristic assigns the most reliable component into all positions and then assigns the unassigned least reliable component into the position with the smallest B-importance. Yao *et al.* [95] proposed the B-importance-based two-stage approach (BITA) for solving the CAP, which combines the LK-type heuristic with ZK-type heuristic.

Meta-heuristic methods have also been used to solve the CAP. Shingyochi *et al.* [96] and Shingyochi and Yamamoto [97] provided the efficient genetic algorithms for find-

[3]In their original definition, the meta-heuristic methods are problem-independent algorithmic frameworks, whereas the heuristic methods are designed for a specific problem and inapplicable for other problems. In the context of the CAP, however, meta-heuristic methods refer not just to a framework but also to an algorithm for solving a specific problem. Hence, a group of methods based on frameworks such as the genetic algorithm, the simulated annealing, and the ant colony optimization is called *meta-heuristics* in this thesis.

ing the optimal arrangement in a Lin/Con/k/n:F systems and Cir/Con/k/n:F system. In addition to the genetic algorithm, Shingyochi *et al.* [101] proposed two different types of simulated annealing algorithms: a standard simulated annealing and improved simulated annealing. The improved one can effectively eliminate the equivalent arrangements, which can reduce the solution space. Yao *et al.* [11] proposed a B-importance-based genetic local search approach (BIGLS). The local search is based on the ZK-type heuristics, which can gradually reduce the solution space and find the optimal solution effectively. The numerical experiments showed that the BIGLS algorithm improved almost all the non-optimal arrangements generated by the BITA [11]. Cai *et al.* [98] proposed a B-importance-based genetic algorithm (BIGA), which integrates both the advantages of BITA and the genetic algorithm to search the near-global optimal solution for Lin/Con/k/n:F systems. The numerical experiments showed that the BIGA obtained better results than the BIGLS in large systems [98]. Recently, Wang *et al.* [103] developed an ant colony optimization algorithm, which is also based on the B-importance. They concluded that the algorithm performed better than the BIGLS algorithm for large Lin/Con/k/n:F systems. Approximate methods for consecutive-k-out-of-n:F systems were proposed, whereas those for connected-**X**-out-of-(m, n):F lattice systems have not been reported so far.

Many studies have been devoted to the CAP, wherein all components are functionally exchangeable and can be assigned to any position in the system. Zhu *et al.* [104] extended this CAP to the multi-type component assignment problem (MCAP) for Lin/Con/k/n:F systems. This problem aims to find a component arrangement with the maximum system reliability , assuming that each component should be assigned only to a position that belongs to the same type as the component. Furthermore, Zhu *et al.* [104] proposed a B-importance-based local search method and B-importance-based genetic algorithm for solving the MCAP.

Table 1.1: Summary of the studies for the consecutive-k-out-of-n:F systems and the connected-$\mathbf{X}$-out-of-(m,n):F lattice systems.

	Reliability Evaluation		Computing System Signature	Component Assignment Problem (variant)	
	Exact method	Bounds	(Stochastic Comparison)	Exact method	Approximate method
(a)	[13, 51, 52, 54, 55, 57]	[66–68]	[81–83]	[88, 89]	[11, 87, 94–103]
(b)	[58–62], Chapter 2	[58, 66–71]	Chapter 3	[90, 91, 93], Chapter 4	—
(c)	[63–65, 105], Chapter 2	[66–69] Chapter 2	Chapter 3	—	—

(a) Consecutive-k-out-of-n:F system.
(b) Connected-(r,s)-out-of-(m,n):F lattice system.
(c) Connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F lattice system.

Table 1.2: Approaches for computing the reliability of connected-(r, s)-out-of-(m, n):F lattice systems and connected-$(1, 2)$-or-$(2, 1)$-out-of-(m, n):F lattice systems.

		Recursive equation approach	FMCIA
(b)	Linear-type system	[58, 59]	[60]
	Circular-type system	[61, 62]	—
	Toroidal-type system	Section 2.1	—
(c)	Linear-type system	[63, 64]	Subsection 2.2.2
	Circular-type system	[65], Subsection 2.2.3	Subsection 2.2.3
	Toroidal-type system	Subsection 2.2.4	Subsection 2.2.4

(b) Connected-(r, s)-out-of-(m, n):F lattice system.
(c) Connected-$(1, 2)$-or-$(2, 1)$-out-of-(m, n):F lattice system.

1.4 Research Scope and Objective

In practice, a system exists such that a cluster of failed components causes system failure, which can be modeled as a connected-**X**-out-of-(m, n):F lattice system. Because of theoretical development and practical applications in the research field of reliability, much effort has been devoted to studying the reliability of this system. The main objective of this thesis is to propose methods and algorithms for the reliability evaluation and the optimal design of connected-**X**-out-of-(m, n):F lattice system; specifically, a connected-(r, s)-out-of-(m, n):F lattice system and a connected-$(1, 2)$-or-$(2, 1)$-out-of-(m, n):F lattice system.

Table 1.1 summarizes the studies of reliability evaluation, computing the system signature, and the component assignment problem. Note that the system signature is necessary to establish the stochastic comparisons. The 3rd row of Table 1.1 shows the studies for the consecutive-k-out-of-n:F systems; the 4th row shows those for connected-(r, s)-out-of-(m, n):F lattice systems; the 5th row shows those for connected-$(1, 2)$-or-$(2, 1)$-out-of-(m, n):F lattice systems. The symbol "—" implies that the topic has not been studied, and for example, "Chapter 2" at the 4th row and the 2nd column means that the exact method for a connected-(r, s)-out-of-(m, n):F lattice system will be discussed in Chapter 2 of this thesis.

Table 1.2 summarizes the existing for computing the reliability of connected-(r, s)-out-of-(m, n):F lattice systems and connected-$(1, 2)$-or-$(2, 1)$-out-of-(m, n):F lattice sys-

tems. As it is clear from Table 1.2, most of the research has been devoted to studying the reliability of linear-type and circular-type systems [58–65], whereas and no study has focused on toroidal-type systems. The application shown in Subsection 1.2.3, however, emphasizes the necessity of studying toroidal-type systems; therefore, we present algorithms for efficiently computing the reliability of a Tor/(r, s)/(m, n):F system. Also, previous studies proposed the recursive equation approaches and the FMCIA separately, and no study compares the efficiency of both methods directly. The comparison enables us to select which approach should be used for efficiently computing system reliability. Thus, this thesis proposes both the recursive equation approach and the FMCIA for computing the reliability of Lin, Cir, and Tor/$(1, 2)$-or-$(2, 1)$/(m, n):F systems. Note that the recursive equation approach for a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system has already been proposed by Yamamoto *et al.* [64]. Then, we comprehensively compare the efficiency of the recursive equation approaches and the FMCIA, which is the first-ever attempt to compare both methods. Furthermore, as the size of a system becomes large, obtaining the exact system reliability becomes time-consuming. Accordingly, it would be beneficial to use appropriate upper and lower bounds if obtaining the exact system reliability is not necessary. One can make a trade-off between the computational effort and the quality of the bounds (closeness to the exact value). Several useful and simple bounds were reported, e.g., [66] and [68], whereas the bounds that require the computational burden but are tighter have not been sufficiently discussed. Tighter bounds can appropriately evaluate the system reliability compared with the existing bounds. Thus, we derive tighter bounds for the reliability of a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system at the expense of the computational effort compared with the existing bounds.

This thesis deals with optimizing reliability for only the linear-type system. One reason for studying the linear-type system is to provide a basis for research on the general connected-**X**-out-of-(m, n):F lattice systems.

In reliability engineering, the system signature has various theoretical applications. For example, the system signature has been proved to be significantly useful for establishing a stochastic comparison, presenting the MTTF and so on. However, the computation of a system signature is known to be challenging, especially when a system has a large number of components. Consequently, its practical applications have generally been limited to relatively small systems. Despite the fact that the system signature of a Lin/Con/k/n:F system has been studied [81–83], no study has discussed the system signature of a connected-**X**-out-of-(m, n):F lattice system. Therefore, methods

for efficiently computing the system signature are proposed for a connected-**X**-out-of-(m, n):F lattice system. The obtained system signatures enable us to establish a stochastic comparison of these systems.

This thesis considers the variant optimal arrangement because most of the optimal arrangements depend on the values of component reliabilities. Table 1.1 shows that many studies proposed several heuristic and meta-heuristic methods [11, 87, 94–103]. The advantage of these methods is usually providing good results in a relatively short time, whereas the drawback is not to be able to guarantee global optimality. Meanwhile, exact methods can provide the optimal arrangement and its system reliability, which can be used to measure the performance of heuristic and meta-heuristic methods. Therefore, this thesis focuses on the exact method because optimal arrangements are often necessary. For a Lin/(r, s)/(m, n):F system, as shown in Table 1.1, while several algorithms were developed for finding the optimal arrangement [91, 93], these algorithms are limited to special cases. This thesis thus develops an algorithm for finding the optimal arrangement of a general Lin/(r, s)/(m, n):F system.

The methods and algorithms resulting from the research in this thesis will be useful for the reliability evaluation and the optimal design of practical systems that can be expressed as connected-**X**-out-of-(m, n):F lattice systems. In some cases, this connected-**X**-out-of-(m, n):F lattice system may be an oversimplification. Ideally, we would like to target any system models and present a method and algorithm for the reliability evaluation and the optimal design, but this is a difficult task. It is meaningful to develop a method for efficiently evaluating the system reliability and designing the optimal system for a system with special structures, e.g., the connected-**X**-out-of-(m, n):F lattice system because this study would also be a clue to the studies for more general systems.

Throughout this thesis, we make the following assumptions unless specified otherwise:

(a) each component and the system can have only two states: either working or failed;

(b) all components are mutually statistically independent;

(c) all components reliabilities are given.

1.5 Organization of the Thesis

This thesis considers three problems: the reliability evaluation, computing the system signature, and the component assignment problem. This thesis consists of five chapters.

Chapter 1: Introduction

Chapter 2: Reliability Evaluation

Chapter 3: Computing the System Signature

Chapter 4: Component Assignment Problem

Chapter 5: Conclusions

Also, Figs. 1.12 and 1.13 show the organization of this thesis.

Chapter 1 briefly explains the background and introduces the mathematical concepts useful for understanding subsequent chapters. In addition, the literature reviews related to this thesis are detailed and systematically classified in the consecutive-k system subject area.

Chapter 2 focuses on the system reliability evaluation, which involves computing the system reliability when the component reliabilities are provided. Section 2.1 provides a recursive equation approaches and subsequently propose two kinds of algorithms for efficiently computing the reliability of a Tor/(r, s)/(m, n):F system. An example is used to illustrate the proposed algorithms, and we evaluate the proposed algorithms. Also, we provide the B-importance for a Tor/(r, s)/(m, n):F system by the proposed algorithm. Section 2.2 proposes the recursive equation approaches for Cir and Tor/$(1, 2)$-or-$(2, 1)$/(m, n):F systems and the FMCIA for Lin, Cir, and Tor/$(1, 2)$-or-$(2, 1)$/(m, n):F systems and compares their efficiency. Note that the recursive equation approach for a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system has already been proposed by Yamamoto *et al.* [64]. As mentioned above, there are no studies on efficiency comparisons between both methods. Section 2.3 discusses the upper and lower bounds for the reliability of a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system. We derive new upper and lower bounds and conduct numerical experiments to evaluate the derived bounds. Finally, Section 2.4 summarizes the contributions of the chapter.

Chapter 3 deals with the system signature of a Lin/(r, s)/(m, n):F system and Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system. The system signature has various theoretical applications; for example, it is used to establish stochastic comparisons of some systems.

Section 3.1 presents the existing methods for computing the system signature, which are compared with the proposed methods in order to investigate the efficiency. Sections 3.2 and 3.3 propose methods for computing the system signatures of a Lin/(r, s)/(m, n):F system and a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system, respectively. An example is used to illustrate the proposed algorithms, and we evaluate the proposed algorithms. Additionally, numerical experiments are conducted to compare the efficiency between the proposed and existing methods. As an example, Section 3.4 compares the connected-**X**-out-of-(m, n):F lattice systems based on the stochastic order. Finally, Section 3.5 summarizes the contributions of the chapter.

Chapter 4 considers the CAP, which intends to find the optimal arrangement of components that maximizes the system reliability. Section 4.1 introduces the notation used throughout this chapter and provides a detailed description of the CAP in a Lin/(r, s)/(m, n):F system. Besides, we present the existing properties (e.g., the necessary conditions that the optimal arrangement must satisfy). Section 4.2 develops an algorithm for finding the optimal arrangement in the general case. To speed up this algorithm, we derive three conditions for pruning, which are based on the existing properties. Also, an example is used to illustrate the proposed algorithm. To decrease the computational burden, Section 4.3 focuses on a special case: $r = m - 1$ and $s = n - 1$. We derive four conditions for pruning and subsequently develop an algorithm for finding the optimal arrangement. Note that this algorithm specializes in the case of $r = m - 1$ and $s = n - 1$. An example is used to illustrate the proposed algorithm. In Section 4.4, to investigate the efficiency of the proposed algorithms, we compare these algorithms with the existing algorithm through the numerical experiment. Finally, Section 4.5 summarizes the contributions of the chapter.

The contributions of this thesis are summarized in Chapter 5, and various future perspectives are discussed.

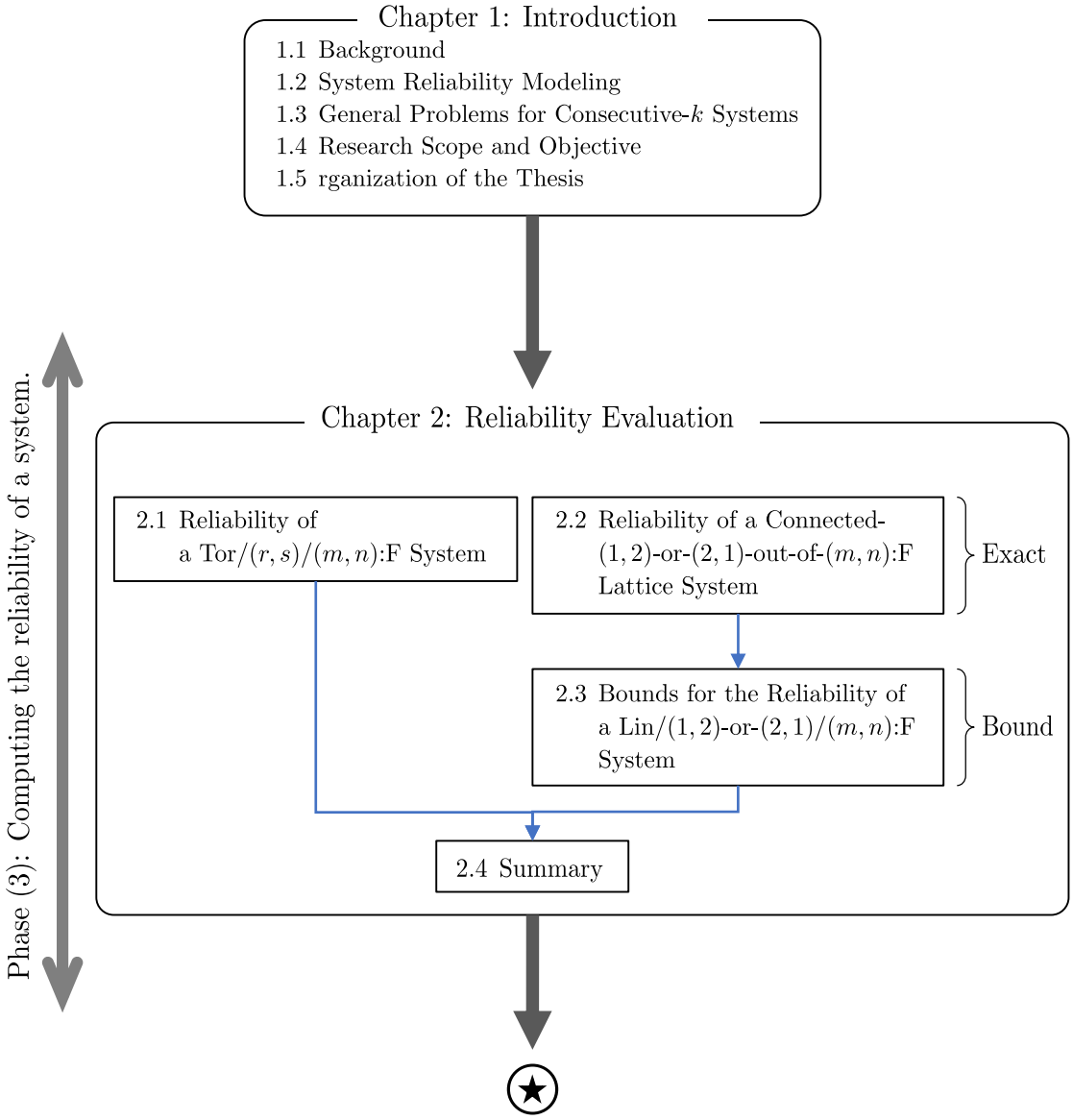


Figure 1.12: Organization of this thesis (Chapters 1 and 2).

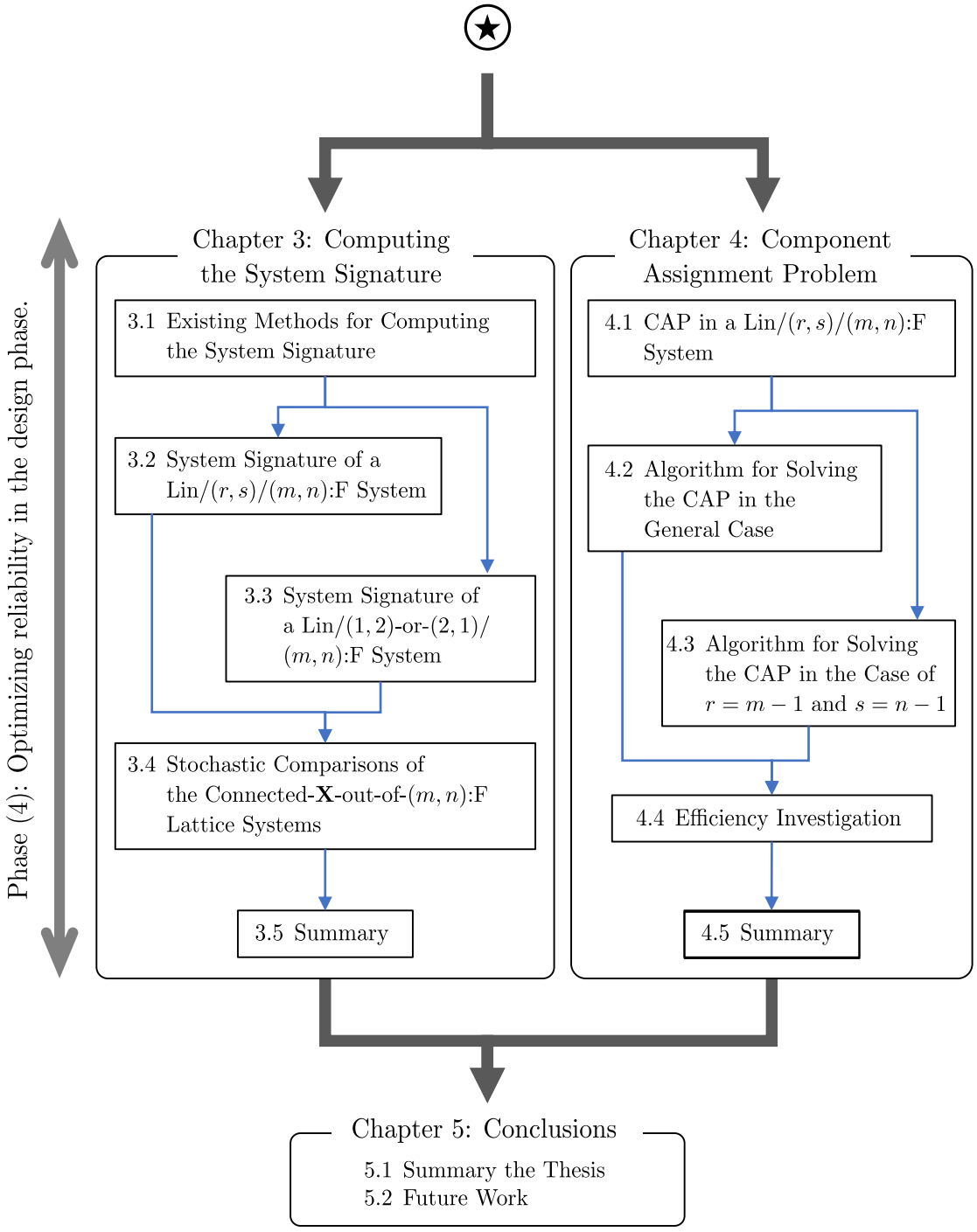


Figure 1.13: Organization of this thesis (Chapters 3, 4, and 5).

Chapter 2

Reliability Evaluation

The aim of this chapter is to propose methods for computing the reliability of some connected-**X**-out-of-(m, n):F lattice systems. The system reliability evaluation is a fundamental step in all reliability studies. Section 2.1 provides a recursive equation approach and subsequently proposes two kinds of algorithms for efficiently computing the reliability of a Tor/(r, s)/(m, n):F system. Also, Section 2.1 provides the B-importance for a Tor/(r, s)/(m, n):F system by the proposed algorithm. Section 2.2 proposes recursive equation approach and the finite Markov chain imbedding approach (FMCIA) for computing the reliability of Lin, Cir, and Tor/$(1, 2)$-or-$(2, 1)$/(m, n):F systems and compares their efficiency. Section 2.3 derives new upper and lower bounds and conducts numerical experiments to evaluate the derived bounds. Finally, Section 2.4 summarizes the contributions of the chapter.

2.1 Reliability of a Tor/(r, s)/(m, n):F System

As already mentioned in Section 1.4, many existing studies focused on linear-type and circular-type connected-**X**-out-of-(m, n):F lattice systems. However, there are practical systems that cannot be evaluated by existing system models because they have a specific structure; one of the examples is a toroidal-type system. In a practical situation, there are various kinds of toroidal-type objects, such as toroidal storage tanks, neutron accelerators, cooling tubes, and offshore platform flotation units [106, 107]. Because a Tor/(r, s)/(m, n):F system might be used to evaluate the reliability of such systems, in this section, we consider the reliability of a Tor/(r, s)/(m, n):F system.

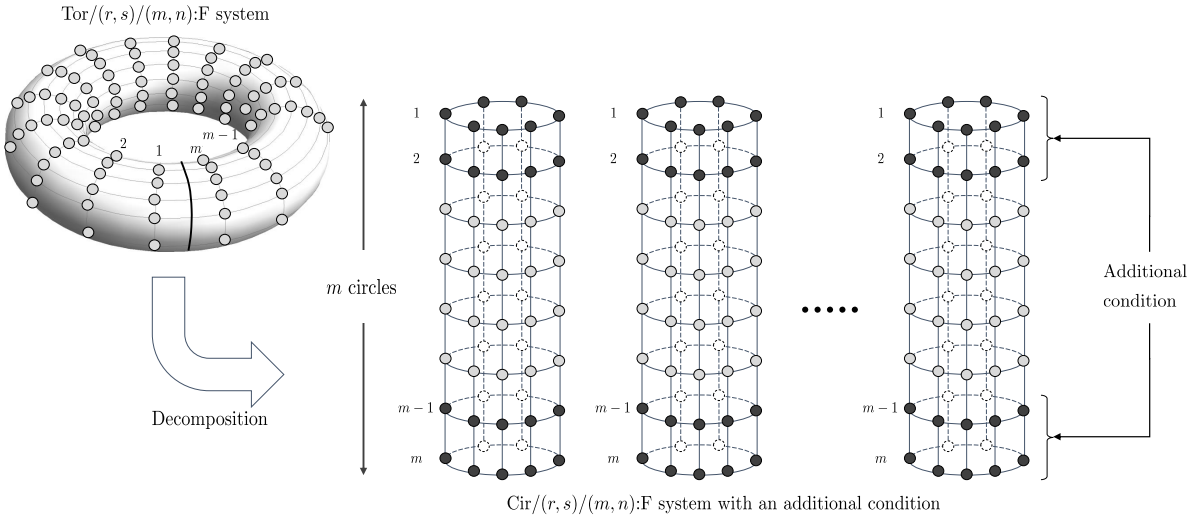


Figure 2.1: Event decomposition of a Tor/(r, s)/(m, n):F system.

2.1.1 Proposal of a Computational Method

This subsection presents a recursive equation approach for efficiently computing the reliability of a Tor/(r, s)/(m, n):F system. First, we employ the event decomposition approach [22] to compute the reliability of the Tor/(r, s)/(m, n):F system. This technique is valuable for computing the reliability of a system and is widely used for deriving the recursive equations [58,62,69]. Likewise, we apply this approach to a Tor/(r, s)/(m, n):F system As a result, "the event that the Tor/(r, s)/(m, n):F system works" can be disjointly decomposed into "several events that each Cir/(r, s)/(m, n):F system with an additional condition works" as shown in Fig. 2.1. This additional condition is explained later. For simplicity, in this chapter, it is assumed that the separation is made between circles 1 and m. Because these events are disjoint, the summation of all the reliabilities of the Cir/(r, s)/(m, n):F systems with an additional condition becomes the reliability of the Tor/(r, s)/(m, n):F system.

For expressing the "additional condition," we define the following notation. For $k = 1, 2, \ldots, m$, $l = 1, 2, \ldots, n$, and $x = 1, 2, \ldots, r - 1$, we define the following random variable.

$$Y_{kl}(x) = \prod_{u=k}^{k+x-1} \prod_{v=l}^{l+s-1} Z_{uv}, \tag{2.1}$$

where Z_{uv} is defined in Eq. (1.5). $Y_{kl}(x)$ takes 1 if all the components fail in the rectangle with four corners (k,l), $(k+x-1,j)$, $(k+x-1,l+s-1)$, and $(k,l+s-1)$; otherwise, it takes 0. In particular, the event $\{Y_{kl}(1)=0\}$ means that at least one of components $(k,l),(k,l+1),\ldots,(k,l+s-1)$ works.

Next, let $\boldsymbol{h},\boldsymbol{g}$ be n-dimensional integer vectors $(h_1,h_2,\ldots,h_n)$, $(g_1,g_2,\ldots,g_n)$, respectively, where $h_l,g_l\in\{0,1,\ldots,r\}$ for $l=1,2,\ldots,n$. We consider an $i\times s$ rectangle with four corners $(1,l)$, (i,l), $(i,l+s-1)$, and $(1,l+s-1)$. For $i=1,2,\ldots,m$ and $l=1,2,\ldots,n$, we define the event $A(i,l;h_l;g_l)$ as

$$A(i,l;h_l;g_l)=$$
$$\begin{cases} \{Y_{1l}(i)=1\}, & \text{if } 1\le i\le h_l \text{ and } g_l=i,\\ \{Y_{1l}(h_l)=1\}\cap\{Y_{il}(1)=0\}, & \text{if } i=h_l+1 \text{ and } g_l=0,\\ \{Y_{1l}(h_l)=1\}\cap\{Y_{h_l+1,l}(1)=0\}\cap W(i,l;h_l;g_l) & \\ \quad\cap\{Y_{i-g_l,l}(1)=0\}\cap\{Y_{i-g_l+1,l}(g_l)=1\}, & \text{if } i\ge h_l+2,\\ \emptyset, & \text{otherwise,}\end{cases} \tag{2.2}$$

where $W(i,l;h_l;g_l)$ means the event that there exists no (r,s) sub-matrix where all the components fail in the rectangle with four corners (h_l+2,l), $(i-g_l-1,l)$, $(i-g_l-1,l+s-1)$, and $(h_l+2,l+s-1)$. The three cases of $A(i,l;h_l;g_l)$ are graphically represented in Fig. 2.2, where the elements of $\boldsymbol{g}$ and $\boldsymbol{h}$ represent the length of a rectangle in which all the components fail.

For $i=r,r+1,\ldots,m$, we define the reliability of a Cir/(r,s)/(i,n):F system in which the states of the components on both ends are given by $\boldsymbol{h}$ and $\boldsymbol{g}$ as follows:

$$R^C(i;\boldsymbol{h};\boldsymbol{g})=\Pr\left\{\bigcap_{l=1}^{n}A(i,l;h_l;g_l)\right\}, \tag{2.3}$$

where both ends mean a few circles including the 1st (ith) circle. Figure 2.2 represents a part of a Cir/(r,s)/(i,n):F system in which the states of the components on both ends are given by $\boldsymbol{h}$ and $\boldsymbol{g}$. Note that, for $i=0$ and any $\boldsymbol{h}$, we define

$$R^C(0;\boldsymbol{h};\boldsymbol{g})=\begin{cases}1, & \text{if } \boldsymbol{g}=(0,\ldots,0),\\ 0, & \text{otherwise.}\end{cases} \tag{2.4}$$

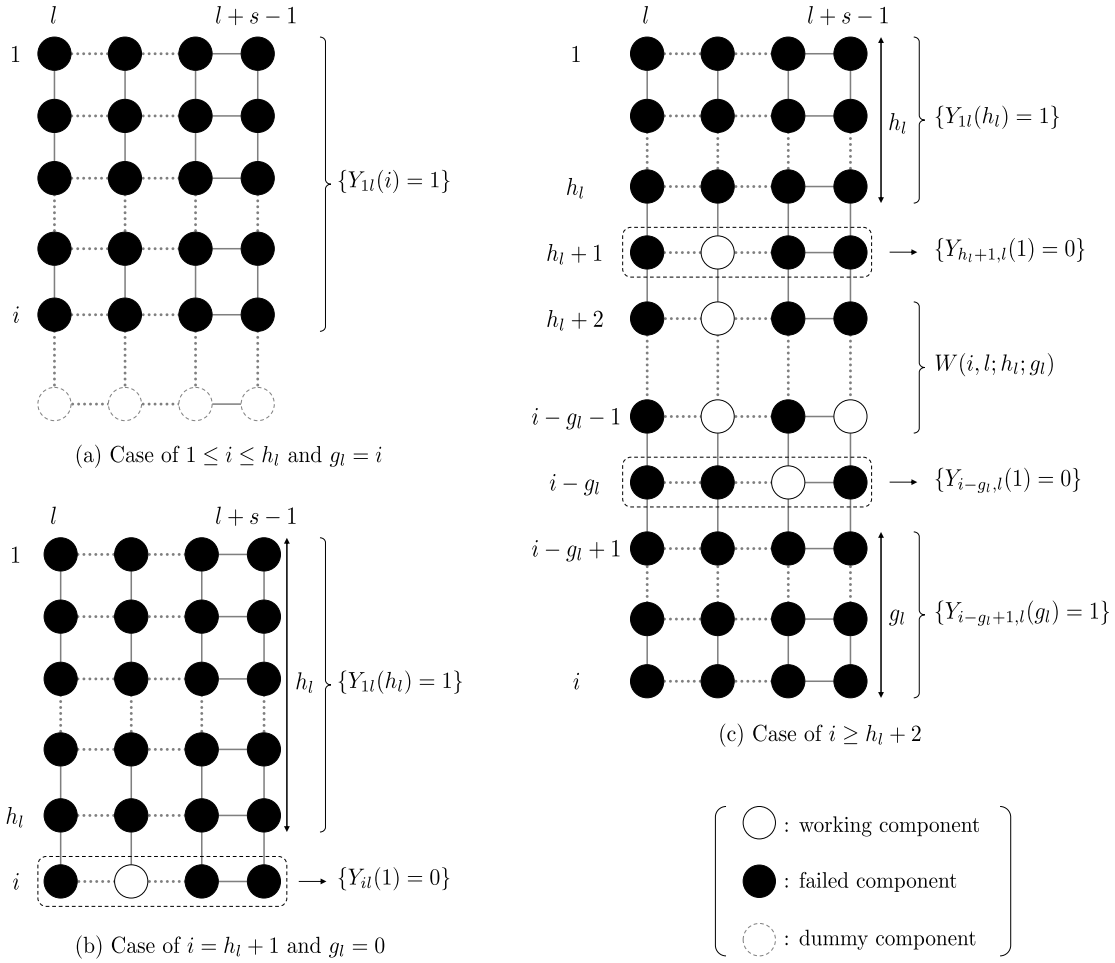


Figure 2.2: Definition of $A(i, l; h_l; g_l)$.

The events that Cir/(r, s)/(m, n):F systems in which the states of the component on both ends are given by $\boldsymbol{h}$ and $\boldsymbol{g}$ are disjoint. Hence, the reliability of the Tor/(r, s)/(m, n):F system can be obtained by summing up the reliabilities of Cir/(r, s)/(m, n):F systems in which the states of the component on both ends are given by $\boldsymbol{h}$ and $\boldsymbol{g}$. However, if there exists an (r, s) sub-matrix where all the components fail when both ends of a Cir/(r, s)/(m, n):F system are connected, then we have

$$R^C(m; \boldsymbol{h}; \boldsymbol{g}) = 0.$$

Therefore, we need to determine $\boldsymbol{h}$ and $\boldsymbol{g}$ such that there exists no (r, s) sub-matrix where all the components fail when both ends are connected. Hence, we define the sets

S and $T(\boldsymbol{h})$ by

$$S = \left\{ \boldsymbol{h} \;\middle|\; \prod_{l=1}^{n} \mathcal{X}(h_l < r) = 1 \right\}, \tag{2.5}$$

and for $\boldsymbol{h} \in S$,

$$T(\boldsymbol{h}) = \left\{ \boldsymbol{g} \;\middle|\; \prod_{l=1}^{n} \mathcal{X}(h_l + g_l < r) = 1 \right\}, \tag{2.6}$$

where $\mathcal{X}(G)$ is an indicator function that takes 1 if argument G is true; 0 otherwise. $T(\boldsymbol{h})$ is a set of $\boldsymbol{g}$ such that there exists no (r, s) sub-matrix where all the components fail when $\boldsymbol{h}$ is given and both ends are connected. By using the above sets, the reliability of the Tor/(r, s)/(m, n):F system is computed by

$$R^T((r, s), (m, n), P) = \sum_{\boldsymbol{h} \in S} \sum_{\boldsymbol{g} \in T(\boldsymbol{h})} R^C(m; \boldsymbol{h}; \boldsymbol{g}). \tag{2.7}$$

However, some $\boldsymbol{g}$s ($\boldsymbol{h}$s) are impossible to exist. For example, suppose $s = 2$, $n = 3$, and $\boldsymbol{g} = (g_1, g_2, g_3) = (1, 0, 1)$. $g_1 = 1$ and $g_3 = 1$ state that components $(i, 1)$, $(i, 2)$, and $(i, 3)$ fail, whereas $g_2 = 0$ states that component either $(i, 2)$ or $(i, 3)$ is working, which is clearly impossible. For convenience, if $\boldsymbol{h}$ or $\boldsymbol{g}$ is impossible to exist, we define

$$R^C(i; \boldsymbol{h}; \boldsymbol{g}) = 0$$

for any i. We provide the sets of $\boldsymbol{h}$ and $\boldsymbol{g}$ that are impossible to exist based on Yamamoto and Akiba [62], which are expressed by

$$C_h = \left\{ \boldsymbol{h} \;\middle|\; \begin{array}{l} h_l > h_{l+1} \text{ and } h_{l+x} < h_{l+x+1} \\ \text{for some } l \in \{1, 2, \ldots, n\} \text{ and some } x \in \{1, 2, \ldots, s-1\} \end{array} \right\}, \tag{2.8}$$

and

$$C_g = \left\{ \boldsymbol{g} \;\middle|\; \begin{array}{l} g_l > g_{l+1} \text{ and } g_{l+x} < g_{l+x+1} \\ \text{for some } l \in \{1, 2, \ldots, n\} \text{ and some } x \in \{1, 2, \ldots, s-1\} \end{array} \right\}, \tag{2.9}$$

where $h_l = h_{l-n}$ and $g_l = g_{l-n}$ for $l = n+1, n+2, \ldots, n+s$.

Using the sets in Eqs. (2.8) and (2.9), for $i = 1, 2, \ldots, m$, if $\boldsymbol{h} \in C_h$ or $\boldsymbol{g} \in C_g$, then we have

$$R^C(i; \boldsymbol{h}; \boldsymbol{g}) = 0,$$

and hence, we can eliminate $\boldsymbol{h} \in C_h$, $\boldsymbol{g} \in C_g$ from the sets S, $T(\boldsymbol{h})$, respectively. Consequently, we can obtain $R^T((r, s), (m, n), P)$ efficiently from Eq. (2.7).

To describe a recursive equation approach, we introduce some additional notation. A vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$ represents the states of n components in a ring, where $x_j = 0$ if the jth component works; otherwise $x_j = 1$ for $j = 1, 2, \ldots, n$. For convenience, $x_j = x_{j-n}$ for $j = n + 1, n + 2, \ldots, n + s - 1$. Also, we introduce an n-dimensional 0-1 binary vector $\boldsymbol{d} = (d_1, d_2, \ldots, d_n)$, and the element d_l is defined, for $l = 1, 2, \ldots, n$, as

$$d_l = \prod_{v=l}^{l+s-1} x_v. \tag{2.10}$$

In words, $d_l = 1$ if all the components $(i, l), (i, l + 1), \ldots, (i, l + s - 1)$ fail; otherwise $d_l = 0$. Note that component (i, l) is regarded as component $(i, l - n)$ for $l = n + 1, n + 2, \ldots, n + s - 1$ and $i = 1, 2, \ldots, m$. Moreover, for $i = 1, 2, \ldots, m$ and $\boldsymbol{d} \in \{0, 1\}^n$, $F_i(\boldsymbol{d})$ is defined as

$$F_i(\boldsymbol{d}) = \Pr\left\{ \bigcap_{l=1}^{n} \{Y_{il}(1) = d_l\} \right\}, \tag{2.11}$$

which is a probability that the states of components in circle i are given by $\boldsymbol{d}$. Because we assume that all components are mutually statistically independent, $F_i(\boldsymbol{d})$ can be computed by

$$F_i(\boldsymbol{d}) = \sum_{\boldsymbol{x} \in U(\boldsymbol{d})} \prod_{j=1}^{n} {p_{ij}}^{1-x_j} (1 - p_{ij})^{x_j}, \tag{2.12}$$

where

$$U(\boldsymbol{d}) = \left\{ \boldsymbol{x} \in \{0, 1\}^n \;\middle|\; \prod_{v=l}^{l+s-1} x_v = d_l, (l = 1, 2, \ldots, n) \right\}. \tag{2.13}$$

Furthermore, let $\boldsymbol{g}'$ be an n-dimensional integer vector $(g'_1, g'_2, \ldots, g'_n)$, where $g'_l \in \{0, 1, \ldots, r\}$ for $l = 1, 2, \ldots, n$.

Using the above notation, we can compute the reliability of the Tor/(r,s)/(m,n):F system by the following theorem.

Theorem 2.1. (a) For $i = 1, 2, \ldots, m$ and $\boldsymbol{h}, \boldsymbol{g} \in S$,

$$R^C(i;\boldsymbol{h};\boldsymbol{g}) = \begin{cases} F_i(\boldsymbol{d}) \displaystyle\sum_{\boldsymbol{g}' \in \Theta(i;\boldsymbol{h};\boldsymbol{g})} R^C(i-1;\boldsymbol{h};\boldsymbol{g}'), & \text{if } \{(1 \le i \le h_l \text{ and } g_l = i) \\ & \quad \text{or } (i = h_l + 1 \text{ and } g_l = 0) \\ & \quad \text{or } (i \ge h_l + 2) \text{ for any } l\} \\ & \quad \text{and } (\boldsymbol{h} \notin C_h \text{ and } \boldsymbol{g} \notin C_g), \\ 0, & \text{otherwise,} \end{cases} \tag{2.14}$$

where

$$\Theta(i;\boldsymbol{h};\boldsymbol{g}) = \left\{ \boldsymbol{g}' \;\middle|\; \begin{array}{ll} g'_l \in \{g_l - 1\}, & \text{if } (2 \le i \le h_l \text{ and } g_l = i) \\ & \text{or } (i \ge h_l + 2 \text{ and } g_l > 0), \\ g'_l \in \{h_l\}, & \text{if } i = h_l + 1 \text{ and } g_l = 0, \\ g'_l \in \{0, 1, \ldots, r-1\}, & \text{if } i \ge h_l + 2 \text{ and } g_l = 0, \end{array} \right\}, \tag{2.15}$$

and for $l = 1, 2, \ldots, n$,

$$d_l = \begin{cases} 0, & \text{if } g_l = 0, \\ 1, & \text{if } g_l > 0. \end{cases} \tag{2.16}$$

As the boundary condition, for $i = 0$,

$$R^C(i;\boldsymbol{h};\boldsymbol{g}) = \begin{cases} 1, & \text{if } \boldsymbol{g} = (0, \ldots, 0), \\ 0, & \text{otherwise.} \end{cases} \tag{2.17}$$

(b)

$$R^T((r,s),(m,n),P) = \sum_{\boldsymbol{h} \in S \setminus C_h} \sum_{\boldsymbol{g} \in T(\boldsymbol{h}) \setminus C_g} R^C(m;\boldsymbol{h};\boldsymbol{g}), \tag{2.18}$$

where the sets $S, T(\boldsymbol{h}), C_h$, and C_g are given by Eqs. (2.5), (2.6), (2.8), and (2.9), respectively.

We provide a proof of Theorem 2.1 in Appendix.

2.1.2 Algorithm and Illustrative Example

In this subsection, we propose an algorithm based on Theorem 2.1 for efficiently computing the reliability of the Tor/(r,s)/(m,n):F system. The critical point of the algorithm is how to compute the recursive equation in Eq. (2.14). Although this recursive equation is expressed as a recursive form, it can also be coded by using iteration, such as a for-loop. When recursion and iteration are compared, a recursive program tends to be simple but is not as efficient as iterative ones. The main reason is that the recursion has function calls, and consequently, they place a heavy burden on memory. For efficient computation, therefore, we employ iteration for implementing the algorithm in this section.

When computing $R^C(i;\boldsymbol{h};\boldsymbol{g})$ by Eq. (2.14) recursively, we can save the required memory space. Generally, a memory with $(r^n \times m)$ is required. However, since $R^C(i;\boldsymbol{h};\boldsymbol{g})$ can be computed from only $R^C(i-1;\boldsymbol{h};\boldsymbol{g}')$, $R^C(i;\boldsymbol{h};\boldsymbol{g})$ can be computed as follows. We prepare two memories with size $(r^n \times 1)$: $R^C_{\rm odd}$ and $R^C_{\rm even}$, and provide the initial values for $R^C_{\rm even}$ by Eq. (2.17). For $i = 1, 2, \ldots, m$, if i is odd, then we compute $R^C_{\rm odd}$ from $R^C_{\rm even}$, and subsequently, $R^C_{\rm even}$ is initialized. If i is even, then we compute $R^C_{\rm even}$ from $R^C_{\rm odd}$, and subsequently, $R^C_{\rm odd}$ is initialized. Such a computation of $R^C(i;\boldsymbol{h};\boldsymbol{g})$ needs only a memory with size $(r^n \times 2)$. An example of this computation is given later.

In summary, this algorithm is characterized by two advantages:

1. It utilizes iteration, and thus, the cost of the function call can be avoided.

2. It saves the required memory space because two small spaces (for odd and even) are used alternatively.

Next, we provide the detailed steps of an algorithm based on Theorem 2.1. Basically, the algorithm is executed in the following steps:

Step 1: Enumerate all elements in $S \setminus C_h$ by Eqs. (2.5) and (2.8), and compute $F_i(\boldsymbol{d})$ for $i = 1, 2, \ldots, m$ and $\boldsymbol{d} \in \{0,1\}^n$ by Eq. (2.12).

Step 2: Select $\boldsymbol{h} \in S \setminus C_h$, and set $i = 0$ and $R^C_{\rm even} = (1, 0, \ldots, 0)^{\mathsf{T}}$ based on Eq. (2.17), and set $R^C_{\rm odd} = (0, \ldots, 0)^{\mathsf{T}}$.

Step 3: Set $i = i + 1$. If i is odd, compute $R^C_{\rm odd}$ from $R^C_{\rm even}$ by Eq. (2.14), and subsequently, initialize $R^C_{\rm even}$; if i is even, compute $R^C_{\rm even}$ from $R^C_{\rm odd}$ by Eq. (2.14), and subsequently, initialize $R^C_{\rm odd}$. Repeat this step until $i = m$.

Step 4: Compute $\sum_{\boldsymbol{g}\in T(\boldsymbol{h})\setminus C_g} R^C(m;\boldsymbol{h};\boldsymbol{g})$ in Eq. (2.18) for the fixed $\boldsymbol{h}$.

Step 5: If all elements in $S \setminus C_h$ have been selected, go to **Step 6**; otherwise, go back to **Step 2**.

Step 6: Obtain $R^T((r,s),(m,n),P)$ by Eq. (2.18).

When computing $\sum_{\boldsymbol{g}'\in\Theta(i;\boldsymbol{h};\boldsymbol{g})} R^C(i-1;\boldsymbol{h};\boldsymbol{g}')$ in Step 3, we need the set $\Theta(i;\boldsymbol{h};\boldsymbol{g})$. Considering when we should obtain the set $\Theta(i;\boldsymbol{h};\boldsymbol{g})$, we have two cases:

Case A) before computing $\displaystyle\sum_{\boldsymbol{g}'\in\Theta(i;\boldsymbol{h};\boldsymbol{g})} R^C(i-1;\boldsymbol{h};\boldsymbol{g}')$s in Step 3;

Case B) in Step 1 as a preprocessing step.

An algorithm in Case A is just called "Algorithm A" in this section. In Algorithm A, the set $\Theta(i;\boldsymbol{h};\boldsymbol{g})$ is generated again and again in Step 3. It is obvious from Eq. (2.15) that $\Theta(i;\boldsymbol{h};\boldsymbol{g})$ does not depend on $\boldsymbol{h}$ and i when $i \geq r+2$. Hence, we enumerate the elements of $\Theta(i;\boldsymbol{h};\boldsymbol{g})$ with $i \geq r+2$ for $\boldsymbol{g} \in S \setminus C_g$ beforehand (i.e., Case B) and store them. As a result, although preprocessing time and extra memory space are needed, we can use the elements instantly, which means that computational complexity may decrease. An algorithm in Case B is just called "Algorithm B" in this section.

Finally, we demonstrate how Algorithm A computes the reliability of the Tor/(r,s)/(m,n):F system. For instance, let us consider a Tor/$(2,2)$/$(3,3)$:F system with common component reliability p and unreliability q, where $p+q=1$ for a simple illustration.

Step 1: We enumerate all the elements in $S \setminus C_h$ as follows:

$$S \setminus C_h = \{(0,0,0),(0,0,1),(0,1,0),(1,0,0),(1,1,1)\},$$

and, we compute $F_i(\boldsymbol{d})$ as follows:

$$\begin{aligned} &F_i((0,0,0)) = p^3 + 3p^2q, && F_i((0,0,1)) = pq^2,\\ &F_i((0,1,0)) = pq^2, && F_i((1,0,0)) = pq^2,\\ &F_i((1,1,1)) = q^3. \end{aligned}$$

For convenience, we use the following notation. For $\boldsymbol{h} \in S \setminus C_h$ if i is even,

$$R^C_{\text{even}} = \begin{pmatrix} R^C(i;\boldsymbol{h};(0,0,0)) \\ R^C(i;\boldsymbol{h};(0,0,1)) \\ R^C(i;\boldsymbol{h};(0,1,0)) \\ R^C(i;\boldsymbol{h};(1,0,0)) \\ R^C(i;\boldsymbol{h};(1,1,1)) \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{pmatrix}^{\mathsf{T}},$$

and, if i is odd,

$$R^C_{\text{odd}} = \begin{pmatrix} R^C(i;\boldsymbol{h};(0,0,0)) \\ R^C(i;\boldsymbol{h};(0,0,1)) \\ R^C(i;\boldsymbol{h};(0,1,0)) \\ R^C(i;\boldsymbol{h};(1,0,0)) \\ R^C(i;\boldsymbol{h};(1,1,1)) \end{pmatrix}^{\mathsf{T}} = \begin{pmatrix} b_1 \\ b_2 \\ b_3 \\ b_4 \\ b_5 \end{pmatrix}^{\mathsf{T}}.$$

Step 2: For example, we select $\boldsymbol{h} = (1,0,0)$, set $i = 0$, $R^C_{\text{even}} = (1,0,0,0,0)$, and set $R^C_{\text{odd}} = (0,0,0,0,0)$.

Step 3: First, we consider the $i = 1$ case. Since i is odd, we compute R^C_{odd} from R^C_{even} as follows:

$$\begin{aligned} b_4 &= F_1((1,0,0)) \sum_{\boldsymbol{g}' \in \Theta(1;(1,0,0);(1,0,0))} R^C(0;(1,0,0);\boldsymbol{g}'), \\ &= pq^2 \times a_1, \\ &= pq^2, \end{aligned}$$

where

$$\Theta(1;(1,0,0);(1,0,0)) = \{(0,0,0)\},$$

and $b_1 = b_2 = b_3 = b_5 = 0$. Subsequently, we initialize R^C_{even}.

Next, we consider the $i = 2$ case. Since i is even, we compute R^C_{even} from R^C_{odd} as

follows:

$$\begin{aligned} a_1 &= F_2((0,0,0)) \sum_{\boldsymbol{g}' \in \Theta(2;(1,0,0);(0,0,0))} R^C(1;(1,0,0);\boldsymbol{g}'), \\ &= (p^3 + 3p^2q) \times b_4, \\ &= (p^3 + 3p^2q)pq^2, \end{aligned}$$

where

$$\Theta(2;(1,0,0);(1,0,0)) = \{(1,0,0)\}.$$

Similarly,

$$\begin{aligned} a_2 &= pq^2 \times b_4 = p^2q^4, \\ a_3 &= pq^2 \times b_4 = p^2q^4, \\ a_4 &= 0, \\ a_5 &= 0. \end{aligned}$$

Subsequently, we initialize R^C_{odd}.

Finally, we consider the $i = 3$ case. Since i is odd, we compute R^C_{odd} from R^C_{even} as follows:

$$\begin{aligned} b_1 &= (p^3 + 3p^2q) \times (a_1 + a_2 + a_3), \\ &= (p^3 + 3p^2q)(p^3 + 3p^2q)pq^2 2p^2q^4, \\ b_2 &= pq^2 \times (a_1 + a_3) = pq^2(p^3 + 3p^2q)pq^2p^2q^4, \\ b_3 &= pq^2 \times (a_1 + a_2) = pq^2(p^3 + 3p^2q)pq^2p^2q^4. \end{aligned}$$

Note that b_4 and b_5 are unnecessary in the next step.

Step 4: We compute

$$\sum_{\boldsymbol{g} \in T((1,0,0)) \setminus C_g} R^C(3;(1,0,0);\boldsymbol{g}) = b_1 + b_2 + b_3,$$

where

$$T((1,0,0)) \setminus C_g = \{(0,0,0),(0,0,1),(0,1,0)\},$$

and memorize the value. **Steps 2–4** continue until all the elements in $S \setminus C_h$ are selected

(**Step 5**).
Step 6: Eventually, we obtain $R^T((2,2),(3,3),P)$ as follows:

$$\begin{aligned}
R^T((2,2),(3,3),P) &= \sum_{\boldsymbol{h}\in S\setminus C_h}\ \sum_{\boldsymbol{g}\in T(\boldsymbol{h})\setminus C_g} R^C(3;\boldsymbol{h};\boldsymbol{g}),\\
&= \sum_{\boldsymbol{g}\in T((0,0,0))\setminus C_g} R^C(3;(0,0,0);\boldsymbol{g}) + \sum_{\boldsymbol{g}\in T((0,0,1))\setminus C_g} R^C(3;(0,0,1);\boldsymbol{g})\\
&\quad + \sum_{\boldsymbol{g}\in T((0,1,0))\setminus C_g} R^C(3;(0,1,0);\boldsymbol{g}) + \sum_{\boldsymbol{g}\in T((1,0,0))\setminus C_g} R^C(3;(1,0,0);\boldsymbol{g})\\
&\quad + \sum_{\boldsymbol{g}\in T((1,1,1))\setminus C_g} R^C(3;(1,1,1);\boldsymbol{g}).
\end{aligned}$$

Also, Algorithm B enumerates all the elements in $\Theta(i;\boldsymbol{h};\boldsymbol{g})$ as follows:

$$\begin{aligned}
\Theta(i;\boldsymbol{h};(0,0,0)) &= \{(0,0,0),(0,0,1),(0,1,0),(1,0,0),(1,1,1)\},\\
\Theta(i;\boldsymbol{h};(0,0,1)) &= \{(0,0,0),(0,1,0),(1,0,0)\},\\
\Theta(i;\boldsymbol{h};(0,1,0)) &= \{(0,0,0),(0,0,1),(1,0,0)\},\\
\Theta(i;\boldsymbol{h};(1,0,0)) &= \{(0,0,0),(0,0,1),(0,1,0)\},\\
\Theta(i;\boldsymbol{h};(1,1,1)) &= \{(0,0,0)\},
\end{aligned}$$

for $i = r+2, r+3, \ldots, m$ and any $\boldsymbol{h}$, and then they are stored in Step 1 as a preprocessing step. As a result, in Step 3 of Algorithm B, we do not need to enumerate the elements in $\Theta(i;\boldsymbol{h};\boldsymbol{g})$ if $i \geq r+2$.

2.1.3 Efficiency Investigation

Computational complexity analysis

Here, the computational complexity analysis is conducted for the comparison of Algorithms A and B. First, we analyze the time and space complexities of Algorithm A. Step 1 takes $O(m2^n)$ time to compute $F_i(\boldsymbol{d})$ for $i = 1, 2, \ldots, m$ and $\boldsymbol{d} \in \{0,1\}^n$ because the maximum number of $F_i(\boldsymbol{d})$s is $m2^n$. In Step 3, once $\boldsymbol{h}$ is determined, for each i, the maximum number of $R^C(i;\boldsymbol{h};\boldsymbol{g})$s is r^n, and the set S has at most r^n elements. Because it takes $O(n)$ time to obtain the set $\Theta(i;\boldsymbol{h};\boldsymbol{g})$, Step 3 needs at most $O(mnr^{2n})$ time. Hence, Algorithm A needs $O(m2^n + r^n(mnr^n))$ $(= O(m2^n + mnr^{2n}))$ time. Also,

Table 2.1: Time and space complexities of Algorithms A and B.

	time	space
Algorithm A	$O(m2^n + r^n(mnr^n))$	$O(m2^n + r^n)$
Algorithm B	$O(m2^n + r^n(n + mr^n))$	$O(m2^n + r^n + r^{2n})$

$F_i(\boldsymbol{d})$ requires $O(m2^n)$ space. The maximum number of the memories required for $R^C(i-1;\boldsymbol{h};\boldsymbol{g}')$ is $O(2r^n)$ because we prepare two r^n memories for odd i and even i. Hence, Algorithm A needs $O(m2^n + r^n)$ $(= O(m2^n + 2r^n))$ space.

Next, we analyze the time and space complexities of Algorithm B. Step 1 takes $O(m2^n)$ time in a similarly to Algorithm A. It also needs at most $O(nr^n)$ time to prepare the set $\Theta(i;\boldsymbol{h};\boldsymbol{g})$. In Step 3, it takes at most $O(mr^{2n})$ time. Hence, Algorithm B needs $O(m2^n + r^n(n + mr^n))$ $(= O(m2^n + nr^n + mr^{2n}))$ time. Also, $F_i(\boldsymbol{d})$ requires $O(m2^n)$ space. $\Theta(i;\boldsymbol{h};\boldsymbol{g})$ needs at most $O(r^n \times r^n)$ space. The maximum number of the memories required for $R^C(i-1;\boldsymbol{h};\boldsymbol{g}')$s is $O(2r^n)$ in a similarly to Algorithm A. Hence, Algorithm B needs $O(m2^n + r^n + r^{2n})$ $(= O(m2^n + 2r^n + r^{2n}))$ space.

Table 2.1 summarizes the time and space complexities of Algorithms A and B. In conclusion, Algorithm A needs less memory space than Algorithm B, but Algorithm B has better time complexity compared to Algorithm A.

From the results, we see that the time complexity of both algorithms is polynomial order for m but exponential order for n. Note that we can obtain the same result even if we swap r and s, m and n simultaneously due to structural symmetry. In other words, the reliability of the Tor$/(r,s)/(m,n)$:F system is equivalent to that of the Tor$/(s,r)/(n,m)$:F system. Thus, in order to efficiently compute the system reliability, the parameter n should be the smaller value of either circle or ring.

Numerical experiment

Here, we investigate the efficiency of Algorithms A and B through numerical experiments. Computation time also plays a key role in the measure of the computational efficiency of different algorithms, and thus, we compare both algorithms in terms of the actual computation time. Each method was programmed in MATLAB R2018a and implemented on a computer with Intel Core i5 3.20 GHz CPU, 8.0 GB memory, Microsoft Windows 10 OS. In the numerical experiments, for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$,

Table 2.2: Comparison of computation times of Algorithms A and B ($(r, s) = (2, 2)$).

m	n	(a) Alg. A (sec.)	(b) Alg. B (sec.)	(b)/(a) (%)	$R^T((2,2),(m,n),P)$
5	4	0.0622	0.0757	121.6	0.9924
10	4	0.0645	0.0757	117.4	0.9848
50	4	0.0861	0.0830	96.4	0.9262
100	4	0.1126	0.0869	77.2	0.8579
5	6	0.0958	0.0860	89.7	0.9886
10	6	0.1346	0.1039	77.2	0.9773
50	6	0.3401	0.1358	39.9	0.3396
100	6	0.5966	0.1687	28.3	0.7945
5	8	0.5142	0.2355	45.8	0.9848
10	8	1.0603	0.2882	27.2	0.9698
50	8	4.8967	0.6848	14.0	0.8578
100	8	9.5726	1.1458	12.0	0.7359
5	10	7.9547	1.8794	23.6	0.9810
10	10	18.7168	2.5650	13.7	0.9624
50	10	105.6483	7.4481	7.0	0.8256
100	10	213.5176	13.6000	6.4	0.6816
5	12	214.7146	44.2102	20.6	0.9773
10	12	505.9578	53.9138	10.7	0.9550
50	12	2785.7716	130.0507	4.7	0.7945
100	12	5747.4053	224.5364	3.9	0.6313
5	14	7461.2738	1549.9041	20.8	0.9735
10	14	17798.3021	1714.3453	9.6	0.9477
50	14	N/A	2939.6827	—	0.7647
100	14	N/A	4797.1235	—	0.5847

the component reliabilities P are given by

$$p_{ij} = \begin{cases} 0.9, & \text{if } i+j \text{ is odd,} \\ 0.8, & \text{if } i+j \text{ is even.} \end{cases}$$

Tables 2.2 and 2.3 show the comparison of the computation time required for the

Table 2.3: Comparison of computation times of Algorithms A and B ($(m, n) = (20, 6)$).

r	s	(a) Alg. A (sec.)	(b) Alg. B (sec.)	(b)/(a) (%)	$R^T((r, s), (20, 6), P)$
2	2	0.1897	0.1137	59.9	0.2497
3	2	7.1225	1.7938	25.2	0.8406
4	2	220.8854	50.3901	22.8	0.9788
2	3	0.1313	0.1026	78.1	0.8406
3	3	2.4281	0.6265	25.8	0.9921
4	3	47.2759	10.4654	22.1	0.9997
2	4	0.0992	0.0891	89.8	0.9787
3	4	0.9341	0.2723	29.1	0.9997
4	4	10.5731	2.3424	22.2	1.0000

cases of $(r, s) = (2, 2)$ and $(m, n) = (20, 6)$, respectively. In these tables, N/A means that the algorithm failed to compute the system reliability in 12 hours. We confirmed that both algorithms provided the same results.

First, we investigate the efficiency of both algorithms in the case where (r, s) is fixed. It is clear from Table 2.2 that the algorithms perform better when the parameter m is large, whereas computation time increases with the parameter n. This is because the number of elements of the sets S and $T(\boldsymbol{h})$ increases exponentially as the parameter n is large. Moreover, when comparing Algorithms A and B, in the case where m and n are small, we found that although a computation time of Algorithm A was shorter than that of Algorithm B, their computation times were sufficiently short. In contrast, when m or n is not small, Table 2.2 shows that Algorithm A requires more computation times compared with Algorithm B.

Next, we investigate the efficiency of both algorithms in the case where (m, n) is fixed. As it is clear from Table 2.3, the computation time of both algorithms increases exponentially as r increases. On the other hand, the result shows that both algorithms perform well as the parameter s becomes large. The reason for this observed behavior is that the number of elements of the sets C_h and C_g increases as the parameter s is large. For example, when $(m, n) = (20, 6)$, if $(r, s) = (2, 2)$, then $|C_h| = 35$; if $(r, s) = (2, 4)$, then $|C_h| = 50$, where $|\Omega|$ refers to the number of elements of the set Ω. Moreover, when comparing Algorithms A and B, we see that the computation time of Algorithm B does not increase with r, which illustrates the efficiency of Algorithm B.

Collectively, these results indicate that Algorithm B outperforms Algorithm A for both cases when we have enough memory space. Note that Algorithm B needs extra memory space ($O(r^{2n})$). The main reason for Algorithm B to be efficient is that we can instantly use the elements of $\Theta(i; \boldsymbol{h}; \boldsymbol{g})$, which are enumerated beforehand and stored. As a result, it is possible to avoid redundant computations.

2.1.4 Birnbaum Importance for a Tor/$(r,s)/(m,n)$:F System

In this subsection, we discuss the Birnbaum importance (B-importance) for a Tor/$(r,s)/(m,n)$:F system. The failure of a component may cause system failure, whereas another failed component has little effect on the system state. Components that tend to cause system failure are considered as more important than the others in the system, and thus, such important components should be strictly maintained. Therefore, we provide B-importance for the Tor/$(r,s)/(m,n)$:F system by the proposed algorithm as a numerical example.

Let $I^B(i,j)$ be the B-importance of component (i,j) in a Tor/$(r,s)/(m,n)$:F system with component reliabilities P, and then, from Eq. (1.19), $I^B(i,j)$ is computed by

$$I^B(i,j) = R^T((r,s),(m,n),(1_{(i,j)},P)) - R^T((r,s),(m,n),(0_{(i,j)},P)), \tag{2.19}$$

where

$$(k_{(i,j)},P) = \begin{pmatrix} p_{11} & \cdots & p_{1j} & \cdots & p_{1n} \\ \vdots & \ddots & \vdots & & \vdots \\ p_{i1} & \cdots & k & \cdots & p_{in} \\ \vdots & & \vdots & \ddots & \vdots \\ p_{m1} & \cdots & p_{mj} & \cdots & p_{mn} \end{pmatrix},$$

for $i = 1,2,\ldots,m$, $j = 1,2,\ldots,n$, and $k \in \{0,1\}$. In words, $(1_{(i,j)},P)$ is a matrix obtained by replacing p_{ij} with 1 in P. The proposed algorithms enable us to easily compute the B-importance.

We computed the B-importance of each component in a Tor/$(2,2)/(7,7)$:F system with $p_{ij} = 0.7$ for any i,j by the proposed algorithm as shown in Table 2.4. The computation time was about 4 seconds. Table 2.4 shows that each component has the same B-importance, which means that each component has an equal opportunity to contribute to the system failure because of structural symmetry. We also computed the

Table 2.4: Birnbaum importance for a Tor/(2, 2)/(7, 7):F system with $p_{ij} = 0.7$ for any i, j.

	Ring 1	Ring 2	Ring 3	Ring 4	Ring 5	Ring 6	Ring 7
Circle 1	0.060	0.060	0.060	0.060	0.060	0.060	0.060
Circle 2	0.060	0.060	0.060	0.060	0.060	0.060	0.060
Circle 3	0.060	0.060	0.060	0.060	0.060	0.060	0.060
Circle 4	0.060	0.060	0.060	0.060	0.060	0.060	0.060
Circle 5	0.060	0.060	0.060	0.060	0.060	0.060	0.060
Circle 6	0.060	0.060	0.060	0.060	0.060	0.060	0.060
Circle 7	0.060	0.060	0.060	0.060	0.060	0.060	0.060

Table 2.5: Birnbaum importance for a Tor/(2, 2)/(7, 7):F system with $p_{4,4} = 0.3$ and $p_{ij} = 0.7$ for the others.

	Ring 1	Ring 2	Ring 3	Ring 4	Ring 5	Ring 6	Ring 7
Circle 1	0.058	0.058	0.058	0.059	0.058	0.058	0.058
Circle 2	0.058	0.058	0.056	0.055	0.056	0.058	0.058
Circle 3	0.058	0.056	0.077	0.098	0.077	0.056	0.058
Circle 4	0.059	0.055	0.098	**0.060**	0.098	0.055	0.059
Circle 5	0.058	0.056	0.077	0.098	0.077	0.056	0.058
Circle 6	0.058	0.058	0.056	0.055	0.056	0.058	0.058
Circle 7	0.058	0.058	0.058	0.059	0.058	0.058	0.058

B-importance of each component in a Tor/(2, 2)/(7, 7):F system with $p_{4,4} = 0.3$ and $p_{ij} = 0.7$ for the others as shown in Table 2.5, where the value denoted in boldface is the least reliable component, which is located at $(4, 4)$. Table 2.5 shows that the components adjacent to component $(4, 4)$ have a high B-importance in order to avoid system failure. Thus, it suggests that we should somehow improve one of the components adjacent to component $(4, 4)$, namely, component $(4, 3)$, $(3, 4)$, $(4, 5)$, or $(5, 4)$.

2.2 Reliability of a Connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F Lattice System

This section proposes the recursive equation approaches for Cir and Tor/$(1,2)$-or-$(2,1)/(m,n)$:F systems and the FMCIA for Lin, Cir, and Tor/$(1,2)$-or-$(2,1)/(m,n)$:F systems. Note that Yamamoto *et al.* [64] proposed the recursive equation approach for a Lin/$(1,2)$-or-$(2,1)/(m,n)$:F system. Because there are no studies on efficiency comparisons between both methods, we compare the efficiency of both methods through numerical experiments. This comparison will give a guideline to decide which method should be used.

2.2.1 Notation

To uniformly describe both methods, we introduce some common notation. First, we focus on a column of a linear-type system (or a ray of a circular-type system). Let $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{y}'$ be m-dimensional integer vectors $(x_1, x_2, \ldots, x_m)$, $(y_1, y_2, \ldots, y_m)$, $(y'_1, y'_2, \ldots, y'_m)$ respectively, where $x_i, y_i, y'_i \in \{0, 1\}$ for $i = 1, 2, \ldots, m$. For example, $x_i = 0$ means that the ith component works and $x_i = 1$ means that it fails. We define the sets W^L and $\Omega^L(\boldsymbol{y})$ as

$$W^L = \left\{ \boldsymbol{y} \;\middle|\; \sum_{i=2}^{m} y_{i-1} y_i = 0 \right\}, \tag{2.20}$$

and

$$\Omega^L(\boldsymbol{y}) = \left\{ \boldsymbol{y}' \in W^L \;\middle|\; \sum_{i=1}^{m} y_i y'_i = 0 \right\}. \tag{2.21}$$

If the states of the m components in a column are given by $\boldsymbol{y}$ and $\boldsymbol{y} \in W^L$, there exists no failure pattern $(2,1)$ in the column. Let us consider the situation where the states of the $2m$ components in the $(j-1)$th and jth columns are given by $\boldsymbol{y}'$ and $\boldsymbol{y}$, respectively. If $\boldsymbol{y}' \in \Omega^L(\boldsymbol{y})$, there exists no failure pattern $(1,2)$ in the $(j-1)$th and jth columns. For $j = 1, 2, \ldots, n$ and $\boldsymbol{y} \in W^L$, $G_j(\boldsymbol{y})$ is a probability that the states of

the m components in the jth column are given by $\boldsymbol{y}$, which is expressed by

$$G_j(\boldsymbol{y}) = \prod_{i=1}^{m} {p_{ij}}^{1-y_i}(1-p_{ij})^{y_i}. \tag{2.22}$$

Second, we focus on a circle of a circular-type or toroidal system. Let $\boldsymbol{h}, \boldsymbol{g}, \boldsymbol{g}'$ be n-dimensional integer vectors $(h_1, h_2, \ldots, h_n)$, $(g_1, g_2, \ldots, g_n)$, $(g'_1, g'_2, \ldots, g'_n)$ respectively, where $h_j, g_j, g'_j \in \{0, 1\}$ for $j = 1, 2, \ldots, n$. The sets W^C and $\Omega^C(\boldsymbol{g})$ are defined as

$$W^C = \left\{ \boldsymbol{g} \;\middle|\; \sum_{j=2}^{n} g_{j-1}g_j = 0 \right\}, \tag{2.23}$$

and

$$\Omega^C(\boldsymbol{g}) = \left\{ \boldsymbol{g}' \in W^C \;\middle|\; \sum_{j=1}^{n} g_j g'_j = 0 \right\}. \tag{2.24}$$

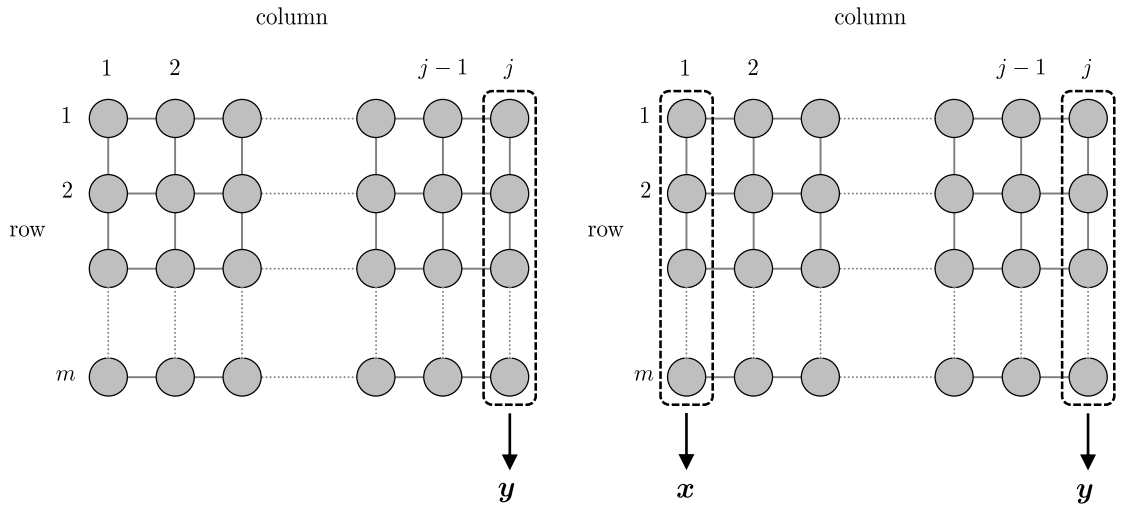
If the states of the n components in a column are given by $\boldsymbol{g}$, and $\boldsymbol{g} \in W^C$, there exists no failure pattern $(1, 2)$ in the circle. Let us consider the situation where the states of the $2n$ components in the $(i-1)$th and ith circles are given by $\boldsymbol{g}'$ and $\boldsymbol{g}$, respectively. If $\boldsymbol{g}' \in \Omega^C(\boldsymbol{g})$, there exists no failure pattern $(2, 1)$ in the $(i-1)$th and ith circles. For $i = 1, 2, \ldots, m$ and $\boldsymbol{g} \in W^C$, $F_i(\boldsymbol{g})$ is expressed by

$$F_i(\boldsymbol{g}) = \prod_{j=1}^{n} {p_{ij}}^{1-g_j}(1-p_{ij})^{g_j}, \tag{2.25}$$

which is a probability that the states of the n components in the ith circle are given by $\boldsymbol{g}$.

2.2.2 Computational Method for a Lin/$(1,2)$-or-$(2,1)$/(m,n):F System

This subsection proposes the FMCIA for computing the reliability of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system. The recursive equation approach in Yamamoto *et al.* [64] was found to be more efficient than a method in Higashiyama [63]. Hence, we only present the recursive equation approach in Yamamoto *et al.* [64] and compare the FMCIA with



(a) Definition of $R^L(j;\boldsymbol{y})$ (b) Definition of $R^L(j;\boldsymbol{x};\boldsymbol{y})$

Figure 2.3: Lin/$(1,2)$-or-$(2,1)$/(m,j):F systems with a condition.

it through numerical experiments.

(A) Recursive equation approach [64]

First, we provide a recursive equation approach for computing the reliability of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system. For $j=1,2,\ldots,n$ and $\boldsymbol{y}\in\{0,1\}^m$, a probability $R^L(j;\boldsymbol{y})$ is defined as

$$R^L(j;\boldsymbol{y})=\Pr\left\{\left\{\prod_{i=1}^{m}\prod_{l=2}^{j}\left(1-Y_{il}^{(1,2)}\right)\times\prod_{i=2}^{m}\prod_{l=1}^{j}\left(1-Y_{il}^{(2,1)}\right)=1\right\}\bigcap\left\{\bigcap_{i=1}^{m}\{Z_{ij}=y_i\}\right\}\right\}, \tag{2.26}$$

which shows the reliability of a Lin/$(1,2)$-or-$(2,1)$/(m,j):F system in which the states of the m components in the jth column are given by $\boldsymbol{y}$. Figure 2.3 (a) illustrates the Lin/$(1,2)$-or-$(2,1)$/(m,j):F system in which the states of the m components in the jth column are given by $\boldsymbol{y}$. Then, we can compute the reliability of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system by the following theorem.

Theorem 2.2 (Yamamoto *et al.* [64])**.** (a) For $j = 1, 2, \ldots, n$ and $\boldsymbol{y}, \boldsymbol{y}' \in \{0,1\}^m$,

$$R^L(j;\boldsymbol{y}) = \begin{cases} G_j(\boldsymbol{y}) \displaystyle\sum_{\boldsymbol{y}' \in \Omega^L(\boldsymbol{y})} R^L(j-1;\boldsymbol{y}'), & \text{if } j \geq 2 \text{ and } \boldsymbol{y} \in W^L, \\ G_j(\boldsymbol{y}), & \text{if } j = 1 \text{ and } \boldsymbol{y} \in W^L, \\ 0, & \text{otherwise,} \end{cases} \tag{2.27}$$

where W^L and $\Omega^L(\boldsymbol{y})$ are given in Eqs. (2.20) and (2.21), respectively.
(b)

$$R^L((m,n),P) = \sum_{\boldsymbol{y} \in W^L} R^L(n;\boldsymbol{y}). \tag{2.28}$$

(B) FMCIA

Next, we provide the FMCIA for computing the reliability of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system. For $j = 1, 2, \ldots, n$ and $\boldsymbol{y}, \boldsymbol{y}' \in W^L$, a transition probability from a state represented by $\boldsymbol{y}'$ to a state represented by $\boldsymbol{y}$ is

$$m^L_{\boldsymbol{y}',\boldsymbol{y}}(j) = \begin{cases} G_j(\boldsymbol{y}), & \text{if } \boldsymbol{y} \in \Omega^L(\boldsymbol{y}'), \\ 0, & \text{otherwise,} \end{cases} \tag{2.29}$$

where $G_j(\boldsymbol{y})$ is given by Eq. (2.22), and then a transition probability matrix is

$$M^L(j) = \left(m^L_{\boldsymbol{y}',\boldsymbol{y}}(j)\right)_{|W^L| \times |W^L|}. \tag{2.30}$$

Then, we can compute the reliability of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system by the following theorem.

Theorem 2.3.

$$R^L((m,n),P) = \boldsymbol{\pi}_0 \left(\prod_{j=1}^{n} M^L(j) \right) \boldsymbol{u}^{\mathsf{T}}, \tag{2.31}$$

where $\boldsymbol{\pi}_0 = (1,0,\ldots,0)_{|W^L|}$, $\boldsymbol{u} = (1,\ldots,1)_{|W^L|}$, and $M^L(j)$ is given by Eq. (2.30).

Theorem 2.3 can be proved similarly to Theorem 2.9.

In the case where the components are independent and identically distributed (the

IID case), all the transition probability matrices are the same as follows:

$$M^L(1) = M^L(2) = \cdots = M^L(n) = M^L. \tag{2.32}$$

A significantly faster method is available to compute $\left(M^L\right)^n$, namely, the fast-matrix-power algorithm [52]. As a result, we can compute the system reliability in the logarithmic order of the number of columns, denoted as n. We provide the following corollary.

Corollary 2.1.

$$R^L((m,n),P) = \boldsymbol{\pi}_0 \left(M^L\right)^n \boldsymbol{u}^{\mathsf{T}}. \tag{2.33}$$

Corollary 2.1 can be proved directly from Theorem 2.3.

2.2.3 Computational Method for a Cir/$(1,2)$-or-$(2,1)$/(m,n):F System

This subsection proposes two kinds of methods for computing the reliability of a Cir/$(1,2)$-or-$(2,1)$/(m,n):F system, i.e., the recursive equation approach and the FM-CIA. Furthermore, for each method, we develop two approaches: (1) ray approach, which focuses on the state of components located at the end ray of the system and increases the number of rays, and (2) circle approach, which focuses on the state of components located at the end circle of the system and increases the number of circles.

(1-A) Recursive equation approach (ray approach)

First, we provide a recursive equation approach for computing the reliability of a Cir/$(1,2)$-or-$(2,1)$/(m,n):F system, which is based on the ray approach. For $j = 2,3,\ldots,n$ and $\boldsymbol{x},\boldsymbol{y} \in \{0,1\}^m$, a probability $R^L(j;\boldsymbol{x};\boldsymbol{y})$ is defined as

$$R^L(j;\boldsymbol{x};\boldsymbol{y}) = \Pr\left\{ \left\{ \prod_{i=1}^{m}\prod_{k=2}^{j}\left(1-Y_{ik}^{(1,2)}\right) \times \prod_{i=2}^{m}\prod_{k=1}^{j}\left(1-Y_{ij}^{(2,1)}\right) = 1 \right\} \bigcap \left\{ \bigcap_{i=1}^{m}\{Z_{i,1}=x_i\} \right\} \bigcap \left\{ \bigcap_{i=1}^{m}\{Z_{ij}=y_i\} \right\} \right\}, \tag{2.34}$$

which shows the reliability of a Lin/$(1,2)$-or-$(2,1)$/(m,j):F system in which the states

of the m components in the 1st column are given by $\boldsymbol{x}$ and the states of the m components in the jth column are given by $\boldsymbol{y}$. Figure 2.3 (b) illustrates this Lin/$(1,2)$-or-$(2,1)/(m,j)$:F system. Then, we can compute the reliability of a Cir/$(1,2)$-or-$(2,1)/(m,n)$:F system by the following theorem.

Theorem 2.4. (a) For $j = 1, 2, \ldots, n$ and $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{y}' \in \{0,1\}^m$,

$$R^L(j;\boldsymbol{x};\boldsymbol{y}) = \begin{cases} G_j(\boldsymbol{y}) \displaystyle\sum_{\boldsymbol{y}' \in \Omega^L(\boldsymbol{y})} R^L(j-1;\boldsymbol{x};\boldsymbol{y}'), & \text{if } j \geq 2 \text{ and } \boldsymbol{x}, \boldsymbol{y} \in W^L, \\ G_j(\boldsymbol{y}), & \text{if } j = 1 \text{ and } \boldsymbol{x}, \boldsymbol{y} \in W^L, \\ 0, & \text{otherwise,} \end{cases} \tag{2.35}$$

where W^L and $\Omega^L(\boldsymbol{y})$ are given in Eqs. (2.20) and (2.21), respectively.
(b)

$$R^C((m,n),P) = \sum_{\boldsymbol{x} \in W^L} \sum_{\boldsymbol{y} \in \Omega^L(\boldsymbol{x})} R^L(n;\boldsymbol{x};\boldsymbol{y}). \tag{2.36}$$

Appendix provides the proof of Theorem 2.4.

(1-B) FMCIA (ray approach)

Next, we provide the FMCIA for computing the reliability of a Cir/$(1,2)$-or-$(2,1)/(m,n)$:F system, which is based on the ray approach. The reliability of a Cir/$(1,2)$-or-$(2,1)/(m,n)$:F system can be obtained by the following theorem.

Theorem 2.5.

$$R^C((m,n),P) = \mathrm{Tr}\left(\prod_{j=1}^{n} M^L(j)\right), \tag{2.37}$$

where $M^L(j)$ is given by Eq. (2.30).

Note that $\mathrm{Tr}(A)$ represents a trace of a matrix A. Theorem 2.5 can be proved similarly to Theorem 2.9.

When the components are independent and identically distributed, we can compute the reliability of a Cir/$(1,2)$-or-$(2,1)/(m,n)$:F system efficiently by the following corollary.

Corollary 2.2.

$$R^C((m,n),P) = \mathrm{Tr}\left(M^L\right)^n, \tag{2.38}$$

where M^L is given by Eq. (2.32).

From Theorem 2.5, Corollary 2.2 is immediately proved.

(2-A) Recursive equation approach (circle approach)

First, we provide a recursive equation approach for computing the reliability of a Cir/$(1,2)$-or-$(2,1)$/(m,n):F system, which is based on the circle approach. For $i = 1,2,\ldots,m$ and $\boldsymbol{g} \in \{0,1\}^n$, a probability $R^C(i;\boldsymbol{g})$ is defined as

$$R^C(i;\boldsymbol{g}) = \Pr\left\{\left\{\prod_{k=1}^{i}\prod_{j=1}^{n}\left(1-Y_{kj}^{(1,2)}\right) \times \prod_{k=2}^{i}\prod_{j=1}^{n}\left(1-Y_{kj}^{(2,1)}\right) = 1\right\} \bigcap \left\{\bigcap_{j=1}^{n}\{Z_{kj} = g_j\}\right\}\right\}, \tag{2.39}$$

which shows the reliability of a Cir/$(1,2)$-or-$(2,1)$/(i,n):F system in which the states of the n components in the ith circle are given by $\boldsymbol{g}$. Figure 2.4 (a) illustrates the Cir/$(1,2)$-or-$(2,1)$/(i,n):F system in which the states of the n components in the ith circle are given by $\boldsymbol{g}$. Then, we can compute the reliability of a Cir/$(1,2)$-or-$(2,1)$/(m,n):F system by the following theorem.

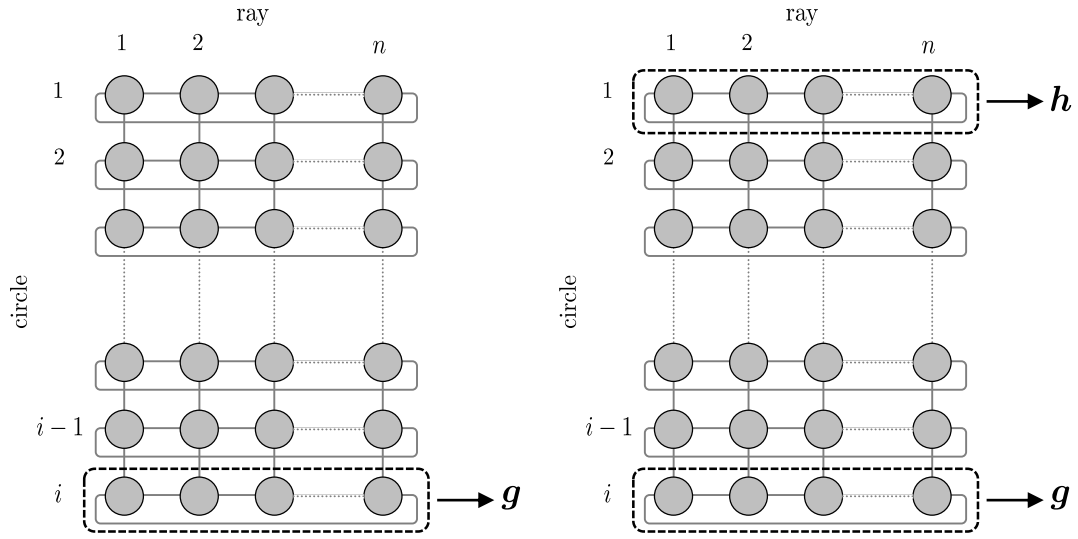
Theorem 2.6. (a) For $i = 1,2,\ldots,m$ and $\boldsymbol{g},\boldsymbol{g}' \in \{0,1\}^n$,

$$R^C(i;\boldsymbol{g}) = \begin{cases} F_i(\boldsymbol{g}) \displaystyle\sum_{\boldsymbol{y}' \in \Omega^C(\boldsymbol{g})} R^C(i-1;\boldsymbol{g}'), & \text{if } i \geq 2 \text{ and } \boldsymbol{g} \in W^C, \\ F_i(\boldsymbol{g}), & \text{if } i = 1 \text{ and } \boldsymbol{g} \in W^C, \\ 0, & \text{otherwise,} \end{cases} \tag{2.40}$$

where W^C and $\Omega^C(\boldsymbol{g})$ are given in Eqs. (2.23) and (2.24), respectively.
(b)

$$R^C((m,n),P) = \sum_{\boldsymbol{y} \in W^C} R^C(m;\boldsymbol{g}). \tag{2.41}$$

We can prove Theorem 2.6 in the same manner as Theorem 2.4.



(a) Definition of $R^C(i;\boldsymbol{g})$ (b) Definition of $R^C(i;\boldsymbol{h};\boldsymbol{g})$

Figure 2.4: Cir/$(1,2)$-or-$(2,1)$/(i,n):F systems with a condition.

(2-B) FMCIA (circle approach)

Next, we provide the FMCIA for computing the reliability of a Cir/$(1,2)$-or-$(2,1)$/(m,n):F system, which is based on the circle approach. For $i = 1,2,\ldots,m$ and $\boldsymbol{g},\boldsymbol{g}' \in W^C$, a transition probability from a state represented by $\boldsymbol{g}'$ to a state represented by $\boldsymbol{g}$ is

$$m^C_{\boldsymbol{g}',\boldsymbol{g}}(i) = \begin{cases} F_i(\boldsymbol{g}), & \text{if } \boldsymbol{g} \in \Omega^C(\boldsymbol{g}'), \\ 0, & \text{otherwise,} \end{cases} \tag{2.42}$$

where $F_i(\boldsymbol{g})$ is given by Eq. (2.25), and then a transition probability matrix is

$$M^C(i) = \left(m^C_{\boldsymbol{g}',\boldsymbol{g}}(i)\right)_{|W^C|\times|W^C|}. \tag{2.43}$$

Then, we can compute the reliability of a Cir/$(1,2)$-or-$(2,1)$/(m,n):F system by the following theorem.

Theorem 2.7.

$$R^C((m,n),P) = \boldsymbol{\pi}_0 \left(\prod_{i=1}^{m} M^C(i)\right) \boldsymbol{u}^{\mathsf{T}}, \tag{2.44}$$

where $\boldsymbol{\pi}_0 = (1, 0, \ldots, 0)_{|W^C|}$, $\boldsymbol{u} = (1, \ldots, 1)_{|W^C|}$, and $M^C(i)$ is given by Eq. (2.43).

Theorem 2.7 can be proved similarly to Theorem 2.9.

In the IID case, all the transition probability matrices are the same as follows:

$$M^C(1) = M^C(2) = \cdots = M^C(m) = M^C. \tag{2.45}$$

Thus, we can compute the reliability of a Cir/$(1,2)$-or-$(2,1)$/(m,n):F system efficiently by the following corollary.

Corollary 2.3.

$$R^C((m,n),P) = \boldsymbol{\pi}_0 \left(M^C\right)^m \boldsymbol{u}^{\mathsf{T}}. \tag{2.46}$$

The proof is straightforward from Theorem 2.7.

2.2.4 Computational Method for a Tor/$(1,2)$-or-$(2,1)$/(m,n):F System

This subsection proposes both the recursive equation approach and the FMCIA for computing the reliability of Tor/$(1,2)$-or-$(2,1)$/(m,n):F systems.

(A) Recursive equation approach

First, we provide a recursive equation approach for computing the reliability of a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system. For $i = 2, 3, \ldots, m$ and $\boldsymbol{h}, \boldsymbol{g} \in \{0,1\}^n$, a probability $R^C(i; \boldsymbol{h}; \boldsymbol{g})$ is defined as

$$R^C(i;\boldsymbol{h};\boldsymbol{g}) = \Pr\left\{ \left\{ \prod_{k=1}^{i}\prod_{j=1}^{n}\left(1 - Y_{kj}^{(1,2)}\right) \times \prod_{k=2}^{i}\prod_{j=1}^{n}\left(1 - Y_{kj}^{(2,1)}\right) = 1 \right\} \right.$$
$$\left. \bigcap \left\{ \bigcap_{j=1}^{n} \{Z_{1j} = h_j\} \right\} \bigcap \left\{ \bigcap_{j=1}^{n} \{Z_{kj} = g_j\} \right\} \right\}, \tag{2.47}$$

which shows the reliability of a Cir/$(1,2)$-or-$(2,1)$/(i,n):F system in which the states of the n components in the 1st circle are given by $\boldsymbol{h}$ and the states of the n components in the ith circle are given by $\boldsymbol{g}$. Figure 2.4 (a) illustrates this Cir/$(1,2)$-or-$(2,1)$/(i,n):F system. Then, we can compute the reliability of a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system by the following theorem.

Theorem 2.8. (a) For $i = 1, 2, \ldots, m$ and $\boldsymbol{h}, \boldsymbol{g}, \boldsymbol{g}' \in \{0,1\}^n$,

$$R^C(i; \boldsymbol{h}; \boldsymbol{g}) = \begin{cases} F_i(\boldsymbol{g}) \displaystyle\sum_{\boldsymbol{g}' \in \Omega^C(\boldsymbol{g})} R^C(i-1; \boldsymbol{h}; \boldsymbol{g}'), & \text{if } i \geq 2 \text{ and } \boldsymbol{h}, \boldsymbol{g} \in W^C, \\ F_i(\boldsymbol{g}), & \text{if } i = 1 \text{ and } \boldsymbol{h}, \boldsymbol{g} \in W^C, \\ 0, & \text{otherwise,} \end{cases} \tag{2.48}$$

where W^C and $\Omega^C(\boldsymbol{g})$ are given in Eqs. (2.23) and (2.24), respectively.
(b)

$$R^T((m,n), P) = \sum_{\boldsymbol{h} \in W^C} \sum_{\boldsymbol{g} \in \Omega^C(\boldsymbol{h})} R^C(m; \boldsymbol{h}; \boldsymbol{g}). \tag{2.49}$$

We can prove Theorem 2.8 in the same manner as we did Theorem 2.4.

(B) FMCIA

Next, we provide the FMCIA for computing the reliability of a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system. The reliability of a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system can be obtained by the following theorem.

Theorem 2.9.

$$R^T((m,n), P) = \mathrm{Tr}\left(\prod_{i=1}^{m} M^C(i)\right), \tag{2.50}$$

where $M^C(i)$ is given by Eq. (2.43).

Appendix provides the proof of Theorem 2.9.

When the components are independent and identically distributed, we can compute the reliability of a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system efficiently by the following corollary.

Corollary 2.4.

$$R^T((m,n), P) = \mathrm{Tr}\left(M^C\right)^m, \tag{2.51}$$

where M^C is given by Eq. (2.45).

Corollary 2.4 follows immediately from Theorem 2.9.

Table 2.6: Approaches for computing the reliability of connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F lattice systems.

	Recursive equation approach	FMCIA	
	(INID & IID cases)	(INID case)	(IID case)
Linear-type	Theorem 2.2 [64]	Theorem 2.3	Corollary 2.1
Circular-type (ray approach)	Theorem 2.4	Theorem 2.5	Corollary 2.2
Circular-type (circle approach)	Theorem 2.6	Theorem 2.7	Corollary 2.3
Toroidal-type	Theorem 2.8	Theorem 2.9	Corollary 2.4

Table 2.6 summarizes the approaches for computing the reliability of connected-$(1,2)$-or-$(2,1)$-out-of-(m,n):F lattice systems.

2.2.5 Efficiency Investigation

In this subsection, we compare the recursive equation approach and the FMCIA in terms of the actual computation time. Two computational experiments were conducted for a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system and a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system. In each experiment, both methods are compared in the case where the components are independent but non-identically distributed (INID case) and the IID case. In the INID case, the component reliabilities P are given, for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, by

$$p_{ij} = \begin{cases} 0.999, & \text{if } i+j \text{ is odd,} \\ 0.995, & \text{if } i+j \text{ is even.} \end{cases}$$

In the IID case, the component reliabilities is given by $p_{ij} = 0.999$ for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$.

Table 2.7 shows the comparison of the computation time required for a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system in the INID case. First, we confirmed that both methods gave the same system reliabilities. The computation times of both methods tend to increase as the number of rows, denoted as m, increases, as shown in Table 2.7. It is obvious from Table 2.7 that the recursive equation approach [64] is more efficient in most cases. Table 2.8 shows the comparison of the computation time required for a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system in the IID case. When the number of columns, denoted as n, is relatively small and the number of rows is large, the recursive equation approach [64]

Table 2.7: Comparison of computation times of the recursive equation approach and the FMCIA for a Lin/$(1,2)$-or-$(2,1)/(m,n)$:F system in the INID case (sec.).

m	n	(A) Recursive equation approach	(B) FMCIA	(B)/(A) (%)	$R^L((m,n),P)$
10	10	0.040	0.039	97.97	0.9991
10	50	0.055	0.055	100.50	0.9954
10	100	0.074	0.082	110.82	0.9907
15	10	0.531	0.484	91.18	0.9986
15	50	0.666	0.889	133.44	0.9929
15	100	0.920	1.112	120.83	0.9858
20	10	37.597	39.689	105.56	0.9982
20	50	49.205	58.860	119.62	0.9905
20	100	64.725	80.040	123.66	0.9810

Table 2.8: Comparison of computation times of the recursive equation approach and the FMCIA for a Lin/$(1,2)$-or-$(2,1)/(m,n)$:F system in the IID case (sec.).

m	n	(A) Recursive equation approach	(B) FMCIA	(B)/(A) (%)	$R^L((m,n),P)$
14	250	0.752	0.583	77.51	0.9933
14	500	1.213	0.598	49.28	0.9866
14	1000	2.098	0.675	32.17	0.9735
14	2000	3.864	0.654	16.93	0.9476
17	250	7.573	14.959	197.53	0.9918
17	500	13.263	15.974	120.45	0.9837
17	1000	24.791	17.384	70.12	0.9676
17	2000	47.741	18.520	38.79	0.9363

is more efficient, whereas the FMCIA outperforms the recursive equation approach [64] when the number of columns is large. Overall, we obtained the following findings. In the INID case, we should use the recursive equation approach; in the IID case, if the number of columns is large, we should use the FMCIA. Note that we can obtain the same result even if we swap m and n due to the system symmetry.

Table 2.9 shows the comparison of the computation time required for a Tor/$(1,2)$-or-$(2,1)/(m,n)$:F system in the INID case. Here, N/A means that a method failed to compute the system reliability in 12 hours. We can see in Table 2.9 that the FMCIA can compute the system reliability in a shorter time than the recursive equation approach in all the cases. Table 2.10 shows the comparison of the computation time

Table 2.9: Comparison of computation times of the recursive equation approach and the FMCIA for a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system in the INID case (sec.).

n	m	(A) Recursive equation approach	(B) FMCIA	(B)/(A) (%)	$R^T((m,n),P)$
10	10	0.258	0.035	13.45	0.9990
10	50	0.642	0.052	8.13	0.9951
10	100	1.055	0.082	7.78	0.9901
15	10	45.358	0.902	1.99	0.9984
15	50	222.912	2.985	1.34	0.9922
15	100	458.170	5.643	1.23	0.9845

Table 2.10: Comparison of computation times of the recursive equation approach and the FMCIA for a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system in the IID case (sec.).

n	m	(A) Recursive equation approach	(B) FMCIA	(B)/(A) (%)	$R^T((m,n),P)$
14	250	295.064	0.467	0.16	0.9930
14	500	599.108	0.490	0.08	0.9861
14	1000	1181.618	0.492	0.04	0.9725
14	2000	2354.054	0.484	0.02	0.9457
17	250	17080.153	9.579	0.06	0.9916
17	500	31205.300	10.226	0.03	0.9832
17	1000	N/A	10.848	—	0.9667
17	2000	N/A	11.771	—	0.9345

required for a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system in the IID case. From Table 2.10, the computation time of the FMCIA does not increase with m, which illustrates the efficiency of the FMCIA. Thus, within the scope of the experiment, we can conclude that the FMCIA outperforms the recursive equation approach. According to the above results, we suggest using the FMCIA for efficiently computing the reliability of the Tor/$(1,2)$-or-$(2,1)$/(m,n):F system.

Remark

Here, we consider how to compute a recursive equation for system reliability. As noted above, a recursive equation can be expressed as not only a recursive form but also iteration, and the iteration is preferable because of the efficiency. A recursive equation can be rewritten as a matrix form and computed by the multiplications of the matrices,

which is called the matrix-based method in this thesis. Note that the matrix-based method generates a matrix from the recursive equation, whereas the FMCIA [60] generates a matrix directly. However, these methods compute the system reliability by the same form such as Eq. (2.31). Hence, we compare the computation times of (A) the recursive equation approach [58] and (B) a matrix-based method in a Lin/(r,s)/(m,n):F system. We consider only the IID case and $r = m-1$ for simplicity. That is, we consider a Lin/(r,s)/$(r+1,n)$:F system with the common component reliability p and unreliability q, where $p+q=1$.

(A) Recursive equation approach First, we present the recursive equation [58] in the case of $r = m-1$. Let $R^L(j;(g_1,g_2))$ be the reliability of a Lin/(r,s)/$(r+1,j)$:F system in which all the components fail in the $r \times g_1$ rectangle with four corners $(1,j)$, $(1,j-g_1+1)$, $(r,j-g_1+1)$, and (r,j) and the $r \times g_2$ rectangle with four corners $(2,j)$, $(2,j-g_2+1)$, $(r+1,j-g_2+1)$, and $(r+1,j)$. Figure 2.5 exemplifies a Lin/$(2,2)$/$(3,j)$:F system when $\boldsymbol{g} = (2,1)$. For the case of $g_1 = 0,1,\ldots,s-1$ and $g_2 = 0,1,\ldots,s-1$, $R^L(j;(g_1,g_2))$ can be computed as follows:

$$R^L(j;(0,0)) = \alpha \times \sum_{(g_1,g_2)\in\{0,1,\ldots,s-1\}^2} R^L(j-1;(g_1,g_2)), \tag{2.52}$$

where $\alpha = 1-(2pq^r+q^{r+1})$. For $g_1 = 1,2,\ldots,s-1$, we have

$$R^L(j;(g_1,0)) = \beta \times \sum_{g_2\in\{0,1,\ldots,s-1\}} R^L(j-1;(g_1-1,g_2)), \tag{2.53}$$

and for $g_2 = 1,2,\ldots,s-1$, we see that

$$R^L(j;(0,g_2)) = \beta \times \sum_{g_1\in\{0,1,\ldots,s-1\}} R^L(j-1;(g_1,g_2-1)), \tag{2.54}$$

where $\beta = pq^r$. For $g_1 = 1,2,\ldots,s-1$ and $g_2 = 1,2,\ldots,s-1$, we have

$$R^L(j;(g_1,g_2)) = \gamma \times R^L(j-1;(g_1,g_2)), \tag{2.55}$$

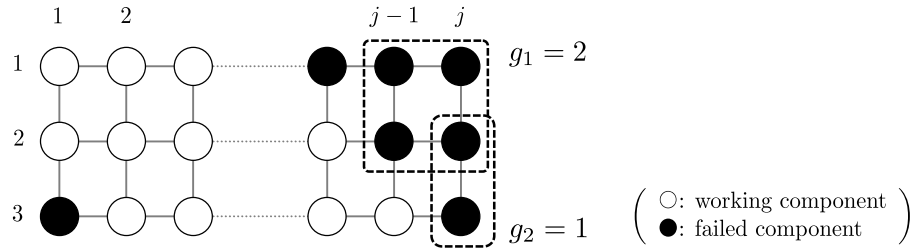


Figure 2.5: Lin/$(2,2)$/$(3,j)$:F system when $\boldsymbol{g} = (2,1)$.

where $\gamma = q^{r+1}$. Using Eqs. (2.52)–(2.55), the reliability of a Lin/(r,s)/(m,n):F system can be computed by

$$R^L((r,s),(r+1,n),P) = \sum_{(g_1,g_2)\in\{0,1,\ldots,s-1\}^2} R^L(n;(g_1,g_2)). \tag{2.56}$$

(B) Matrix-based method Second, we provide a matrix-based method. From Eqs. (2.52)–(2.56), we can compute the reliability of a Lin/(r,s)/(m,n):F system as follows:

$$R^L((r,s),(r+1,n),P) = \boldsymbol{u}\,(M_s)^n\,\boldsymbol{\pi}_0^{\mathsf{T}}, \tag{2.57}$$

where $\boldsymbol{u} = (1,\ldots,1)_d$, $\boldsymbol{\pi}_0 = (1,0,\ldots,0)_d$, and the general form of the matrix M_s can be expressed, for $s = 2,3,\ldots,n-1$, as

$$M_s = \begin{pmatrix} A_s & A_{s-1} & \cdots & A_2 & \alpha \\ B_s & & & & \\ & B_{s-1} & & O & \\ & & \ddots & & \\ O & & & B_2 & 0 \end{pmatrix}_{d\times d}, \tag{2.58}$$

Table 2.11: Comparison of computation times of (A) Recursive equation approach and (B) Matrix-based method ($(r,s)=(4,4), p=0.05$).

(m,n)	$(5,10)$	$(5,10^2)$	$(5,10^3)$	$(5,10^4)$	$(5,10^5)$
(A) Recursive equation approach (sec.)	0.0424	0.0487	0.0552	0.1679	0.8362
(B) Matrix-based method (sec.)	0.0093	0.0109	0.0124	0.0110	0.0114
(B)/(A) (%)	21.88	22.37	22.46	6.55	1.36
$R^L((r,s),(r+1,n),P)$	0.9998	0.9973	0.9728	0.7587	0.0632

Table 2.12: Comparison of computation times of (A) Recursive equation approach and (B) Matrix-based method ($(m,n)=(5,1000), p=0.05$).

(r,s)	(4,20)	(4,30)	(4,40)	(4,50)	(4,60)
(A) Recursive equation approach (sec.)	0.2087	0.3472	0.7159	0.7787	1.0112
(B) Matrix-based method (sec.)	0.0158	0.0639	0.2243	1.0332	2.1626
(B)/(A) (%)	7.57	18.40	31.33	132.68	213.87
$R^L((r,s),(r+1,n),P)$	0.006	0.513	0.915	0.988	0.998

where, for $l=2,3,\ldots,s$,

$$A_l = \begin{pmatrix} \alpha & \alpha & \alpha & \cdots & \alpha \\ & & O_{(s-l)\times l} & & \\ 2\beta & \beta & \beta & \cdots & \beta \\ & \beta & & O & \\ & & \ddots & & \\ & O & & \beta & 0 \end{pmatrix}_{s\times l}, \qquad B_l = \begin{pmatrix} \gamma & & O \\ & \ddots & \\ O & & \gamma \\ & & 0 \end{pmatrix}_{(l-1)\times l},$$

$$d = 1 + 2 + \cdots + (s-1) = \frac{1}{2}s(s+1), \tag{2.59}$$

and O means a zero matrix.

We investigate the efficiency of the recursive equation approach and the matrix-based method through numerical experiments. Tables 2.11 and 2.12 show the comparison of the computation time required for the cases of $(r,s)=(4,4)$ and $(m,n)=(5,1000)$, respectively. We confirmed that both methods provided the same system

reliabilities. First, we investigate the efficiency of both methods in the case where s is fixed. As can be seen from Table 2.11, the matrix-based method is more efficient, especially for systems with large n. The reason is that by utilizing the fast-matrix-power algorithm [52], the reliability can be obtained in logarithmic time for the parameter n. Second, we investigate the efficiency of both methods in the case where n is fixed. It is clear from Table 2.12 that, when $s \leq 40$, the matrix-based method performs better than the recursive equation approach, whereas when $s \geq 50$, the recursive equation approach is more efficient. This is because as the parameter s is large, the size of matrix (M_s) is also large, and as a result, the required computation time typically grows rapidly. However, M_s is a matrix in which many or most of the elements have a value of zero, which is called a sparse matrix. The multiplications of this sparse matrix are easy to compute by using computing software, e.g., MATLAB. Overall, we can conclude that the matrix-based method outperforms the recursive equation approach for computing the reliability of a Lin/(r, s)/$(r + 1, n)$:F system with common component reliability p when n is large and s is small.

2.3 Bounds for the Reliability of a Lin/$(1, 2)$-or-$(2, 1)/(m, n)$:F System

In this section, we consider upper and lower bounds for the reliability of a Lin/$(1, 2)$-or-$(2, 1)/(m, n)$:F system. One can make a trade-off between the computational effort and the quality of the bounds (closeness to the exact value). Several useful and simple bounds were reported, e.g., [66–68], whereas the bounds that require the computational burden but are tighter have not been sufficiently discussed. Tighter bounds can appropriately evaluate the system reliability compared with the existing bounds. Thus, this section derives tighter bounds for the reliability of a Lin/$(1, 2)$-or-$(2, 1)/(m, n)$:F system at the expense of the computational effort.

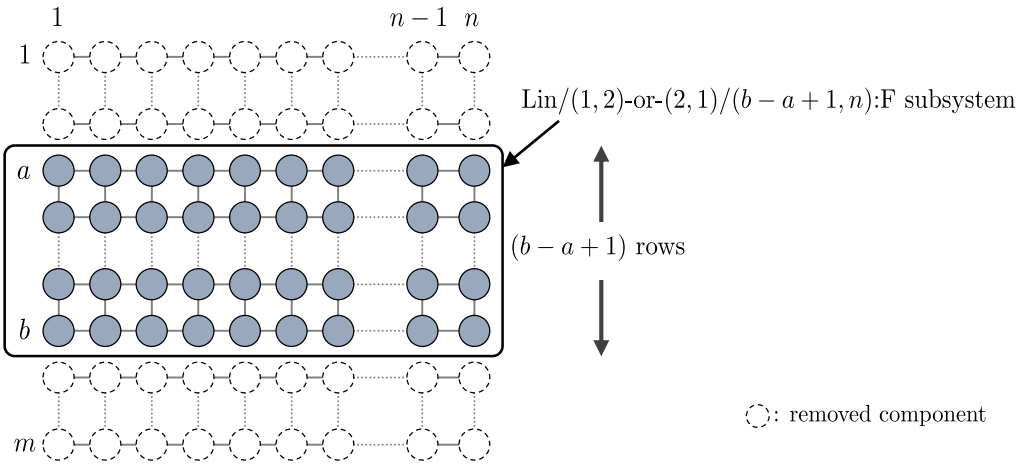


Figure 2.6: Lin/$(1,2)$-or-$(2,1)$/$(b-a+1,n)$:F subsystem.

2.3.1 Derivation of Upper and Lower Bounds

First, we introduce some notation. Recall that the reliabilities of the components in a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system is

$$P = (p_{ij})_{1\le i\le m, 1\le j\le n},$$

where p_{ij} is the reliability of component (i,j) for $i = 1,2,\ldots,m$ and $j = 1,2,\ldots,n$. We consider a sub-matrix obtainable by removing some of the rows from the matrix P, which is defined as follows:

$$P_{[a,b]} = (p_{ij})_{a\le i\le b, 1\le j\le n}.$$

Accordingly, $R^L((b-a+1,n), P_{[a,b]})$ denotes the reliability of a Lin/$(1,2)$-or-$(2,1)$/$(b-a+1,n)$:F subsystem, where $1 \le a < b \le m$. Here, a Lin/$(1,2)$-or-$(2,1)$/$(b-a+1,n)$:F subsystem is obtained by removing rows $1,2,\ldots,a-1,b+1,b+2,\ldots,m$ from a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system as shown in Fig. 2.6.

In addition, we define probabilities $S^{(\gamma)}((b-a+1,n), P_{[a,b]})$ as

$$S^{(\gamma)}((b-a+1,n), P_{[a,b]}) = \Pr\left\{ \prod_{i=a, i\ne\gamma}^{b} \prod_{j=2}^{n} \left(1 - Y_{ij}^{(1,2)}\right) \times \prod_{i=a+1}^{b} \prod_{j=1}^{n} \left(1 - Y_{ij}^{(2,1)}\right) = 1 \right\}, \tag{2.60}$$

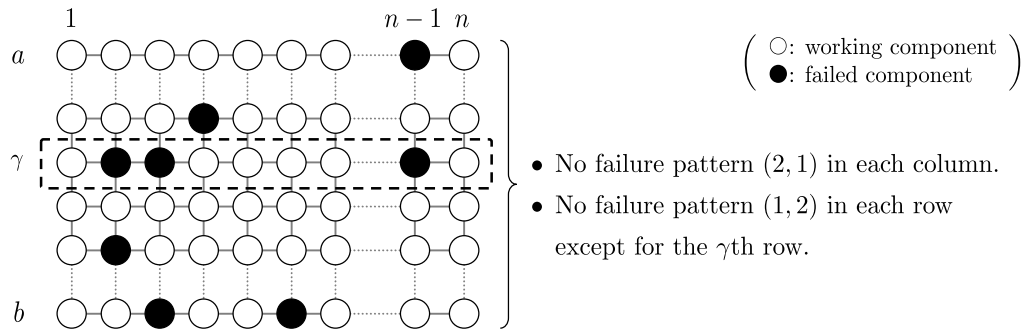


Figure 2.7: Definition of $S^{(\gamma)}((b-a+1,n),P_{[a,b]})$.

where $a \leq \gamma \leq b$. In words, $S^{(\gamma)}((b-a+1,n),P_{[a,b]})$ means a probability that a Lin/$(1,2)$-or-$(2,1)$/$(b-a+1,n)$:F subsystem has no failure pattern $(2,1)$ in each column and no failure pattern $(1,2)$ in each row except for the γth row for $a \leq x \leq b, a \neq b$ as shown in Fig. 2.7. $S^{(\gamma)}((b-a+1,n),P_{[a,b]})$ is used to compute an upper bound that is derived in this section.

We explain the idea behind the derived bounds for the reliability of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system. From Eq. (1.18), we can obtain the upper bound for the system reliability as follows:

$$R^L((b,n),P_{[1,b]}) \times R^L((m-b,n),P_{[b+1,m]}). \tag{2.61}$$

Let $D_b([1,m],n)$ denote the difference between the exact system reliability and its upper bound in Eq. (2.61), and then, $D_b([1,m],n)$ can be expressed, for $1 \leq a < b \leq m$, as

$$R^L((m,n),P) = R^L((b,n),P_{[1,b]}) \times R^L((m-b,n),P_{[b+1,m]}) - D_b([1,m],n), \tag{2.62}$$

or equivalently,

$$D_b([1,m],n) = R^L((b,n),P_{[1,b]}) \times R^L((m-b,n),P_{[b+1,m]}) - R^L((m,n),P). \tag{2.63}$$

Here, the difference $D_b([l_1,l_2],n)$ represents the probability that a Lin/$(1,2)$-or-$(2,1)$/(l_2-l_1+1,n):F system has at least one failure pattern $(2,1)$ in rows b and $b+1$ and no failure patterns in any other positions. This is graphically presented in Fig. 2.8.

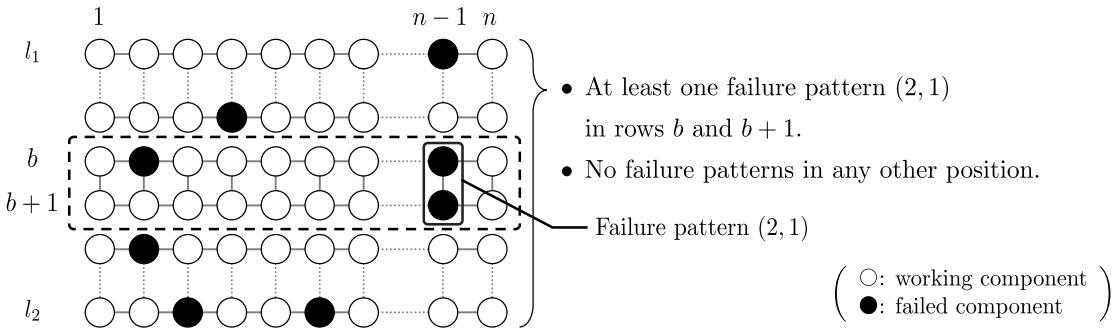


Figure 2.8: Definition of $D_b([l_1, l_2], n)$.

The reliabilities of the subsystems ($R^L((b, n), P_{[1,b]})$ and $R^L((m - b, n), P_{[b+1,m]})$) can be computed by the existing method, such as Theorem 2.2 [64] and Theorem 2.3, whereas the computation of $D_b([1, m], n)$ in Eq. (2.62) is often difficult. Hence, we present the upper and lower bounds for the probability $D_b([1, m], n)$. If we substitute the upper bound of $D_b([1, m], n)$ into $D_b([1, m], n)$ in Eq. (2.62), instead of the exact value, then the value of the right-hand side is less than the exact value and exhibits the negative error. Consequently, the lower bounds for the system reliability can be obtained. Meanwhile, if we substitute the lower bound of $D_b([1, m], n)$ into $D_b([1, m], n)$ in Eq. (2.62), then the value of the right-hand side is always greater than the exact value and exhibits the positive error. Consequently, the upper bounds for the system reliability can be obtained.

We present the following theorem.

Theorem 2.10. The lower and upper bounds $(LB_D(b, l_1, l_2), UB_D(b, l_1, l_2))$ are given by

$$\begin{aligned} LB_D(b, l_1, l_2) =& R^L((b, n), P_{[1,b]}) \times R^L((m - b, n), P_{[b+1,m]}) \\ & - R^L((l_1 - 1, n), P_{[1,l_1-1]}) \times D_b([l_1, l_2], n) \times R^L((m - l_2, n), P_{[l_2+1,m]}), \end{aligned} \tag{2.64}$$

and

$$\begin{aligned} UB_D(b, l_1, l_2) =& R^L((b, n), P_{[1,b]}) \times R^L((m - b, n), P_{[b+1,m]}) \\ & - S^{(l_1)}((l_1, n), P_{[1,l_1]}) \times D_b([l_1, l_2], n) \times S^{(l_2)}((m - l_2 + 1, n), P_{[l_2,m]}), \end{aligned} \tag{2.65}$$

where

$$D_b([l_1,l_2],n) = R^L((b-l_1+1,n),P_{[l_1,b]}) \times R^L((l_2-b-1,n),P_{[b+1,l_2]}) - R^L((l_2-l_1+1,n),P_{[l_1,l_2]}). \quad (2.66)$$

Appendix provides the proof of Theorem 2.10.

Although Theorem 2.2 [64] and Theorem 2.3 can provide the reliability of the subsystem $R^L((b-a+1,n),P_{[a,b]}))$, a method for computing $S^{(\gamma)}((b-a+1,n),P_{[a,b]})$, which is defined in Eq. (2.60), has not been reported. Thus, we propose a recursive equation approach for efficiently computing $S^{(\gamma)}((b-a+1,n),P_{[a,b]})$. First, we define the following sets:

$$W^S = \left\{ \boldsymbol{y} \;\middle|\; \sum_{i=a+1}^{b} y_{i-1}y_i = 0 \right\}, \quad (2.67)$$

and

$$\Omega^S(\boldsymbol{y},\gamma) = \left\{ \boldsymbol{y}' \in W^S \;\middle|\; \sum_{i=a,i\neq\gamma}^{b} y_i y_i' = 0 \right\}. \quad (2.68)$$

The following proposition can provide $S^{(\gamma)}((b-a+1,n),P_{[a,b]})$.

Proposition 2.1. (a) For $j=1,2,\ldots,n$ and $\boldsymbol{y},\boldsymbol{y}' \in \{0,1\}^{(b-a+1)}$,

$$S(j;\boldsymbol{y}) = \begin{cases} G_j(\boldsymbol{y}) \displaystyle\sum_{\boldsymbol{y}'\in\Omega^S(\boldsymbol{y},\gamma)} S(j-1;\boldsymbol{y}'), & \text{if } j\geq 2 \text{ and } \boldsymbol{y}\in W^S, \\ G_j(\boldsymbol{y}), & \text{if } j=1 \text{ and } \boldsymbol{y}\in W^S, \\ 0, & \text{otherwise,} \end{cases} \quad (2.69)$$

where W^S and $\Omega^S(\boldsymbol{y},\gamma)$ are given in Eqs. (2.67) and (2.68), respectively, and

$$G_j(\boldsymbol{y}) = \prod_{i=a}^{b} {p_{ij}}^{1-y_i}(1-p_{ij})^{y_i}.$$

(b)

$$S^{(\gamma)}((b-a+1,n),P_{[a,b]}) = \sum_{\boldsymbol{y}\in W^S} S(n;\boldsymbol{y}). \tag{2.70}$$

This proposition can be proved similarly to Theorem 2.4.

2.3.2 Numerical Experiment

In this subsection, we conduct the numerical experiments to compare the derived bounds with the existing bounds. For $i = 1,2,\ldots,m$ and $j = 1,2,\ldots,n$, the component reliabilities P are given by

$$p_{ij} = \begin{cases} \alpha, & \text{if } i+j \text{ is odd,} \\ \beta, & \text{if } i+j \text{ is even.} \end{cases}$$

When we compute the bounds in Theorem 2.10, the values and the computation times may be affected by changing the parameters: b, l_1, and l_2, which determine the size of subsystems. Note that we should obtain the greatest of the lower bounds and the smallest of the upper bounds.

First, we investigate the effects of the parameter b as a preliminary experiment. Here, we set $l_1 = b - w$ and $l_2 = b + w$. Table 2.13 shows the number of the rows of the subsystems (b and $m-b$), the values of the derived bounds, and the times required for obtaining the upper and lower bounds for $w \in \{5,6,7\}$. From Table 2.13, we can see that the parameter b affects the computation times although it hardly changes the obtained values. This suggests that both b and $m-b$ should be small simultaneously to compute the values in a shorter time. Thus, we set

$$b = \lfloor m/2 \rfloor, \tag{2.71}$$

where $\lfloor a \rfloor$ denotes the greatest integer less than or equal to a.

Second, we investigate the effects of the parameters l_1 and l_2 as a preliminary experiment. The experimental results are listed in Table 2.14, including the number of the rows of the subsystems ($l_1 - 1$, $l_2 - l_1 + 1$, $m - l_2$ for the lower bound and l_1, $l_2 - l_1 + 1$, $m - l_2 + 1$ for the upper bound), the values of the derived bounds, and the times required for obtaining the upper and lower bounds. The result in Table 2.14 shows

Table 2.13: Bounds for the reliability of the Lin/$(1,2)$-or-$(2,1)$/$(36,100)$:F system and their computation times when $(\alpha,\beta)=(0.995,0.999)$ for various b.

w	b	$m-b$	Lower bound ($LB_D(b,l_1,l_2)$)	Upper bound ($UB_D(b,l_1,l_2)$)	time [s]
5	15	21	0.965594	0.965595	96.07
	16	20	0.965594	0.965595	37.43
	17	19	0.965594	0.965595	16.83
	18	18	0.965594	0.965595	11.28
6	15	21	0.965594	0.965595	92.52
	16	20	0.965594	0.965595	36.04
	17	19	0.965594	0.965595	16.31
	18	18	0.965594	0.965595	10.98
7	15	21	0.965594	0.965595	91.52
	16	20	0.965594	0.965595	36.13
	17	19	0.965594	0.965595	16.48
	18	18	0.965594	0.965595	11.24

that the parameters hardly change the obtained values, whereas the computation times are affected by the parameters. From the results, basically, we suggest minimizing the maximum number of rows of the subsystems to compute the values in a shorter time. Therefore, we set

$$l_1 = \lfloor m/3 \rfloor, \tag{2.72}$$

and

$$l_2 = m - l_1 + 1. \tag{2.73}$$

Third, we compare the derived bounds $LB_D(b,l_1,l_2)$ and $UB_D(b,l_1,l_2)$ with the existing upper bound ($UB_{FK}^{(G)}$) and lower bounds (LB_{EP} and $LB_{EP}^{(G)}$). Boutsikas and Koutras [67] compared some existing upper and lower bounds for the reliability of a Lin/(r,s)/(m,n):F system and pointed out that the best available bounds were the upper bound ($UB_{FK}^{(G)}$) and the lower bound ($LB_{EP}^{(G)}$). We present the upper bound of Boutsikas and Koutras [67] for the reliability of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system

Table 2.14: Bounds for the reliability of the Lin/$(1,2)$-or-$(2,1)$/$(36,100)$:F system and their computation times when $(\alpha,\beta)=(0.995,0.999)$ for various l_1 and l_2.

Lower bound ($LB_D(b,l_1,l_2)$)				Upper bound ($UB_D(b,l_1,l_2)$)				
l_1-1	l_2-l_1+1	$m-l_2$	value	l_1	l_2-l_1+1	$m-l_2+1$	value	time [s]
8	20	8	0.965594	9	20	9	0.965595	40.98
9	18	9	0.965594	10	18	10	0.965595	14.77
10	16	10	0.965594	11	16	11	0.965595	10.75
11	14	11	0.965594	12	14	12	0.965595	10.19
12	12	12	0.965594	13	12	13	0.965595	10.09
13	10	13	0.965594	14	10	14	0.965595	10.42
14	8	14	0.965594	15	8	15	0.965595	11.18
15	6	15	0.965594	16	6	16	0.965595	12.70
16	4	16	0.965594	17	4	17	0.965595	16.80

as follows:

$$UB_{FK}^{(G)}=\prod_{i=1}^{m-1}\prod_{j=1}^{n-1}(1-\alpha_{ij})\times\prod_{i=1}^{m-1}(1-\beta_i)\times\prod_{j=1}^{n-1}(1-\gamma_j), \tag{2.74}$$

where

$$\begin{aligned}
\alpha_{ij}&=p_{i+1,j-1}p_{i,j-1}p_{i-1,j}p_{i-1,j+1}(q_{ij}q_{i,j+1}+q_{ij}q_{i+1,j}-q_{ij}q_{i,j+1}q_{i+1,j}),\\
\beta_i&=p_{i-1,n}p_{i,n-1}p_{i+1,n-1}q_{in}q_{i+1,n},\\
\gamma_j&=p_{m,j-1}p_{m-1,j}p_{m-1,j+1}q_{mj}q_{m,j+1},
\end{aligned}$$

Note that $p_{ij}=1$ if $i=0$ or $j=0$ for convenience, and q_{ij} is the unreliability of component (i,j) ($q_{ij}=1-p_{ij}$). Also, applying Eq. (1.17) to the Lin/$(1,2)$-or-$(2,1)$/(m,n):F system yields

$$LB_{EP}^{(G)}(b)=R^L((b,n),P_{[1,b]})\times S^{(b)}((m-b+1,n),P_{[b,m]}), \tag{2.75}$$

where $b=\lfloor m/2\rfloor$. In addition, LB_{EP}, which is a well-known simple lower bounds, is computed from Eq. (1.16).

Table 2.15 compares the derived and existing lower and upper bounds. The first row lists the parameters for the system size and the second row lists the component

Table 2.15: Comparison of the derived and existing lower and upper bounds for a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system.

(m,n)	$(30,30)$				$(40,40)$			
(α,β)	$(0.950,0.990)$		$(0.995,0.999)$		$(0.950,0.990)$		$(0.995,0.999)$	
	time	value	time	value	time	value	time	value
$LB_D(b,l_1,l_2)$	0.67	0.448896	0.66	0.991412	45.51	0.238073	45.36	0.984654
LB_{EP}	0.00	0.418860	0.02	0.991338	0.00	0.210054	0.01	0.984521
$LB_{EP}^{(G)}(b)$	2.43	0.436528	2.45	0.991121	308.00	0.229368	307.30	0.984267
$UB_D(b,l_1,l_2)$	0.54	0.449056	0.47	0.991412	45.93	0.238224	46.01	0.984655
$UB_{FK}^{(G)}$	0.01	0.466385	0.01	0.991449	0.01	0.255203	0.01	0.984721

reliability, and the others list the computation times (sec.) and values for the upper and lower bounds. From the results, we found that the derived bounds needed more computation times compared with the existing ones except for $LB_{EP}^{(G)}(b)$. Furthermore, we obtained the bounds that are tighter than the best of the existing bounds within the scope of the experiment. From Table 2.15, we have tighter upper and lower bounds as the component reliability is high. In particular, when $(\alpha,\beta)=(0.995,0.999)$, the upper bound $UB_D(b,l_1,l_2)$ and the lower bound $LB_D(b,l_1,l_2)$ were coincident to at least five after the decimal point. It is noteworthy that the values can be regarded as the exact system reliability up to five decimal places.

2.4 Summary

This chapter focused on the system reliability evaluation. Here, we summarize the individual contributions of each part.

(A) Computing the reliability of the Tor/(r,s)/(m,n):F system Section 2.1 considered the reliability of the Tor/(r,s)/(m,n):F system. First, we provided a recursive equation approach, and, for efficiently computing the system reliability, we proposed two kinds of algorithms. One of the algorithms was incorporated with the idea that the elements of $\Theta(i;\boldsymbol{h};\boldsymbol{g})$ are enumerated beforehand and stored with the purpose of increasing efficiency, which enables us to avoid redundant computations. It was theoretically shown that the algorithm with the above idea required extra memory space but has better time complexity compared to the other one. Computational results sug-

gest that this idea is particularly useful. Furthermore, we obtained the B-importance for the Tor/(r, s)/(m, n):F system by the proposed algorithm as a numerical example.

(B) Comparison of the recursive equation approach and the FMCIA Section 2.2 proposed the recursive equation approach and the FMCIA for computing the reliability of Lin, Cir, and Tor/$(1, 2)$-or-$(2, 1)$/(m, n):F systems. Note that the recursive equation approach for a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system has already been proposed by Yamamoto *et al.* [64]. One of the contributions of this chapter is to describe them in a unified manner. Because there were no studies reported on efficiency comparisons between both methods, we investigated the efficiency of both methods through numerical experiments. In conclusion, we see that their efficiency depends on the situation. Specifically, for a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system, in the INID case, the recursive equation approach was more efficient. In the IID case, if the number of columns was large, the FMCIA outperformed the recursive equation approach. On the other hand, for a Tor/$(1, 2)$-or-$(2, 1)$/(m, n):F system, within the scope of the experiment, we can conclude that the FMCIA outperforms the recursive equation approach. This result enables us to select which exact method should be used for efficiently computing the system reliability.

(C) Bounds for the reliability of a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system Section 2.3 derived the upper and lower bounds for the reliability of a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system. Moreover, we conducted numerical experiments to evaluate the derived bounds. From the results, within the scope of the experiment, it was found that the derived bounds were tighter than the best of the existing bounds, although they required more computation time. Therefore, It was concluded that we obtained the tighter bounds at the expense of the computational effort compared with the existing bounds.

Together with the results of the numerical experiments in this chapter, we can select appropriate methods for computing the reliability of a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system according to the following guideline. Note that we can obtain the same result even if we swap m and n due to the system symmetry. If $\min\{m, n\} \leq 20$, we suggest using the exact methods. In the INID case, the recursive equation approach in Theorem 2.2 [64] should be used; in the IID case, if the number of columns is large, the

FMCIA in Theorem 2.3. should be used. If $20 \leq \min\{m, n\} \leq 40$, we suggest using the bounds in Theorem 2.10. It is worth noting that the upper and lower bounds are approximately coincident when the component reliabilities are high. If $40 \leq \min\{m, n\}$, we should use the existing simple bounds, e.g., LB_{EP} [66] and $UB_{FK}^{(G)}$ [68].

Chapter 3

Computing the System Signature

The aim of this chapter is to propose methods for computing the system signatures of a Lin/(r,s)/(m,n):F system and Lin/$(1,2)$-or-$(2,1)$/(m,n):F system. The system signature has various theoretical applications; for example, it is used to establish stochastic comparisons among some systems. Section 3.1 presents the existing methods for computing the system signature, which are compared with the proposed methods in order to investigate the efficiency. Sections 3.2 and 3.3 propose methods for computing the system signatures of a Lin/(r,s)/(m,n):F system and a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system, respectively. Additionally, numerical experiments are conducted to compare the efficiency between the methods proposed in Sections 3.2 and 3.3 and existing methods. As applications of the system signature, Section 3.4 compares the connected-**X**-out-of-(m,n):F lattice systems based on the stochastic order. Finally, Section 3.5 summarizes the contributions of the chapter.

3.1 Existing Methods for Computing the System Signature

Let us consider a coherent system consisting of N components whose lifetimes are independent and identically distributed (IID). Recall that a coherent system is defined in Section 1.3. Let $T_1, T_2, \ldots, T_N$ be the lifetimes of the N components, and let T be the lifetime of the system. Recall that Samaniego [79] defined the system signature as

a probability vector $\boldsymbol{s}(N) = (s_1(N), s_2(N), \ldots, s_N(N))$, and its element is defined as

$$s_i(N) = \Pr\{T = T_{i:N}\}, \tag{3.1}$$

where $T_{i:N}$ is the ith order statistic (that is, the ith smallest value) among $T_1, T_2, \ldots, T_N$ for $i = 1, 2, \ldots, N$. However, the computation of a system signature is known to be challenging, especially when a system has a large number of components. This section presents several existing methods for computing a system signature.

The most common method is to directly compute the system signature based on the definition in Kochar *et al.* [108]. In terms of the orderings of the lifetimes of components, the system signature can be computed as

$$s_i(N) = \frac{\text{Number of orderings for which the } i\text{th failure causes system failure}}{N!}. \tag{3.2}$$

From the assumption that T_i, $i = 1, 2, \ldots, N$, are IID, the $N!$ permutations of these N distinct failure times are equally likely. Although it is simple and straightforward, this method becomes time-consuming and possibly inapplicable to large size systems.

One approach is a modular decomposition, which can reduce the computational complexity if the system signatures of all modules are given. Da *et al.* [109,110] derived two basic formulas for computing the system signature of a system that can be decomposed into two subsystems (modules). Marichal *et al.* [111] considered the general case where a system can be partitioned into several disjoint modules and provided a general formula of the system signature in terms of the system signatures of the modules. Recently, Jia *et al.* [112] derived the explicit formulas for computing the system signature of a modular system. However, the modular decomposition cannot be used to compute the system signature of the Lin/(r, s)/(m, n):F system and Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system because these systems cannot be decomposed into disjoint modules.

Other approaches compute a system signature via additional information, e.g., dominations[1], minimal cut set[2], and minimal path set[3]. For example, Boland *et al.* [113] provided the linkage between the dominations and the system signatures, which enables us to compute the system signature from the dominations. Also, Da *et al.* [114]

[1]Let us suppose that a system with N components, and all components have a common reliability p. The reliability of the system can be expressed as $\sum_{i=1}^{N} d_i p^i$, where coefficients d_i are called *dominations*.

[2]A minimal set of components whose failure ensures the failure of the system.

[3]A minimal set of components whose functioning ensures the functioning of the system.

proposed an algorithm for computing the system signature under the assumption that minimal cut/path sets are given in advance or easy to obtain. Let us consider a coherent system with N components and k minimal cut sets: $C_1, C_2, \ldots, C_k$. Letting D be a subset of $\{1, 2, \ldots, k\}$ including the empty set, then we define

$$n_D = \left| \bigcup_{i \in D} C_i \right|,$$

where $|\Omega|$ refers to the number of elements of the set Ω. Using the above notation, Da *et al.* [114] presented the following explicit formulas for computing the system signature via minimal cut sets:

$$s_i(N) = \sum_{D \subseteq \{1,2,\ldots,k\}} (-1)^{|D|} \frac{\binom{i-1}{n_D}}{\binom{n}{n_D}} - \sum_{D \subseteq \{1,2,\ldots,k\}} (-1)^{|D|} \frac{\binom{i}{n_D}}{\binom{n}{n_D}}. \tag{3.3}$$

Boland [115] provided an equation for computing a system signature from a well-known quantity. For $i = 1, 2, \ldots, N$, the ith element of a system signature can be computed as follows:

$$s_i(N) = \frac{r_{i-1}(N)}{\binom{N}{i-1}} - \frac{r_i(N)}{\binom{N}{i}}, \tag{3.4}$$

where $r_i(N)$ is the number of path sets of a system with exactly i failed components for $i = 0, 1, \ldots, N$. Equation (3.4) implies that if $r_i(N)$s are obtained, we can compute the system signature easily. In other words, a classical combinational approach enables us to compute a system signature. Typically, this approach tends to be applied to compute the system signatures of systems with regular structures, such as consecutive-k systems. For example, Eryilmaz [116] and Eryilmaz and Tuncel [117] gave $r_i(N)$s of consecutive-k-out-of-n:F systems in order to compute the system signature. Triantafyllou [14] provided a comprehensive review of computing the system signatures of various consecutive-k systems.

A few methods for obtaining the number of path sets $r_i(N)$ of a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system were reported. El-Sayed [118] derived an equation in the $m = 2$ case, which was the first-ever attempt to obtain $r_i(N)$s of a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system. Subsequently, Ishikawa *et al.* [119] derived two equations in the cases of $m = 2$ and 3. When $m = 2$, an equation provided in Ishikawa *et al.* [119] has a smaller

number of terms than in that of El-Sayed [118]. Thus, we present the equation derived by Ishikawa *et al.* [119]. Letting, for convenience of notation, $r_z(m, j)$ be the number of path sets of a Lin/$(1, 2)$-or-$(2, 1)/(m, j)$:F system with exactly z failed components for $z = 0, 1, \ldots, 2n$ and $j = 1, 2, \ldots, n$. The number of path sets $r_z(2, n)$ is given by

$$r_z(2, j) = r_z(2, j-1) + r_{z-1}(2, j-1) + r_{z-1}(2, j-2), \tag{3.5}$$

for $j = 3, 4, \ldots, n$ and $z = 2, 3, \ldots, 2n$. As the boundary condition, for $j \leq 2$ or $z \leq 1$,

$$r_z(2, j) = \begin{cases} 1, & \text{if } z = 0 \text{ and } j \geq 1, \\ 2j, & \text{if } z = 1 \text{ and } j \geq 1, \\ 2, & \text{if } z = 2 \text{ and } j = 2, \\ 0, & \text{otherwise.} \end{cases} \tag{3.6}$$

Furthermore, Ishikawa *et al.* [119] gave the equation for obtaining $r_z(3, n)$s in the $m = 3$ case. The number of path sets $r_z(3, n)$ is given by

$$\begin{aligned} r_z(3, j) =& r_z(3, j-1) + r_{z-1}(3, j-1) + 2r_{z-1}(3, j-2) + 3r_{z-2}(3, j-2) \\ &+ r_{z-3}(3, j-2) + r_{z-2}(3, j-3) - r_{z-4}(3, j-3) - r_{z-4}(3, j-4), \end{aligned} \tag{3.7}$$

for $j = 5, 6, \ldots, n$ and $z = 4, 5, \ldots, 3n$. As the boundary condition, for $j \leq 4$ or $z \leq 3$,

$$r_z(3, j) = \begin{cases} 1, & \text{if } z = 0 \text{ and } j \geq 1, \\ 3j, & \text{if } z = 1 \text{ and } j \geq 1, \\ \frac{1}{2}(9j^2 - 13j + 6), & \text{if } z = 2 \text{ and } j \geq 1, \\ \frac{1}{2}(9j^3 - 39j^2 + 64j - 40), & \text{if } z = 3 \text{ and } j \geq 1, \\ 6, & \text{if } z = 4 \text{ and } j = 3, \\ 61, & \text{if } z = 4 \text{ and } j = 4, \\ 1, & \text{if } z = 5 \text{ and } j = 3, \\ 18, & \text{if } z = 5 \text{ and } j = 4, \\ 2, & \text{if } z = 6 \text{ and } j = 4, \\ 0, & \text{otherwise.} \end{cases} \tag{3.8}$$

Note that the number of rows and the number of columns are symmetric, and thus, we can also obtain the number of path sets when $n = 2$ and 3 by Eqs. (3.5) and (3.7), respectively.

Although the equations for the cases of $m = 2$ and 3 were reported, the explicit representations are not available for any m and n. Generally, it is possible to derive the equation for obtaining the number of path sets when m is given. However, the approach proposed by Ishikawa *et al.* [119] is quite cumbersome because the number of terms increases as the parameter m increases. Furthermore, an equation for obtaining $r_i(N)$s of the Lin/(r, s)/(m, n):F system has not been reported so far. Therefore, we propose the efficient methods for computing the system signatures of a Lin/(r, s)/(m, n):F system and Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system.

3.2 System Signature of a Lin/(r, s)/(m, n):F System

In this section, we compute the system signature of a Lin/(r, s)/(m, n):F system. When using Eq. (3.4) to obtain the system signature, we need the number of path sets of the Lin/(r, s)/(m, n):F system. Accordingly, we propose an efficient method, depending on a structure of the Lin/(r, s)/(m, n):F system.

3.2.1 Notation

In this subsection, we define some notations. To indicate whether or not component (i, j) works, for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, we define the indicator variable x_{ij} by

$$x_{ij} = \begin{cases} 0, & \text{if component } (i, j) \text{ works,} \\ 1, & \text{if component } (i, j) \text{ fails.} \end{cases} \tag{3.9}$$

The states of the m components in the jth column are represented by the state vector $\boldsymbol{x}_j = (x_{1j}, x_{2j}, \ldots, x_{mj})$ for $j = 1, 2, \ldots, n$. For $j = 1, 2, \ldots, n$, the number of the failed components in the jth column, denoted by $N(\boldsymbol{x}_j)$, is given by

$$N(\boldsymbol{x}_j) = \sum_{a=1}^{m} x_{aj}. \tag{3.10}$$

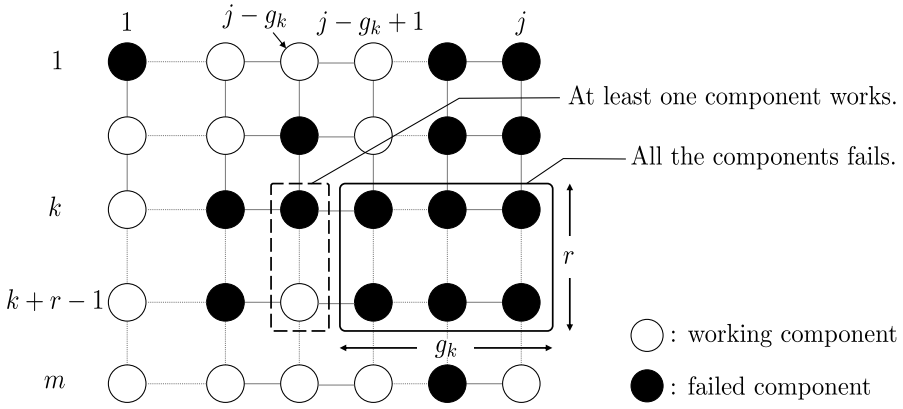


Figure 3.1: Definition of $A_j(\boldsymbol{g})$.

The states of the mn components in a Lin/(r,s)/(m,n):F system can be expressed as a matrix $(\boldsymbol{x}_1^\mathsf{T}, \boldsymbol{x}_2^\mathsf{T}, \ldots, \boldsymbol{x}_n^\mathsf{T})$. A structure function of a Lin/(r,s)/(m,n):F system, denoted by $\phi^{RS}(\boldsymbol{x}_1^\mathsf{T}, \boldsymbol{x}_2^\mathsf{T}, \ldots, \boldsymbol{x}_n^\mathsf{T})$, is given by

$$\phi^{RS}(\boldsymbol{x}_1^\mathsf{T}, \boldsymbol{x}_2^\mathsf{T}, \ldots, \boldsymbol{x}_n^\mathsf{T}) = 1 - \prod_{k=1}^{m-r+1} \prod_{l=1}^{n-s+1} \left(1 - \prod_{u=k}^{k+r-1} \prod_{v=l}^{l+s-1} x_{uv}\right), \tag{3.11}$$

where $\boldsymbol{a}^\mathsf{T}$ is the transpose of row vector $\boldsymbol{a}$, and $\phi^{RS}(\boldsymbol{x}_1^\mathsf{T}, \boldsymbol{x}_2^\mathsf{T}, \ldots, \boldsymbol{x}_n^\mathsf{T})$ takes 0 if the system works and 1 otherwise. Letting $\alpha_z((r,s),(m,n))$ denote the number of path sets of a Lin/(r,s)/(m,n):F system with exactly z failed components, then we have

$$\alpha_z((r,s),(m,n)) = \left| \left\{ \ (\boldsymbol{x}_1^\mathsf{T}, \ldots, \boldsymbol{x}_n^\mathsf{T}) \ \middle| \ \phi^{RS}(\boldsymbol{x}_1^\mathsf{T}, \ldots, \boldsymbol{x}_n^\mathsf{T}) = 0 \text{ and } \sum_{a=1}^{n} N(\boldsymbol{x}_a) = z \ \right\} \right|. \tag{3.12}$$

3.2.2 Proposal of a Computational Method

In this subsection, we derive a recursive equation for obtaining $\alpha_z((r,s),(m,n))$ by utilizing the event decomposition approach [22] in order to compute the system signature. Let $\boldsymbol{g}$ be an $(m-r+1)$-dimensional vector $(g_1, g_2, \ldots, g_{m-r+1})$, where $g_k \in \{0, 1, \ldots, s-1\}$ for $k = 1, 2, \ldots, m-r+1$. The elements of $\boldsymbol{g}$ represents the length of a rectangle in which all the components fail. For $j = 1, 2, \ldots, n$, we define the

following indicator variable:

$$A_j(\boldsymbol{g}) = \prod_{k=1}^{m-r+1} \left(\left(\prod_{u=k}^{k+r-1} \prod_{v=j-g_k+1}^{j} x_{uv} \right) \times \left(1 - \prod_{u=k}^{k+r-1} x_{u,j-g_k} \right) \right), \tag{3.13}$$

where $\prod_{i=a}^{b} x_i = 1$ for $a > b$. Here, $A_j(\boldsymbol{g})$ takes 1 if "all the components fail in an $r \times g_k$ rectangle with four corners $(k, j-g_k+1)$, (k, j), $(k+r-1, j)$, and $(k+r-1, j-g_k+1)$" and "at least one component works in components $(k, j-g_k)$, $(k+1, j-g_k), \ldots, (k+r-1, j-g_k)$," and 0 otherwise for all $k = 1, 2, \ldots, m-r+1$. Figure 3.1 illustrates $A_j(\boldsymbol{g})$. The element of $\boldsymbol{g}$ represents the width of a rectangle in which all components fail, as shown in Fig. 3.1. Let $\alpha(j; z; \boldsymbol{g})$ be the number of path sets of a Lin/(r, s)/(m, j):F system with exactly z failed components and the additional condition given by $\boldsymbol{g}$, and it is defined as

$$\alpha(j; z; \boldsymbol{g}) = \left| \left\{ \ (\boldsymbol{x}_1^{\mathsf{T}}, \ldots, \boldsymbol{x}_j^{\mathsf{T}}) \ \middle| \ \phi^{RS}(\boldsymbol{x}_1^{\mathsf{T}}, \ldots, \boldsymbol{x}_j^{\mathsf{T}}) = 0, \sum_{a=1}^{j} N(\boldsymbol{x}_a) = z, \text{ and } A_j(\boldsymbol{g}) = 1 \ \right\} \right|. \tag{3.14}$$

In other words, $\alpha(j; z; \boldsymbol{g})$ is the number of path sets of a Lin/(r, s)/(m, j):F system that satisfies the following two conditions:

(a) The system has exactly z failed components.

(b) The system fulfills the additional condition, namely, $A_j(\boldsymbol{g}) = 1$.

Figure 3.2 illustrates a Lin/$(2, 2)$/$(3, j)$:F system with exactly z failed components and the additional condition given by $\boldsymbol{g} = (1, 0)$.

From Eqs. (3.12) and (3.14), for $z = 0, 1, \ldots, mn$, we obtain

$$\alpha_z((r, s), (m, n)) = \sum_{\boldsymbol{g} \in S} \alpha(n; z; \boldsymbol{g}), \tag{3.15}$$

where

$$S = \left\{ \ \boldsymbol{g} \ \middle| \ \prod_{k=1}^{m-r+1} \mathcal{X}(g_k < s) = 1 \ \right\}, \tag{3.16}$$

and $\mathcal{X}(G)$ is the indicator function which takes 1 if argument G is true and 0 otherwise.

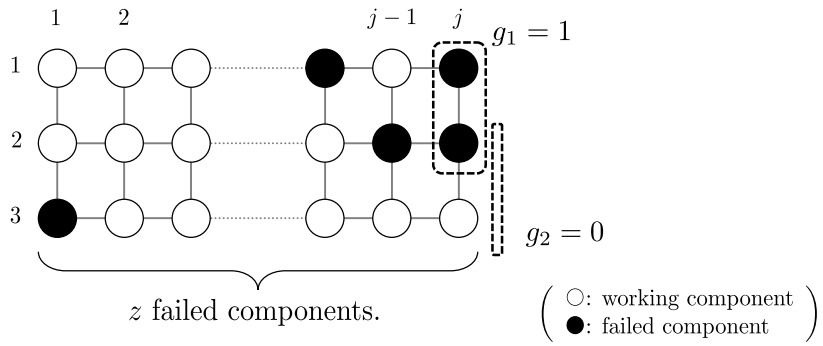


Figure 3.2: Lin/$(2,2)$/$(3,j)$:F system with exactly z failed components and the additional condition given by $\boldsymbol{g} = (1,0)$.

Equation (3.15) shows that the number of path sets of a Lin/(r,s)/(m,n):F system with exactly z failed components can be obtained by summing up $\alpha(n;z;\boldsymbol{g})$s for all $\boldsymbol{g} \in S$.

However, some $\boldsymbol{g}$s are impossible to exist. For example, suppose $r = 2$, $m = 4$, and $\boldsymbol{g} = (g_1, g_2, g_3) = (1,0,1)$. $g_1 = 1$ and $g_3 = 1$ state that the components $(1,j)$, $(2,j)$, $(3,j)$, and $(4,j)$ have failed, whereas $g_2 = 0$ states that the component $(2,j)$ or $(3,j)$ is working, which is obviously impossible. For convenience, if $\boldsymbol{g}$ is impossible to exist, we define $\alpha(j;z;\boldsymbol{g}) = 0$ for any $j = 1,2,\ldots,n$ and $z = 0,1,\ldots,mn$. Yamamoto and Akiba [62] proved that $\boldsymbol{g}$s are impossible for $\boldsymbol{g} \in E$, where

$$E = \left\{ \boldsymbol{g} \in S \;\middle|\; \begin{array}{l} g_i > g_{i+1} \text{ and } g_{i+u} < g_{i+u+1} \text{ for some } i \in \{1,2,\ldots,m-r-1\} \\ \text{and some } u \in \{1,2,\ldots,r-1\} \end{array} \right\}. \tag{3.17}$$

Hence, for $\boldsymbol{g} \in E$, it follows that

$$\alpha(j;z;\boldsymbol{g}) = 0,$$

for any $j = 1,2,\ldots,n$ and $z = 0,1,\ldots,mn$. Accordingly, by eliminating $\boldsymbol{g} \in E$ from the set S, we can obtain $\alpha_z((r,s),(m,n))$ efficiently from Eq. (3.15).

Using the above notation, we can compute the system signature of a Lin/(r,s)/(m,n):F system by utilizing the following theorem.

Theorem 3.1. (a) For $j = 1, 2, \ldots, n$, $z = 0, 1, \ldots, mn$, and $\boldsymbol{g} \in S$,

$$\alpha(j; z; \boldsymbol{g}) = \begin{cases} \binom{mj}{z}, & \text{if } 0 \le z \le r-1 \text{ and } \boldsymbol{g} = (0, \ldots, 0), \\ \displaystyle\sum_{\boldsymbol{y} \in U(\boldsymbol{g})} \sum_{\boldsymbol{g}' \in \Theta(\boldsymbol{g})} \alpha(j-1; z - N(\boldsymbol{y}); \boldsymbol{g}'), & \text{if } r \le z \le mj \text{ and } \boldsymbol{g} \notin E, \\ 0, & \text{otherwise,} \end{cases} \tag{3.18}$$

where $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ ($y_i \in \{0, 1\}$ for $i = 1, 2, \ldots, m$), $\boldsymbol{g}' = (g'_1, g'_2, \ldots, g'_{m-r+1})$ ($g'_k \in \{0, 1, \ldots, s-1\}$ for $k = 1, 2, \ldots, m-r+1$),

$$U(\boldsymbol{g}) = \left\{ \boldsymbol{y} \;\middle|\; \begin{array}{l} \displaystyle\prod_{u=k}^{k+r-1} y_u = \mathcal{X}(g_k \ge 1), \text{ for } k = 1, 2, \ldots, m-r+1, \\ \text{and } y_i \in \{0, 1\} \text{ for } i = 1, 2, \ldots, m \end{array} \right\}, \tag{3.19}$$

$$\Theta(\boldsymbol{g}) = \left\{ \boldsymbol{g}' \;\middle|\; \begin{array}{ll} g'_k \in \{g_k - 1\}, & \text{if } g_k \ge 1, \\ g'_k \in \{0, 1, \ldots, s-1\}, & \text{if } g_k = 0, \end{array} \right\}, \tag{3.20}$$

and S and E are given by Eqs. (3.16) and (3.17), respectively. The boundary condition is given, for $z = 0, 1, \ldots, mn$ and $\boldsymbol{g} \in S$, by

$$\alpha(0; z; \boldsymbol{g}) = \begin{cases} 1, & \text{if } z = 0 \text{ and } \boldsymbol{g} = (0, \ldots, 0), \\ 0, & \text{otherwise.} \end{cases} \tag{3.21}$$

(b) For $z = 0, 1, \ldots, mn$,

$$\alpha_z((r, s), (m, n)) = \sum_{\boldsymbol{g} \in S \setminus E} \alpha(n; z; \boldsymbol{g}). \tag{3.22}$$

(c) For $i = 1, 2, \ldots, mn$,

$$s_i(mn) = \frac{\alpha_{i-1}((r, s), (m, n))}{\binom{mn}{i-1}} - \frac{\alpha_i((r, s), (m, n))}{\binom{mn}{i}}. \tag{3.23}$$

The proof is in Appendix.

3.2.3 Algorithm and Illustrative Example

In this subsection, we discuss an algorithm based on Theorem 3.1 for efficiently computing the system signature of the Lin/(r,s)/(m,n):F system. Basically, the algorithm involves the following steps:

Step 1: Enumerate all elements in S and E by Eqs. (3.16) and (3.17), respectively. Set $j=0$ and $\alpha(0;z;\boldsymbol{g})$ by Eq. (3.21) for $z=0,1,\ldots,mn$ and $\boldsymbol{g}\in S$.

Step 2: Set $j=j+1$, and compute $\alpha(j;z;\boldsymbol{g})$ for $z=0,1,\ldots,mn$ and $\boldsymbol{g}\in S$ from Eq. (3.18). Repeat this step until $j=n$.

Step 3: Compute $\alpha_z((r,s),(m,n))$ for $z=0,1,2,\ldots,mn$ from Eq. (3.22).

Step 4: Obtain $s_i(mn)$ for $i=1,2,\ldots,mn$ by Eq. (3.23).

Next, we demonstrate how this algorithm obtains the system signature of a Lin/(r,s)/(m,n):F system. For instance, let us consider a Lin/$(2,2)$/$(3,3)$:F system.
Step 1: We enumerate all the elements in S and E as follows:

$$\begin{aligned} S &= \{(0,0),(0,1),(1,0),(1,1)\}, \\ E &= \emptyset, \end{aligned}$$

and, for $z=0,1,2,\ldots,9$, we have

$$\alpha(0;z;\boldsymbol{g}) = \begin{cases} 1, & \text{if } z=0 \text{ and } \boldsymbol{g}=(0,0), \\ 0, & \text{otherwise.} \end{cases} \tag{3.24}$$

Step 2: First, we consider the $j=1$ case. If $0\le z\le 1$ and $\boldsymbol{g}=(0,0)$, then we can easily compute $\alpha(1;z;\boldsymbol{g})$ as follows:

$$\begin{aligned} \alpha(1;0;(0,0)) &= \binom{3}{0} = 1, \\ \alpha(1;1;(0,0)) &= \binom{3}{1} = 3. \end{aligned}$$

Table 3.1: $\alpha(1; z; \boldsymbol{g})$ of the Lin/(2, 2)/(3, 3):F system.

z	0	1	2	3	4	5	6	7	8	9
$\boldsymbol{g} = (0, 0)$	1	3	1	0	0	0	0	0	0	0
$\boldsymbol{g} = (0, 1)$	0	0	1	0	0	0	0	0	0	0
$\boldsymbol{g} = (1, 0)$	0	0	1	0	0	0	0	0	0	0
$\boldsymbol{g} = (1, 1)$	0	0	0	1	0	0	0	0	0	0

Otherwise, $\alpha(1; z; \boldsymbol{g})$ is computed recursively; for example, if $z = 2$ and $\boldsymbol{g} = (0, 1)$,

$$\alpha(1; 2; (0, 1)) = \sum_{\boldsymbol{y} \in U((0,1))} \sum_{\boldsymbol{g}' \in \Theta((0,1))} \alpha(0; 2 - N(\boldsymbol{y}); \boldsymbol{g}').$$

From Eqs. (3.19) and (3.20), since

$$U((0, 1)) = \{(0, 1, 1)\},$$
$$\Theta((0, 1)) = \{(0, 0), (1, 0)\},$$

we can get

$$\begin{aligned}
\alpha(1; 2; (0, 1)) &= \alpha(0; 2 - N((0, 1, 1)); (0, 0)) + \alpha(0; 2 - N((0, 1, 1)); (0, 1)), \\
&= \alpha(0; 0; (0, 0)) + \alpha(0; 0; (0, 1)), \\
&= 1 + 0 = 1. \qquad \text{(from Eq. (3.24))}
\end{aligned}$$

Also, if $z = 2$ and $\boldsymbol{g} = (1, 0)$, we have

$$\begin{aligned}
\alpha(1; 2; (1, 0)) &= \alpha(0; 0; (0, 0)) + \alpha(0; 0; (1, 0)), \\
&= 1 + 0 = 1. \qquad \text{(from Eq. (3.24))}
\end{aligned}$$

Table 3.1 lists all the $\alpha(1; z; \boldsymbol{g})$s, which can be computed following a similar way.

Next, we consider the $j = 2$ case. If $0 \leq z \leq 1$ and $\boldsymbol{g} = (0, 0)$, then we can easily

Table 3.2: $\alpha(2; z; \boldsymbol{g})$ of the Lin/(2, 2)/(3, 3):F system.

z	0	1	2	3	4	5	6	7	8	9
$\boldsymbol{g} = (0, 0)$	1	6	13	13	6	1	0	0	0	0
$\boldsymbol{g} = (0, 1)$	0	0	1	3	2	0	0	0	0	0
$\boldsymbol{g} = (1, 0)$	0	0	1	3	2	0	0	0	0	0
$\boldsymbol{g} = (1, 1)$	0	0	0	1	3	1	0	0	0	0

compute $\alpha(2; z; \boldsymbol{g})$ as follows:

$$\alpha(2; 0; (0, 0)) = \binom{6}{0} = 1,$$
$$\alpha(2; 1; (0, 0)) = \binom{6}{1} = 6.$$

Otherwise, $\alpha(2; z; \boldsymbol{g})$ is computed recursively; for example, if $z = 4$ and $\boldsymbol{g} = (1, 0)$, since

$$U((1, 0)) = \{(1, 1, 0)\},$$
$$\Theta((1, 0)) = \{(0, 0), (0, 1)\},$$

we have

$$\begin{aligned}
\alpha(2; 4; (1, 0)) &= \sum_{\boldsymbol{y} \in U((0,1))} \sum_{\boldsymbol{g}' \in \Theta((0,1))} \alpha(1; 4 - N(\boldsymbol{y}); \boldsymbol{g}'), \\
&= \alpha(1; 4 - N((0, 1, 1)); (0, 0)) + \alpha(1; 4 - N((0, 1, 1)); (0, 1)), \\
&= \alpha(1; 2; (0, 0)) + \alpha(1; 2; (0, 1)), \\
&= 1 + 1 = 2. \qquad \text{(from Table 3.1)}
\end{aligned}$$

Table 3.2 lists all the $\alpha(2; z; \boldsymbol{g})$s, which can be computed following a similar way.

Finally, we consider the $j = 3$ case. All the $\alpha(3; z; \boldsymbol{g})$s are listed in Table 3.3.

Step 3: We compute $\alpha_z((2, 2), (3, 3))$ for $z = 0, 1, 2, \ldots, 9$ from Eq. (3.22); for example,

Table 3.3: $\alpha(3; z; \boldsymbol{g})$ of the Lin/(2, 2)/(3, 3):F system.

z	0	1	2	3	4	5	6	7	8	9
$\boldsymbol{g} = (0, 0)$	1	9	34	71	88	61	19	2	0	0
$\boldsymbol{g} = (0, 1)$	0	0	1	6	14	16	8	1	0	0
$\boldsymbol{g} = (1, 0)$	0	0	1	6	14	16	8	1	0	0
$\boldsymbol{g} = (1, 1)$	0	0	0	1	6	13	13	6	1	0

when $z = 0$, we have

$$\begin{aligned}\alpha_0((2,2),(3,3)) &= \sum_{\boldsymbol{g}\in S\setminus E} \alpha(3;0;\boldsymbol{g}),\\ &= \alpha(3;0;(0,0)) + \alpha(3;0;(0,1)) + \alpha(3;0;(1,0)) + \alpha(3;0;(1,1)),\\ &= 1+0+0+0 = 1. \qquad \text{(from Table 3.3)}\end{aligned}$$

Also, when $z = 4$, we have

$$\begin{aligned}\alpha_4((2,2),(3,3)) &= \sum_{\boldsymbol{g}\in S\setminus E} \alpha(3;4;\boldsymbol{g}),\\ &= \alpha(3;4;(0,0)) + \alpha(3;4;(0,1)) + \alpha(3;4;(1,0)) + \alpha(3;4;(1,1)),\\ &= 88+14+14+6 = 122. \qquad \text{(from Table 3.3)}\end{aligned}$$

Similarly, we can compute all the $\alpha_z((2, 2), (3, 3))$s as follows:

$$\begin{aligned}&\alpha_0((2,2),(3,3)) = 1, && \alpha_1((2,2),(3,3)) = 9,\\ &\alpha_2((2,2),(3,3)) = 36, && \alpha_3((2,2),(3,3)) = 84,\\ &\alpha_4((2,2),(3,3)) = 122, && \alpha_5((2,2),(3,3)) = 106,\\ &\alpha_6((2,2),(3,3)) = 48, && \alpha_7((2,2),(3,3)) = 10,\\ &\alpha_8((2,2),(3,3)) = 1, && \alpha_9((2,2),(3,3)) = 0.\end{aligned}$$

Step 4: We obtain $s_i(9)$ for $i = 1, 2, \ldots, 9$ by Eq. (3.23); for example, when $i = 7$, we

have

$$\begin{aligned} s_7(9) &= \frac{\alpha_6((2,2),(3,3))}{\binom{9}{6}} - \frac{\alpha_7((2,2),(3,3))}{\binom{9}{7}}, \\ &= \frac{48}{84} - \frac{10}{36} = \frac{37}{126}. \end{aligned}$$

Eventually, we can obtain the system signature of the Lin/(2, 2)/(3, 3):F system as follows:

$$\boldsymbol{s}(9) = \left(0, 0, 0, \frac{2}{63}, \frac{8}{63}, \frac{17}{63}, \frac{37}{126}, \frac{1}{6}, \frac{1}{9}\right).$$

Some obtained system signatures are listed in Table 3.4.

Table 3.4: System signatures of the Lin/(r,s)/(m,n):F systems.

(r,s)	(m,n)	system signature
$(2,2)$	$(3,3)$	$\boldsymbol{s}(9) = \left(0,0,0,\frac{2}{63},\frac{8}{63},\frac{17}{63},\frac{37}{126},\frac{1}{6},\frac{1}{9}\right)$
$(2,2)$	$(3,4)$	$\boldsymbol{s}(12) = \left(0,0,0,\frac{2}{165},\frac{8}{165},\frac{5}{44},\frac{19}{99},\frac{479}{1980},\frac{217}{990},\frac{7}{55},\frac{1}{22},0\right)$
$(2,2)$	$(4,4)$	$\boldsymbol{s}(16) = \left(0,0,0,\frac{9}{1820},\frac{9}{455},\frac{48}{1001},\frac{893}{10010},\frac{107}{780},\frac{139}{780},\frac{389}{2002},\frac{172}{1001},\frac{103}{910},\frac{79}{1820},0,0,0\right)$
$(2,2)$	$(4,5)$	$\boldsymbol{s}(20) = \left(0,0,0,\frac{4}{1615},\frac{16}{1615},\frac{943}{38760},\frac{151}{3230},\frac{12827}{167960},\frac{27649}{251940},\frac{693}{4940},\frac{13347}{83980},\frac{7909}{50388},\frac{97}{741},\frac{1127}{12920},\frac{163}{3876},\frac{473}{38760},\frac{3}{1615},0,0,0\right)$
$(2,3)$	$(3,4)$	$\boldsymbol{s}(12) = \left(0,0,0,0,0,\frac{1}{231},\frac{2}{77},\frac{43}{495},\frac{199}{990},\frac{19}{66},\frac{5}{22},\frac{1}{6}\right)$
$(2,3)$	$(4,4)$	$\boldsymbol{s}(16) = \left(0,0,0,0,0,\frac{3}{4004},\frac{9}{2002},\frac{133}{8580},\frac{31}{780},\frac{1649}{20020},\frac{155}{1092},\frac{1103}{5460},\frac{207}{910},\frac{13}{70},\frac{1}{10},0\right)$
$(2,3)$	$(4,5)$	$\boldsymbol{s}(20) = \left(0,0,0,0,0,\frac{3}{12920},\frac{9}{6460},\frac{811}{167960},\frac{1057}{83980},\frac{12553}{461890},\frac{9391}{184756},\frac{5294}{62985},\frac{15563}{125970},\frac{6209}{38760},\frac{579}{3230},\frac{429}{2584},\frac{381}{3230},\frac{16}{285},\frac{3}{190},0\right)$
$(3,3)$	$(4,4)$	$\boldsymbol{s}(16) = \left(0,0,0,0,0,0,0,0,\frac{1}{2860},\frac{9}{2860},\frac{9}{572},\frac{101}{1820},\frac{267}{1820},\frac{11}{42},\frac{4}{15},\frac{1}{4}\right)$
$(3,3)$	$(4,5)$	$\boldsymbol{s}(20) = \left(0,0,0,0,0,0,0,0,\frac{3}{83980},\frac{27}{83980},\frac{27}{16796},\frac{1471}{251940},\frac{1429}{83980},\frac{161}{3876},\frac{1669}{19380},\frac{2909}{19380},\frac{683}{3230},\frac{127}{570},\frac{31}{190},\frac{1}{10}\right)$
$(3,4)$	$(4,5)$	$\boldsymbol{s}(20) = \left(0,0,0,0,0,0,0,0,0,0,0,\frac{2}{62985},\frac{8}{20995},\frac{4}{1615},\frac{443}{38760},\frac{1589}{38760},\frac{113}{969},\frac{68}{285},\frac{11}{38},\frac{3}{10}\right)$

3.2.4 Efficiency Investigation

In this subsection, we investigate the efficiency of the proposed method based on Theorem 3.1. The system signature of a Lin/(r,s)/(m,n):F system can be computed based on the proposed method, an enumeration method based on Eq. (3.4), and Da's [114] method, which is available because the minimal cut sets of Lin/(r,s)/(m,n):F systems can be easily obtained. Here, we explain an enumeration method based on Eq. (3.4). Let us consider a system with N components. This method enumerates 2^N possible system states that depend on the states of N components. These 2^N system states are identified for being a working or failed state of the system, and we count the number of the failed components in the working system. After checking all the system states, the number of path sets of the system with exactly i failed components ($i = 0, 1, \ldots, N$) can be obtained. From Eq. (3.4), eventually, we can get the system signature. Each method was programmed in MATLAB R2018a and implemented in a computer running on Microsoft Windows 10 with Intel Core i5 3.20 GHz and 8.0 GB memory. Computation time plays an essential role in the measure of the computational efficiency of different methods; thus, we compare the efficiency of the above-mentioned methods in terms of the required computation time.

First, we confirmed that these methods provided the same system signatures. Table 3.5 compares the computation time required for each case, where N/A indicates that the method failed to obtain a system signature in 12 hours. Table 3.5 indicates that the enumeration method requires longer computation time. For example, for the Lin/$(2,2)$/$(6,6)$:F system, the proposed method took only 0.341 seconds, whereas the enumeration method did not obtain the system signature within the time limit of 12 hours. The main reason for this is that the enumeration method needs to generate 2^{mn} states of mn components for counting the path sets of a certain size, making it time-consuming, especially for the systems with a large number of components. Da's method [114] performs well when the number of minimal cut sets ($(m-r+1)\times(n-s+1)$) is small, but its computation time increases exponentially with the number of minimal cut sets. More specifically, this method cannot obtain the system signatures within the time limit of 12 h when $(m-r+1)\times(n-s+1) \geq 25$.

On the other hand, the proposed method obtains the system signatures in a relatively shorter time. These results indicate that the proposed method is more efficient than the Da's method [114] in computing the system signature when the number of

Table 3.5: Comparison of computation times for system signatures of the Lin/(r, s)/(m, n):F systems (sec.).

(r, s)	(m, n)	Enumeration method	Da's method [114]	Proposed method
$(2, 2)$	$(4, 4)$	0.190	0.398	0.041
$(3, 3)$	$(4, 4)$	0.208	0.065	0.041
$(2, 2)$	$(6, 6)$	N/A	N/A	0.341
$(3, 3)$	$(6, 6)$	N/A	8.865	0.299
$(4, 4)$	$(6, 6)$	N/A	0.160	0.253
$(5, 5)$	$(6, 6)$	N/A	0.064	0.130
$(2, 2)$	$(8, 8)$	N/A	N/A	4.131
$(3, 3)$	$(8, 8)$	N/A	N/A	11.421
$(4, 4)$	$(8, 8)$	N/A	N/A	15.178
$(5, 5)$	$(8, 8)$	N/A	71.982	12.531
$(6, 6)$	$(8, 8)$	N/A	1.318	5.121
$(7, 7)$	$(8, 8)$	N/A	0.067	1.508
$(2, 2)$	$(10, 10)$	N/A	N/A	98.246
$(3, 3)$	$(10, 10)$	N/A	N/A	503.051
$(4, 4)$	$(10, 10)$	N/A	N/A	754.212
$(5, 5)$	$(10, 10)$	N/A	N/A	950.535
$(6, 6)$	$(10, 10)$	N/A	N/A	729.464
$(7, 7)$	$(10, 10)$	N/A	421.830	332.592
$(8, 8)$	$(10, 10)$	N/A	5.380	87.540
$(9, 9)$	$(10, 10)$	N/A	0.161	17.423

minimal cut sets is not very small.

In summary, these results imply that the proposed method outperforms the other methods in terms of obtaining the system signature of a Lin/(r, s)/(m, n):F system. The proposed method is more computationally efficient than Da's method [114] for those systems with not less than nine minimal cut sets within the scope of the experiment. Therefore, we can conclude that the proposed algorithm would be definitely applicable to a Lin/(r, s)/(m, n):F system with up to around 100 components.

3.3 System Signature of a Lin/$(1,2)$-or-$(2,1)/(m,n)$:F System

In this section, we compute the system signature of a Lin/$(1,2)$-or-$(2,1)/(m,n)$:F system.

3.3.1 Proposal of a Computational Method

First, we define some notations. A structure function of a Lin/$(1,2)$-or-$(2,1)/(m,n)$:F system, denoted by $\phi^{OR}(\boldsymbol{x}_1^{\mathsf{T}}, \boldsymbol{x}_2^{\mathsf{T}}, \ldots, \boldsymbol{x}_n^{\mathsf{T}})$, is given by

$$\phi^{OR}(\boldsymbol{x}_1^{\mathsf{T}}, \boldsymbol{x}_2^{\mathsf{T}}, \ldots, \boldsymbol{x}_n^{\mathsf{T}}) = 1 - \prod_{i=1}^{m}\prod_{j=2}^{n}(1 - x_{i,j-1}x_{ij}) \times \prod_{i=2}^{m}\prod_{j=1}^{n}(1 - x_{i-1,j}x_{ij}), \tag{3.25}$$

where $\phi^{OR}(\boldsymbol{x}_1^{\mathsf{T}}, \boldsymbol{x}_2^{\mathsf{T}}, \ldots, \boldsymbol{x}_n^{\mathsf{T}})$ takes 0 if the system works and 1 otherwise. Letting $\beta_z(m,n)$ denote the number of path sets of a Lin/$(1,2)$-or-$(2,1)/(m,n)$:F system with exactly z failed components, then we have

$$\beta_z(m,n) = \left| \left\{ (\boldsymbol{x}_1^{\mathsf{T}}, \ldots, \boldsymbol{x}_n^{\mathsf{T}}) \;\middle|\; \phi^{OR}(\boldsymbol{x}_1^{\mathsf{T}}, \ldots, \boldsymbol{x}_n^{\mathsf{T}}) = 0 \text{ and } \sum_{a=1}^{n} N(\boldsymbol{x}_a) = z \right\} \right|. \tag{3.26}$$

We define $\beta(j; z; \boldsymbol{y})$ as

$$\beta(j; z; \boldsymbol{y}) = \left| \left\{ (\boldsymbol{x}_1^{\mathsf{T}}, \ldots, \boldsymbol{x}_j^{\mathsf{T}}) \;\middle|\; \begin{array}{l} \phi^{OR}(\boldsymbol{x}_1^{\mathsf{T}}, \ldots, \boldsymbol{x}_j^{\mathsf{T}}) = 0, \sum_{a=1}^{j} N(\boldsymbol{x}_a) = z, \\ \text{and } x_{ij} = y_i, (i = 1, 2, \ldots, m) \end{array} \right\} \right|. \tag{3.27}$$

In other words, $\beta(j; z; \boldsymbol{y})$ is the number of path sets of a Lin/$(1,2)$-or-$(2,1)/(m,j)$:F system that satisfies the following two conditions:

(a) The system has exactly z failed components.

(b) The states of the m components in the jth column are given by a vector $\boldsymbol{y}$.

Figure 3.3 illustrates a Lin/$(1,2)$-or-$(2,1)/(3,j)$:F system with exactly z failed components and states of the m components in the jth column are given by $\boldsymbol{y} = (1,0,1)$.

Using the above notation, we can compute the system signature of a Lin/$(1,2)$-or-$(2,1)/(m,n)$:F system by utilizing the following theorem:

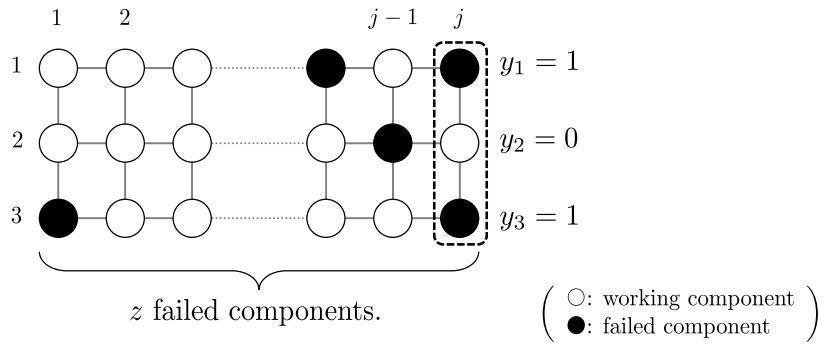


Figure 3.3: Lin/$(1,2)$-or-$(2,1)/(3,j)$:F system with exactly z failed components and states of the m components in the jth column are given by $\boldsymbol{y} = (1,0,1)$.

Theorem 3.2. (a) For $j = 1, 2, \ldots, n$, $z = 0, 1, \ldots, mn$, and $\boldsymbol{y} \in \{0,1\}^m$,

$$\beta(j; z; \boldsymbol{y}) = \begin{cases} \displaystyle\sum_{\boldsymbol{x} \in \Omega(\boldsymbol{y})} \beta(j-1; z - N(\boldsymbol{y}); \boldsymbol{x}), & \text{if } N(\boldsymbol{y}) \le z \le N(\boldsymbol{y}) + \left\lceil \frac{m(j-1)}{2} \right\rceil \\ & \text{and } \boldsymbol{y} \in W, \\ 0, & \text{otherwise,} \end{cases} \tag{3.28}$$

where $\lceil a \rceil$ denotes the smallest integer greater than or equal to a, and $\boldsymbol{x} = (x_1, x_2, \ldots, x_m), \boldsymbol{y} = (y_1, y_2, \ldots, y_m)$ $(x_i, y_i \in \{0,1\}$ for $i = 1, 2, \ldots, m)$,

$$W = \left\{ \boldsymbol{y} \;\middle|\; \sum_{i=2}^{m} y_{i-1} y_i = 0 \right\}, \tag{3.29}$$

and

$$\Omega(\boldsymbol{y}) = \left\{ \boldsymbol{x} \in W \;\middle|\; \sum_{i=1}^{m} x_i y_i = 0 \right\}. \tag{3.30}$$

The boundary condition is given, for $z = 0, 1, \ldots, mn$ and $\boldsymbol{y} \in \{0,1\}^m$, by

$$\beta(0; z; \boldsymbol{y}) = \begin{cases} 1, & \text{if } z = 0 \text{ and } \boldsymbol{y} = (0, \ldots, 0), \\ 0, & \text{otherwise.} \end{cases} \tag{3.31}$$

(b) For $z = 0, 1, \ldots, mn$,

$$\beta_z(m,n) = \sum_{\boldsymbol{y} \in W} \beta(n; z; \boldsymbol{y}). \tag{3.32}$$

(c) For $i = 1, 2, \ldots, mn$,

$$s_i(mn) = \frac{\beta_{i-1}(m,n)}{\binom{mn}{i-1}} - \frac{\beta_i(m,n)}{\binom{mn}{i}}. \tag{3.33}$$

We provide a proof of Theorem 3.2 in Appendix.

3.3.2 Algorithm and Illustrative Example

In this subsection, we provide the detailed steps of an algorithm based on Theorem 3.2 for efficiently computing the system signature of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system. Basically, the algorithm is executed in the following steps:

Step 1: Enumerate all the elements in W by Eq. (3.29), and set $j = 0$ and $\beta(0; z; \boldsymbol{y})$ for $z = 0, 1, \ldots, mn$ and $\boldsymbol{y} \in \{0,1\}^m$ from Eq. (3.31).

Step 2: Set $j = j + 1$, and compute $\beta(j; z; \boldsymbol{y})$ for $z = 0, 1, \ldots, mn$ and $\boldsymbol{y} \in \{0,1\}^m$ from Eq. (3.28). Repeat this step until $j = n$.

Step 3: Compute $\beta_z(m,n)$ for $z = 0, 1, 2, \ldots, mn$ from Eq. (3.32).

Step 4: Obtain $s_i(mn)$ for $i = 1, 2, \ldots, mn$ by Eq. (3.33).

Next, we demonstrate how this algorithm obtains the system signature of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system. For instance, let us consider a Lin/$(1,2)$-or-$(2,1)$/$(3,3)$:F system.

Step 1: We enumerate all the elements in W as follows:

$$W = \{(0,0,0), (0,0,1), (0,1,0), (1,0,0), (1,0,1)\},$$

and, for $z = 0, 1, \ldots, 9$, we have

$$\beta(0; z; \boldsymbol{y}) = \begin{cases} 1, & \text{if } z = 0 \text{ and } \boldsymbol{y} = (0,0,0), \\ 0, & \text{otherwise.} \end{cases} \tag{3.34}$$

Table 3.6: $\beta(1; z; \boldsymbol{y})$ of the Lin/$(1,2)$-or-$(2,1)/(3,3)$:F system.

z	0	1	2	3	4	5	6	7	8	9
$\boldsymbol{y} = (0,0,0)$	1	0	0	0	0	0	0	0	0	0
$\boldsymbol{y} = (0,0,1)$	0	1	0	0	0	0	0	0	0	0
$\boldsymbol{y} = (0,1,0)$	0	1	0	0	0	0	0	0	0	0
$\boldsymbol{y} = (1,0,0)$	0	1	0	0	0	0	0	0	0	0
$\boldsymbol{y} = (1,0,1)$	0	0	1	0	0	0	0	0	0	0

Step 2: First, we consider the $j = 1$ case. We compute $\beta(1; z; \boldsymbol{y})$ recursively; for example, if $z = 1$ and $\boldsymbol{g} = (0,0,1)$,

$$\beta(1;1;(0,0,1)) = \sum_{\boldsymbol{x} \in \Omega((0,0,1))} \beta(0; 1 - N((0,0,1)); \boldsymbol{x}).$$

Since

$$\Omega((0,0,1)) = \{(0,0,0), (0,1,0), (1,0,0)\},$$

we can get

$$\begin{aligned} \beta(1;1;(0,0,1)) &= \beta(0;0;(0,0,0)) + \beta(0;0;(0,1,0)) + \beta(0;0;(1,0,0)), \\ &= 1 + 0 + 0 = 1. \qquad \text{(from Eq. (3.34))} \end{aligned}$$

Table 3.6 lists all the $\beta(1; z; \boldsymbol{y})$s, which can be computed following a similar way.

Next, we consider the $j = 2$ case. $\beta(2; z; \boldsymbol{y})$ is computed recursively; for example, if $z = 2$ and $\boldsymbol{y} = (0,1,0)$, since

$$\Omega((0,1,0)) = \{(0,0,0), (0,0,1), (1,0,0), (1,0,1)\},$$

Table 3.7: $\beta(2; z; \boldsymbol{y})$ of the Lin/$(1,2)$-or-$(2,1)$/$(3,3)$:F system.

z	0	1	2	3	4	5	6	7	8	9
$\boldsymbol{y} = (0,0,0)$	1	3	1	0	0	0	0	0	0	0
$\boldsymbol{y} = (0,0,1)$	0	1	2	0	0	0	0	0	0	0
$\boldsymbol{y} = (0,1,0)$	0	1	2	1	0	0	0	0	0	0
$\boldsymbol{y} = (1,0,0)$	0	1	2	0	0	0	0	0	0	0
$\boldsymbol{y} = (1,0,1)$	0	0	1	1	0	0	0	0	0	0

Table 3.8: $\beta(3; z; \boldsymbol{y})$ of the Lin/$(1,2)$-or-$(2,1)$/$(3,3)$:F system.

z	0	1	2	3	4	5	6	7	8	9
$\boldsymbol{y} = (0,0,0)$	1	6	8	2	0	0	0	0	0	0
$\boldsymbol{y} = (0,0,1)$	0	1	5	5	1	0	0	0	0	0
$\boldsymbol{y} = (0,1,0)$	0	1	5	6	1	0	0	0	0	0
$\boldsymbol{y} = (1,0,0)$	0	1	5	5	1	0	0	0	0	0
$\boldsymbol{y} = (1,0,1)$	0	0	1	4	3	1	0	0	0	0

we have

$$\begin{aligned}\beta(2;2;(0,1,0)) =& \sum_{\boldsymbol{x}\in\Omega((0,1,0))} \beta(1; 2 - N((0,1,0)); \boldsymbol{x}),\\ =&\beta(1;1;(0,0,0)) + \beta(1;1;(0,0,1)) + \beta(1;1;(1,0,0)) + \beta(1;1;(1,0,1)),\\ =&0+1+1+0=2. \qquad \text{(from Table 3.6)}\end{aligned}$$

Table 3.7 lists all the $\beta(2; z; \boldsymbol{y})$s, which can be computed following a similar way.

Finally, we consider the $j = 3$ case. All the $\beta(3; z; \boldsymbol{y})$s are listed in Table 3.8.

Step 3: We compute $\beta_z(3,3)$ for $z = 0, 1, 2, \ldots, 9$ from Eq. (3.32); for example, when $z = 2$, we have

$$\begin{aligned}\beta_2(3,3) =& \sum_{\boldsymbol{y}\in W} \beta(3;2;\boldsymbol{y}),\\ =&\beta(3;2;(0,0,0)) + \beta(3;2;(0,0,1)) + \beta(3;2;(0,1,0))\\ &+ \beta(3;2;(1,0,0)) + \beta(3;2;(1,0,1)),\\ =&8+5+5+5+1=24. \qquad \text{(from Table 3.8)}\end{aligned}$$

Similarly, we can compute all the $\beta_z(3,3)$s as follows:

$$\begin{aligned}
&\beta_0(3,3) = 1, && \beta_1(3,3) = 9,\\
&\beta_2(3,3) = 24, && \beta_3(3,3) = 22,\\
&\beta_4(3,3) = 6, && \beta_5(3,3) = 1,\\
&\beta_6(3,3) = 0, && \beta_7(3,3) = 0,\\
&\beta_8(3,3) = 0, && \beta_9(3,3) = 0.
\end{aligned}$$

Step 4: We obtain $s_i(9)$ for $i = 1, 2, \ldots, 9$ by Eq. (3.33); for example, when $i = 5$, we have

$$\begin{aligned}
s_5(9) &= \frac{\beta_4(3,3)}{\binom{9}{4}} - \frac{\beta_5(3,3)}{\binom{9}{5}},\\
&= \frac{6}{126} - \frac{1}{126} = \frac{5}{126}.
\end{aligned}$$

Eventually, we can obtain the system signature of the Lin/(1, 2)-or-(2, 1)/(3, 3):F system as follows:

$$\boldsymbol{s}(9) = \left(0, \frac{1}{3}, \frac{17}{42}, \frac{3}{14}, \frac{5}{126}, \frac{1}{126}, 0, 0, 0\right).$$

Some obtained system signatures are listed in Table 3.9.

Table 3.9: System signatures of the Lin/$(1,2)$-or-$(2,1)$/(m,n):F systems.

(m,n)	system signature
$(3,3)$	$\boldsymbol{s}(9) = \left(0, \frac{1}{3}, \frac{17}{42}, \frac{3}{14}, \frac{5}{126}, \frac{1}{126}, 0, 0, 0\right)$
$(3,4)$	$\boldsymbol{s}(12) = \left(0, \frac{17}{66}, \frac{119}{330}, \frac{128}{495}, \frac{199}{1980}, \frac{19}{924}, \frac{1}{462}, 0, 0, 0, 0, 0\right)$
$(3,5)$	$\boldsymbol{s}(15) = \left(0, \frac{22}{105}, \frac{62}{195}, \frac{123}{455}, \frac{722}{5005}, \frac{237}{5005}, \frac{46}{5005}, \frac{8}{6435}, \frac{1}{6435}, 0, 0, 0, 0, 0, 0\right)$
$(3,6)$	$\boldsymbol{s}(18) = \left(0, \frac{3}{17}, \frac{115}{408}, \frac{109}{408}, \frac{184}{1071}, \frac{1052}{13923}, \frac{353}{15912}, \frac{757}{175032}, \frac{131}{218790}, \frac{1}{24310}, 0, 0, 0, 0, 0, 0, 0, 0\right)$
$(3,7)$	$\boldsymbol{s}(21) = \left(0, \frac{16}{105}, \frac{503}{1995}, \frac{1541}{5985}, \frac{1004}{5355}, \frac{5405}{54264}, \frac{31099}{813960}, \frac{4229}{406980}, \frac{757}{377910}, \frac{83}{293930}, \frac{11}{352716}, \frac{1}{352716}, 0, 0, 0, 0, 0, 0, 0, 0, 0\right)$
$(3,8)$	$\boldsymbol{s}(24) = \left(0, \frac{37}{276}, \frac{691}{3036}, \frac{79}{322}, \frac{1383}{7084}, \frac{15927}{134596}, \frac{3673}{67298}, \frac{281}{14858}, \frac{795}{163438}, \frac{609}{653752}, \frac{5}{36708}, \frac{235}{16224936}, \frac{1}{1352078}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\right)$
$(4,4)$	$\boldsymbol{s}(16) = \left(0, \frac{1}{5}, \frac{43}{140}, \frac{123}{455}, \frac{167}{1092}, \frac{95}{1716}, \frac{25}{2002}, \frac{41}{25740}, \frac{1}{6435}, 0, 0, 0, 0, 0, 0, 0\right)$
$(4,5)$	$\boldsymbol{s}(20) = \left(0, \frac{31}{190}, \frac{151}{570}, \frac{423}{1615}, \frac{413}{2280}, \frac{91}{1020}, \frac{599}{19380}, \frac{3727}{503880}, \frac{4}{3315}, \frac{61}{461890}, \frac{1}{92378}, 0, 0, 0, 0, 0, 0, 0, 0, 0\right)$
$(4,6)$	$\boldsymbol{s}(24) = \left(0, \frac{19}{138}, \frac{16}{69}, \frac{2627}{10626}, \frac{2059}{10626}, \frac{352}{3059}, \frac{746}{14421}, \frac{8537}{490314}, \frac{193}{44574}, \frac{769}{980628}, \frac{1}{9614}, \frac{5}{477204}, \frac{1}{1352078}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\right)$
$(5,5)$	$\boldsymbol{s}(25) = \left(0, \frac{2}{15}, \frac{779}{3450}, \frac{3083}{12650}, \frac{4709}{24150}, \frac{3173}{26565}, \frac{47191}{841225}, \frac{43277}{2163150}, \frac{98467}{18386775}, \frac{797}{742900}, \frac{1013}{6128925}, \frac{47}{2228700}, \frac{13}{5200300}, \frac{1}{5200300}, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0\right)$

3.3.3 Efficiency Investigation

In this subsection, we investigate the efficiency of the proposed method based on Theorem 3.2. The system signature of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system can be computed by the proposed method, Eqs. (3.5) and (3.7), an enumeration method based on Eq. (3.4), and Da's method [114]. The enumeration method, Eqs. (3.5) and (3.7) were selected for comparison with the proposed method here. Since the result in Subsection 3.2.4 shows that Da's method [114] is inefficient in dealing with a system with a large number of minimal cut sets, we do not compare the proposed method with Da's method [114]. Each method was programmed in MATLAB R2018a and was implemented on a computer with Intel Core i5 3.20 GHz CPU, 8.0 GB memory, Microsoft Windows 10 OS. We compare the efficiency of the above methods in terms of the required computation time.

First of all, we confirmed that these methods provided the same system signatures. Table 3.10 shows the comparison of the computation time required for each case, where N/A means that the method failed to obtain a system signature in 12 hours. Note that Eqs. (3.5) and (3.7) can be applied only for the cases of $m = 2$ and 3 (or $n = 2$ and 3), and thus, the cases where we cannot obtain a system signature are shown as a symbol "—". When comparing the proposed method and Eqs. (3.5) and (3.7), we see in Table 3.10 that Eqs. (3.5) and (3.7) can obtain the system signature in a shorter time. However, the computation times required for the proposed algorithm are sufficiently short. Additionally, Eqs. (3.5) and (3.7) are available for the cases of $m = 2$ and 3, whereas the proposed method can obtain the system signature general Lin/$(1,2)$-or-$(2,1)$/(m,n):F systems at least in theory. Next, we compare the proposed method and the enumeration method. Table 3.10 indicates that when the system size is small, although the proposed method requires more computation time compared with the enumeration method, their computation times are sufficiently short. On the other hand, as the system size is larger, the enumeration method becomes time-consuming. For example, for the Lin/$(1,2)$-or-$(2,1)$/$(6,6)$:F system, the proposed method took only 0.035 seconds, whereas the enumeration method could not obtain the system signature within the time limit of 12 hours. Also, from Table 3.10, we confirmed that the proposed method could provide the system signature of the larger Lin/$(1,2)$-or-$(2,1)$/(m,n):F systems. Furthermore, from Table 3.10, we found that a computation time for a Lin/$(1,2)$-or-$(2,1)$/$(10,15)$:F system (1.453 seconds) was shorter than that of

Table 3.10: Comparison of computation times for system signatures of Lin/$(1,2)$-or-$(2,1)$/(m,n):F systems (sec.).

(m,n)	Proposed method	Enumeration method	Eq. (3.5)	Eq. (3.7)
(2,2)	0.019	0.013	0.003	—
(2,3)	0.023	0.013	0.006	0.002
(3,3)	0.023	0.013	—	0.005
(3,4)	0.023	0.016	—	0.006
(4,4)	0.025	0.164	—	—
(4,5)	0.028	1.573	—	—
(5,5)	0.031	48.279	—	—
(5,6)	0.031	2023.453	—	—
(6,6)	0.035	N/A	—	—
(10,5)	0.500	N/A	—	—
(10,10)	1.328	N/A	—	—
(10,15)	1.453	N/A	—	—
(10,20)	1.625	N/A	—	—
(15,5)	2.922	N/A	—	—
(15,10)	6.766	N/A	—	—
(15,15)	13.016	N/A	—	—
(15,20)	22.688	N/A	—	—

a Lin/$(1,2)$-or-$(2,1)$/$(15,10)$:F system (6.766 seconds). Note that we can get the same results even if we interchange parameters m and n because of system symmetry. Thus, in order to compute a system signature in a shorter time, the parameter m should be the smaller value of the row or column.

In summary, the proposed method is more computationally efficient than the enumeration method for those systems with not less than 16 components within the scope of the experiment. Also, Eqs. (3.5) and (3.7) can obtain the system signature in a shorter time, but the computation times required for the proposed algorithm are sufficiently short. Therefore, we can conclude that the proposed algorithm would be definitely applicable to a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system with up to around 300 components.

3.4 Stochastic Comparisons of the Connected-X-out-of-(m, n):F Lattice Systems

In this section, we compare the lifetimes of Lin/(r, s)/(m, n):F systems based on the system signatures. By comparing the system signatures, we can compare the lifetimes of the systems and determine the system that works properly for a longer time. The concept of stochastic order is a useful tool in comparing system lifetimes [120]. Many types of stochastic relationships are commonly used, such as the usual stochastic order, the hazard rate order, the likelihood ratio order, and the stochastic precedence [121]. In this section, we focus on the usual stochastic order and the stochastic precedence. Although the hazard rate order and the likelihood ratio order are omitted in this thesis, they can be used similarly to the usual stochastic order.

Let T_A and T_B be the lifetimes of two systems. T_A is said to be smaller than T_B in the usual stochastic order (denoted by $T_A \leq_{st} T_B$) if and only if the following inequality holds.

$$\Pr\{T_A > t\} \leq \Pr\{T_B > t\}, \tag{3.35}$$

for all $t \in (0, \infty)$. Kochar *et al.* [108] used a system signature to make a stochastic comparison between various systems.

Proposition 3.1 (Kochar *et al.* [108])**.** Let $\boldsymbol{s}_A(N) = (s_{A,1}(N), s_{A,2}(N), \ldots, s_{A,N}(N))$ and $\boldsymbol{s}_B(N) = (s_{B,1}(N), s_{B,2}(N), \ldots, s_{B,N}(N))$ be the system signatures of two systems with N IID components, and let T_A and T_B be the lifetimes of the two systems. If

$$\sum_{j=i}^{N} s_{A,j}(N) \leq \sum_{j=i}^{N} s_{B,j}(N), \tag{3.36}$$

holds for all $i = 1, 2, \ldots, N$, then $T_A \leq_{st} T_B$.

Proposition 3.1 [108] cannot be employed directly to compare the lifetimes of the systems with different sizes. Thus, Navarro *et al.* [122] suggested "converting" the smaller of two systems into a system of the same size as the larger one by adding irrelevant components to the smaller system. Note that both systems have the same lifetime distribution because irrelevant components are added. Consequently, the lifetimes of the systems can be compared based on the system signature. According to

Navarro *et al.* [122], two systems with common IID component lifetimes are said to be "equivalent" if the distributions of the lifetimes of the systems are identical.

Proposition 3.2 (Navarro *et al.* [122])**.** Let $\boldsymbol{s}(N)$ be the system signatures of a system with N IID components. Then, for any positive integer v, the system signature $\boldsymbol{s}^*(N+v)$ of an equivalent system with $N+v$ components is given by

$$s_k^*(N+v) = \frac{N}{N+v}\frac{1}{\binom{N+v-1}{k-1}} \times \sum_{i=\max(1,k-v)}^{\min(k,N)} \binom{N-1}{i-1}\binom{v}{k-i} s_i(N), \tag{3.37}$$

for $k = 1, 2, \ldots, N+v$.

When we compare the lifetimes of systems with different sizes, Proposition 3.2 [122] enables us to find an equivalent system with the same size. As a result, we can use Proposition 3.1 [108] to compare the lifetimes of systems.

Although the usual stochastic order is useful when it is applicable, there exist systems that cannot be ordered in the usual stochastic order. Arcones *et al.* [121] introduced the concept of the stochastic precedence, which is an alternative approach to make a stochastic comparison between random variables. T_A is said to be smaller than T_B in the stochastic precedence (denoted by $T_A \leq_{sp} T_B$) if and only if $\Pr\{T_A \leq T_B\} \geq 0.5$. Hollander and Samaniego [123] used the stochastic precedence to compare the lifetimes of the systems.

Proposition 3.3 (Hollander and Samaniego [123])**.** Let $\boldsymbol{s}_A(N) = (s_{A,1}(N), s_{A,2}(N), \ldots, s_{A,N}(N))$ and $\boldsymbol{s}_B(M) = (s_{B,1}(M), s_{B,2}(M), \ldots, s_{B,M}(M))$ be the system signatures of two systems with N and M IID components, respectively, and let T_A and T_B be the lifetimes of two systems. If $\Pr\{T_A \leq T_B\} \geq 0.5$, then $T_A \leq_{sp} T_B$, where

$$\Pr\{T_A \leq T_B\} = \sum_{i=1}^{N}\sum_{j=1}^{M} s_{A,i}(N) s_{B,j}(M) \times \sum_{k=i}^{N} \left[\frac{\binom{N}{k}\binom{M}{j}}{\binom{N+M}{k+j}}\right] \left(\frac{j}{k+j}\right). \tag{3.38}$$

The main advantage of the stochastic precedence is that we can present orderings such that T_A stochastically precedes T_B or vice versa, even for systems that cannot be ordered in the usual stochastic order. The relationship between the usual stochastic order ($T_A \leq_{st} T_B$) and the stochastic precedence ($T_A \leq_{sp} T_B$) is given by [121],

$$T_A \leq_{st} T_B \Rightarrow T_A \leq_{sp} T_B. \tag{3.39}$$

Table 3.11: Orderings in the usual stochastic order among some Lin/(r,s)/(m,n):F systems.

(r,s)		$(2,2)$	$(2,3)$	$(3,3)$	$(2,2)$	$(2,3)$	$(3,3)$	$(3,3)$	$(3,4)$	$(4,4)$	$(3,3)$	$(3,4)$	$(4,4)$
	(m,n)	$(4,5)$	$(4,5)$	$(4,5)$	$(5,5)$	$(5,5)$	$(5,5)$	$(5,6)$	$(5,6)$	$(5,6)$	$(6,6)$	$(6,6)$	$(6,6)$
$(2,2)$	$(4,5)$	$=_{st}$	$\leq_{st}$	$\leq_{st}$	$\geq_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$
$(2,3)$	$(4,5)$		$=_{st}$	$\leq_{st}$	$\geq_{st}$	$\geq_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$	N/C	$\leq_{st}$	$\leq_{st}$
$(3,3)$	$(4,5)$			$=_{st}$	$\geq_{st}$	$\geq_{st}$	$\geq_{st}$	$\geq_{st}$	$\leq_{st}$	$\leq_{st}$	$\geq_{st}$	N/C	$\leq_{st}$
$(2,2)$	$(5,5)$				$=_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$
$(2,3)$	$(5,5)$					$=_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$	$\leq_{st}$	N/C	$\leq_{st}$	$\leq_{st}$
$(3,3)$	$(5,5)$						$=_{st}$	$\geq_{st}$	$\leq_{st}$	$\leq_{st}$	$\geq_{st}$	N/C	$\leq_{st}$
$(3,3)$	$(5,6)$							$=_{st}$	$\leq_{st}$	$\leq_{st}$	$\geq_{st}$	$\leq_{st}$	$\leq_{st}$
$(3,4)$	$(5,6)$								$=_{st}$	$\leq_{st}$	$\geq_{st}$	$\geq_{st}$	$\leq_{st}$
$(4,4)$	$(5,6)$									$=_{st}$	$\geq_{st}$	$\geq_{st}$	$\geq_{st}$
$(3,3)$	$(6,6)$										$=_{st}$	$\leq_{st}$	$\leq_{st}$
$(3,4)$	$(6,6)$											$=_{st}$	$\leq_{st}$
$(4,4)$	$(6,6)$												$=_{st}$

For illustration, we present the stochastic orderings among some Lin/(r,s)/(m,n):F systems using the system signatures obtained by the proposed method based on Theorem 3.1. Table 3.11 shows the orderings in the usual stochastic order among some Lin/(r,s)/(m,n):F systems. Let $T_{(r,s)(m,n)}$ be the lifetime of a Lin/(r,s)/(m,n):F system, and the inequality sign $\leq_{st}$ in the 3rd row and 4th column of Table 3.11 indicates the stochastic ordering $T_{(2,2)(4,5)} \leq_{st} T_{(2,3)(4,5)}$. As shown in Table 3.11, we identified the stochastic orderings in 62 pairs out of the 66 possible pairwise. Because the orderings of the remaining four pairs cannot be presented in the usual stochastic order, they are represented by "N/C." However, we can use the stochastic precedence to compare the lifetimes of the systems. By Proposition 3.3 [123], we have,

$$\Pr\{T_{(2,3)(4,5)} \leq T_{(3,3)(6,6)}\} = 0.607401733,$$
$$\Pr\{T_{(3,3)(4,5)} \leq T_{(3,4)(6,6)}\} = 0.494663828,$$
$$\Pr\{T_{(2,3)(5,5)} \leq T_{(3,3)(6,6)}\} = 0.679896023,$$
$$\Pr\{T_{(3,3)(5,5)} \leq T_{(3,4)(6,6)}\} = 0.577954080,$$

Thus, we can obtain the stochastic precedence orderings as follows:

$$
\begin{aligned}
T_{(2,3)(4,5)} &\leq_{sp} T_{(3,3)(6,6)}, \\
T_{(3,3)(4,5)} &\geq_{sp} T_{(3,4)(6,6)}, \\
T_{(2,3)(5,5)} &\leq_{sp} T_{(3,3)(6,6)}, \\
T_{(3,3)(5,5)} &\leq_{sp} T_{(3,4)(6,6)}.
\end{aligned}
$$

3.5 Summary

This chapter dealt with the system signature of a Lin/(r,s)/(m,n):F system and a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system. The system signature is connected to many other well-known reliability concepts and has various theoretical applications. For example, the system signature is used to establish stochastic comparisons among some systems. Also, the reliability polynomial can be easily expressed in terms of the system signature, which means that once a system signature is obtained, various reliability indices such as the reliability polynomial, failure rate, and mean time to failure are easy to derive. However, its practical applications have generally been limited to relatively small systems because of the difficulties of its computation. Thus, this chapter computed the system signature of a Lin/(r,s)/(m,n):F system and a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system. The main contribution lies in the derivation of the number of path sets for these systems with exactly z failed components, which can be used to compute the system signature. We conducted numerical experiments to investigate the efficiency of the proposed method and then found that the proposed methods were efficient. Also, we presented the stochastic orderings of some connected-**X**-out-of-(m,n):F lattice systems, and the orderings could determine the system that works properly for a longer time. From the discussions provided in this chapter, it concludes that the system signature could have a certain degree of utility in the context of the comparisons of the systems.

Chapter 4

Component Assignment Problem

The aim of this chapter is to develop algorithms for finding an optimal arrangement of a Lin/(r, s)/(m, n):F system. Section 4.1 introduces the notation used throughout this chapter and provides a detailed description of the CAP in a Lin/(r, s)/(m, n):F system. Section 4.2 develops an algorithm for finding the optimal arrangement. Section 4.3 proposes an algorithm that specializes to the case of $r = m - 1$ and $s = n - 1$. In Section 4.4, to investigate the efficiency of the proposed algorithms, we compare these algorithms with the existing algorithm through numerical experiment. Finally, Section 4.5 summarizes the contributions of the chapter.

4.1 CAP in a Lin/(r, s)/(m, n):F System

4.1.1 Formulation of the CAP

First, we describe the CAP in a Lin/(r, s)/(m, n):F system. Throughout this chapter, we assume that, not only

(a) each component and the system can have only two states: either working or failed,

(b) all components are mutually statistically independent,

(c) all components reliabilities are given,

but also

(d) the components are functionally exchangeable.

Assumption (d) means that a component can be assigned to any position.

Next, we define some notation. For $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, let (i, j) be a position at the ith row and the jth column in a system. The symbol $\pi(i, j)$ represents an index of a component assigned to position (i, j) for $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, where $\pi(i, j) \in \{1, 2, \ldots, mn\}$. In this chapter, a set of positions is called "area." In addition, an arrangement of mn components in which component $\pi(i, j)$ is assigned to position (i, j) is given by

$$\Pi = (\pi(i, j))_{1 \leq i \leq m, 1 \leq j \leq n}. \tag{4.1}$$

For $\tau \in \{1, 2, \ldots, mn\}$, let τ be an index of the component, and the reliability of component τ is denoted by p_τ, where $p_1 < p_2 < \cdots < p_{mn}$ without loss of generality. In other words, p_τ represents the reliability of the τth least reliable component. The vector $\boldsymbol{p}$ represents an mn-vector of the component reliabilities $(p_1, p_2, \ldots, p_{mn})$. We denote the reliability of the Lin/(r, s)/(m, n):F system under an arrangement Π with component reliabilities $\boldsymbol{p}$ by $R^L((r, s), (m, n), \boldsymbol{p}; \Pi)$.

Under the assumption that the components are functionally exchangeable, the system reliability may be improved by appropriately assigning components. The CAP is to find an arrangement that maximizes the reliability of the Lin/(r, s)/(m, n):F system (Π^*), that is,

$$\Pi^* = \arg\max_{\Pi \in \Omega} R^L((r, s), (m, n), \boldsymbol{p}; \Pi), \tag{4.2}$$

where Ω is a set of all arrangements.

4.1.2 Properties of the CAP in the Lin/(r, s)/(m, n):F System

In this subsection, we review the properties of the CAP in Lin/(r, s)/(m, n):F system. Recall that we should perform the pruning as many times as possible to enhance the speed of the B&B method. In the CAP, basically, there are three kinds of the pruning as follows [88]:

(a) pruning based on the necessary condition that the optimal arrangement must satisfy;

(b) pruning for eliminating the equivalent arrangement;

(c) pruning based on system reliability.

A component arrangement that does not satisfy the necessary conditions will not be optimal. Thus, we need to evaluate only the arrangements that satisfy all of the necessary conditions. By reducing the solution space according to the necessary conditions, it reduces the computational effort of finding the optimal arrangement. It is noteworthy that the magnitude of component reliabilities determines whether or not the necessary conditions are satisfied even if the exact component reliabilities are not known. This means that we do not need any computations of the system reliability at the time when we check whether or not the necessary conditions are satisfied. After that, to find the optimal arrangement, only the arrangements that satisfy all of the necessary conditions need to be evaluated.

Also, when we solve the CAP in a Lin/(r,s)/(m,n):F system, several equivalent arrangements exist. We now introduce the concept of equivalent arrangements.

Definition 4.1. Two component arrangements, denoted by Π_A, Π_B, with component reliabilities $\boldsymbol{p}$ are said to be *equivalent* if

$$R^L((r,s),(m,n),\boldsymbol{p};\Pi_A) = R^L((r,s),(m,n),\boldsymbol{p};\Pi_B), \tag{4.3}$$

holds for any $\boldsymbol{p}$.

Once one of the equivalent arrangements is enumerated, we do not need to enumerate the others when we find an optimal arrangement. By eliminating the equivalent assignment, the solution space can be reduced, and consequently, we can find the optimal arrangement efficiently.

Next, we provide the existing necessary conditions that the optimal arrangement must satisfy. Koutras *et al.* [90] established the necessary conditions for the optimal arrangement as follows:

Lemma 4.1 (Koutras *et al.* [90])**.** The optimal arrangement of the Lin/(r,s)/(m,n):F system satisfies the following conditions.

(a) If $2 \le j \le \min\{s, n-s+1\}$, for $i = 1,2,\ldots,m$, $\pi(i,j-1) < \pi(i,j)$.

(b) If $\max\{n-s+2, s+1\} \le j \le n$, for $i = 1,2,\ldots,m$, $\pi(i,j-1) > \pi(i,j)$.

(c) If $2 \le i \le \min\{r, m-r+1\}$, for $j = 1,2,\ldots,n$, $\pi(i-1,j) < \pi(i,j)$.

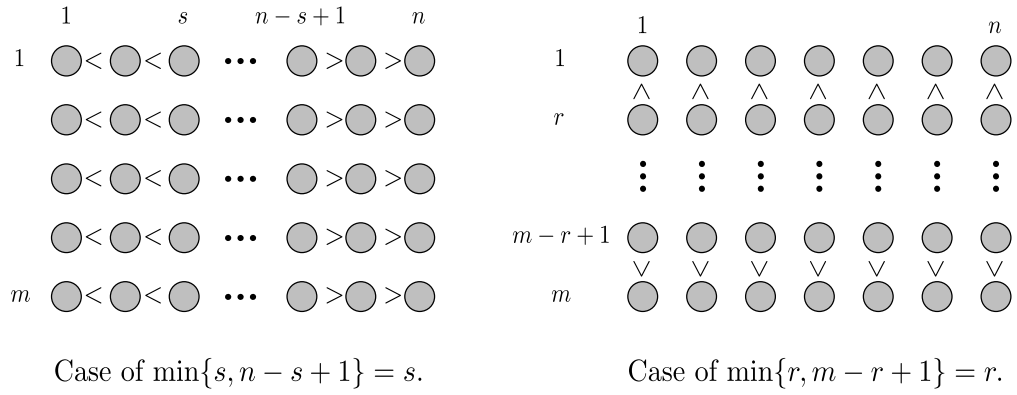


Figure 4.1: Explanation of Lemma 4.1.

(d) If $\max\{m-r+2, r+1\} \leq i \leq m$, for $j = 1, 2, \ldots, n$, $\pi(i-1, j) > \pi(i, j)$.

Note that $\pi(i, j-1) < \pi(i, j)$ implies $p_{\pi(i,j-1)} < p_{\pi(i,j)}$. Figure 4.1 explains Lemma 4.1, where the symbol "$<$ ($>$)" means the magnitude of the component reliabilities. Hence, Lemma 4.1 (a) means that, if a component arrangement is optimal, for $2 \leq j \leq \min\{s, n-s+1\}$, a component assigned to position (i, j) is more reliable than that assigned to position $(i, j-1)$. Broadly speaking, Lemma 4.1 states that the nearer to the center a component is located, the more reliable component should be assigned. Lemma 4.1 can be proved with the permutation importance [124, 125], where the definition of the permutation importance is given in the Appendix. Because Koutras *et al.* [90] did not provide the details of the proof, the Appendix provides the proof.

Omura *et al.* [93] gave the areas where exchanging any two components does not change the system reliability as follows:

Lemma 4.2 (Omura *et al.* [93])**.** Exchanging any two components in the following areas does not change the system reliability.

(a) If $2r > m$, for $j = 1, 2, \ldots, n$, $A_j = \{(i, j) \mid m - r + 1 \leq i \leq r\}$.

(b) If $2s > n$, for $i = 1, 2, \ldots, m$, $B_i = \{(i, j) \mid n - s + 1 \leq j \leq s\}$.

Since Lemma 4.2 was provided without the details of the proof in Omura *et al.* [93], the proof is presented in Appendix. Figure 4.2 explains Lemma 4.2. It is clear from

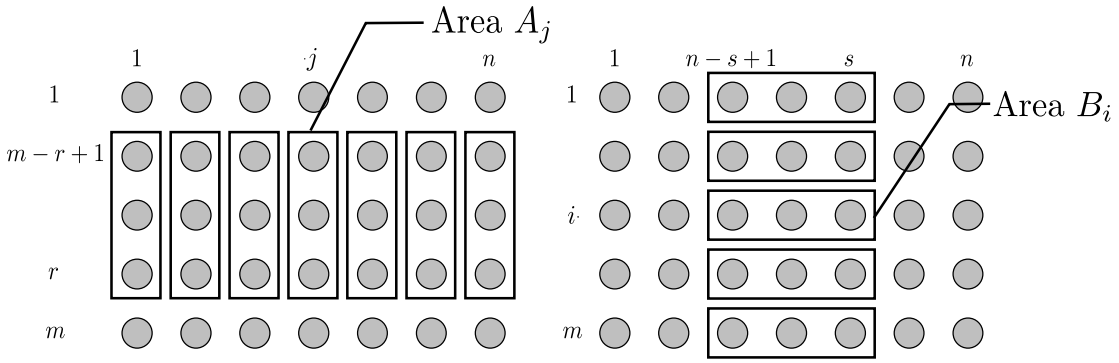


Figure 4.2: Explanation of Lemma 4.2.

Lemma 4.2 that one arrangement and an arrangement obtainable by swapping two components in area A_j or B_i are equivalent. Thus, by fixing the magnitude of the component reliabilities in each area, we delete all but one of the equivalent arrangements. Hence, we provide the following lemma.

Lemma 4.3. We enumerate the arrangements satisfying the following conditions in order to delete all but one of the equivalent arrangements.

(a) If $2r > m$ and $m - r + 2 \leq i \leq r$, $\pi(i, j-1) < \pi(i, j)$ for $j = 1, 2, \ldots, n$.

(b) If $2s > n$ and $n - s + 2 \leq j \leq s$, $\pi(i-1, j) < \pi(i, j)$ for $i = 1, 2, \ldots, m$.

Furthermore, Omura *et al.* [93] provided the following proposition.

Proposition 4.1 (Omura *et al.* [93])**.** The optimal arrangement of a Lin/(r, s)/(m, n):F system satisfies the following conditions. If $2r > m$ and $2s > n$, then the $(2r-m)(2s-n)$ most reliable components are assigned to area C, where

$$C = \{(i, j) \mid m - r + 1 \leq i \leq r, n - s + 1 \leq j \leq s\}.$$

Combining Lemmas 4.1 and 4.2 gives Proposition 4.1 straightforward. Figure 4.3 explains Proposition 4.1.

Finally, we explain the equivalent arrangements due to symmetry. When the component reliabilities are given, the reliability of a Lin/(r, s)/(m, n):F system is determined by the relative arrangements of components. Hence, the arrangements obtainable by

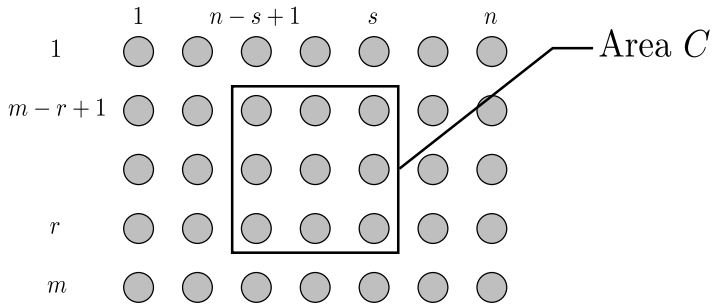


Figure 4.3: Explanation of Proposition 4.1.

flipping the left and right ends or the top and bottom of the arrangement or by rotating 180 degrees clockwise are equivalent. To delete all but one of the equivalent arrangements, we enumerate the arrangements that satisfy the following conditions.

(a) If $m \neq n$ or $r \neq s$, then

$$\pi(1,1) = \min\{\pi(1,1), \pi(m,1), \pi(1,n), \pi(m,n)\}, \tag{4.4}$$

(b) If $m = n$ and $r = s$, then Eq. (4.4) and

$$\pi(m,1) < \pi(1,n). \tag{4.5}$$

Equation (4.4) means that the least reliable component in positions $(1,1), (m,1), (1,n)$, and (m,n) should be assigned to position $(1,1)$. In this chapter, an arrangement that satisfies the above conditions is called an "arrangement with symmetry eliminated."

Let us consider a situation that we assign the components in order of positions $(1,1), (2,1), \ldots, (m,1), (1,2), \ldots, (m,2), (1,3), \ldots, (m,n)$. Yamamoto *et al.* [126] provided a condition for enumerating the only arrangements with symmetry eliminated. Now, we define the following notation. For $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, we have

$$w = i + (j-1)m, \tag{4.6}$$

where $w \in \{1, 2, \ldots, mn\}$. Let L_w denote a set of the components that have not been assigned when the wth component is assigned, and let $\max^{(z)}(L_w)$ denote the index of the zth most reliable component in the set L_w. Yamamoto *et al.* [126] gave the following

proposition for enumerating the only arrangements with symmetry eliminated.

Proposition 4.2 (Yamamoto *et al.* [126])**.** When we assign the components in order of positions $(1,1),(2,1),\ldots,(m,1),(1,2),\ldots,(m,2),(1,3),\ldots,(m,n)$, the arrangements with symmetry eliminated satisfy the following condition:
(1) If $m \neq n$ or $r \neq s$, then

(a) for $w = 1$, $\pi(1,1) < mn - 3$.

(b) for $1 < w < m$, $\max^{(3)}(L_w) > \pi(1,1)$.

(c) for $w = m$, $\pi(1,1) < \pi(m,1)$.

(d) for $m < w < mn - m + 1$, $\max^{(2)}(L_w) > \pi(1,1)$.

(e) for $w = mn - m + 1$, $\pi(1,1) < \pi(1,n)$.

(f) for $mn - m + 1 < w < mn$, $\max^{(1)}(L_w) > \pi(1,1)$.

(g) for $w = mn$, $\pi(1,1) < \pi(m,n)$.

(2) If $m = n$ and $r = s$, then the above conditions (a), (b), (f), and (g) and

(c') for $w = m$, $\pi(1,1) < \pi(m,1) \leq mn - 1$.

(d') for $m < w < mn - m + 1$, $\max^{(2)}(L_w) > \pi(1,1)$ and $\max^{(1)}(L_w) > \pi(m,1)$.

(e') for $w = mn - m + 1$, $\pi(m,1) < \pi(1,n)$.

Proposition 4.2 (a) implies that the index of the component that assigned to position $(1,1)$ is smaller than $(mn-3)$. This proposition enables us to keep only one arrangement and removes the other arrangements that are equivalent to the arrangement. Note that, although Yamamoto *et al.* [126] gave the above condition to the facility layout problem, it is also applicable to the CAP in the Lin/(r,s)/(m,n):F system.

4.2 Algorithm for Solving the CAP in the General Case

4.2.1 Conditions for the Pruning

In our proposed algorithm, when enumerating all possible candidates for the optimal arrangement, we assign the components in the order of positions

$(1,1),(2,1),\ldots,(m,1),(1,2),\ldots,(m,2),(1,3),\ldots,(m,n)$ in the depth-first search (DFS) tree. To enhance the speed of the algorithm, we should perform the pruning as many times as possible in the DFS tree. Thus, we derive three conditions for the pruning of the DFS tree, and they are based on Lemmas 4.1, 4.3, Propositions 4.1, and 4.2.

First, we provide the following condition.

Condition 1. When a component is assigned to position (i,j), if the following condition is not satisfied, then we perform the pruning.

(i) If $2 \le j \le s$, $\pi(i,j-1) < \pi(i,j)$ for $i = 1,2,\ldots,m$.

(ii) If $\max\{n-s+2, s+1\} \le j \le n$, $\pi(i,j-1) > \pi(i,j)$ for $i = 1,2,\ldots,m$.

(iii) If $2 \le i \le r$, $\pi(i-1,j) < \pi(i,j)$ for $j = 1,2,\ldots,n$.

(iv) If $\max\{m-r+2, r+1\} \le i \le m$, $\pi(i-1,j) > \pi(i,j)$ for $j = 1,2,\ldots,n$.

The proof follows directly from Lemmas 4.1 and 4.3.

Next, we provide a condition, which restricts the components assigned to position $(1,1)$.

Condition 2. When a component is assigned to position $(1,1)$, if the following condition is not satisfied, then we perform the pruning.

(i) If $2r \ge m$ and $2s \ge n$, then $\pi(1,1) = 1$.

(ii) If $2r \ge m$ and $2s < n$, then $\pi(1,1) \le mn - 2sm + 1$.

(iii) If $2r < m$ and $2s \ge n$, then $\pi(1,1) \le mn - 2rn + 1$.

(iv) If $2r < m$ and $2s < n$, then $\pi(1,1) \le mn - 4rs + 1$.

Recall that component τ means the τth least reliable component ($\tau \in \{1,2,\ldots,mn\}$). For example, Condition 2 states that, if $2r \ge m$ and $2s < n$, then the τth least reliable component ($1 \le \tau \le mn - 2sm + 1$) should be assigned to position $(1,1)$ in the optimal arrangement. This condition restricts the components assigned to position $(1,1)$, and consequently, the number of candidates for the optimal arrangement can be reduced. Condition 2 can be easily established from Lemmas 4.1 and 4.3.

Conditions 1 and 2 can reduce the small number of the possible solutions, and subsequently, the remaining possible solutions are evaluated by the system reliability. When finding the optimal arrangement, we compute the system reliability again and again, and thus, the efficient computations can be very critical. In the proposed algorithm, we use the recursive equation [58].

We define some additional notation to provide the recursive equation. A vector $\boldsymbol{x} = (x_1, x_2, \ldots, x_m)$ represents the states of the m components assigned on the jth column, where $x_i = 0$ if the ith component works; otherwise $x_i = 1$ for $i = 1, 2, \ldots, m$. Also, we introduce an $(m-r+1)$-dimensional 0-1 binary vector $\boldsymbol{d} = (d_1, d_2, \ldots, d_{m-r+1})$, and the element d_k is defined, for $k = 1, 2, \ldots, m-r+1$, as

$$d_k = \prod_{u=k}^{k+r-1} x_u. \tag{4.7}$$

In words, $d_k = 1$ if all the components $\pi(k, j), \pi(k+1, j), \ldots, \pi(k+r-1, j)$ fail; otherwise $d_k = 0$. Moreover, for $j = 1, 2, \ldots, n$ and $\boldsymbol{d} \in \{0, 1\}^{m-r+1}$, a probability $F_j(\boldsymbol{d})$ is defined as

$$F_j(\boldsymbol{d}) = \sum_{\boldsymbol{x} \in U(\boldsymbol{d})} \prod_{i=1}^{m} {p_{\pi(i,j)}}^{1-x_i} (1 - p_{\pi(i,j)})^{x_i}, \tag{4.8}$$

where

$$U(\boldsymbol{d}) = \left\{ \boldsymbol{x} \in \{0, 1\}^{m-r+1} \;\middle|\; \prod_{u=k}^{k+r-1} x_u = d_k, (k = 1, 2, \ldots, m-r+1) \right\}. \tag{4.9}$$

Let $\boldsymbol{g}$ be an $(m-r+1)$-dimensional vector $(g_1, g_2, \ldots, g_{m-r+1})$, where $g_k \in \{0, 1, \ldots, s-1\}$ for $k = 1, 2, \ldots, m-r+1$. For $j = 1, 2, \ldots, n$, we define the following indicator variable:

$$A_j(\boldsymbol{g}) = \prod_{k=1}^{m-r+1} \left(\left(\prod_{u=k}^{k+r-1} \prod_{v=j-g_k+1}^{j} Z_{uv} \right) \times \left(1 - \prod_{u=k}^{k+r-1} Z_{u,j-g_k} \right) \right), \tag{4.10}$$

where $\prod_{i=a}^{b} x_i = 1$ for $a > b$. Here, the event $\{A_j(\boldsymbol{g}) = 1\}$ means that "all the components fail in an $r \times g_k$ rectangle with four corners $(k, j-g_k+1)$, (k, j), $(k+r-1, j)$, and $(k+r-1, j-g_k+1)$" and "at least one component works in components $(k, j-g_k)$,

$(k+1, j-g_k), \ldots, (k+r-1, j-g_k)$," and 0 otherwise for all $k = 1, 2, \ldots, m-r+1$. Besides, the set S is defined as

$$S = \left\{ \boldsymbol{g} \;\middle|\; \prod_{k=1}^{m-r+1} \mathcal{X}(g_k < s) = 1 \right\}, \tag{4.11}$$

where $\mathcal{X}(G)$ is an indicator function which takes 1 if argument G is true; 0 otherwise. Moreover, Yamamoto and Akiba [62] proved that $\boldsymbol{g}$s are impossible for $\boldsymbol{g} \in E$, where

$$E = \left\{ \boldsymbol{g} \in S \;\middle|\; \begin{array}{l} g_k > g_{k+1} \text{ and } g_{k+x} < g_{k+x+1} \text{ for some } k \in \{1, 2, \ldots, m-r-1\} \\ \text{and some } x \in \{1, 2, \ldots, r-1\} \end{array} \right\}. \tag{4.12}$$

For $j = 1, 2, \ldots, n$ and $\boldsymbol{g} \in S$, the reliability of the Lin/(r, s)/(m, j):F system in which the states of the components on the end are given by $\boldsymbol{g}$, denoted by $R^L((r, s), (m, j), \boldsymbol{p}; \Pi^{(j)}; \boldsymbol{g})$, is defined by

$$R^L((r, s), (m, j), \boldsymbol{p}; \Pi^{(j)}; \boldsymbol{g}) = \Pr\left\{ \bigcap_{k=1}^{m-r+1} \bigcap_{l=1}^{j-s+1} \left\{ \prod_{u=k}^{k+r-1} \prod_{v=l}^{l+s-1} Z_{uv} = 0 \right\} \bigcap A_j(\boldsymbol{g}) \right\}. \tag{4.13}$$

Here, $\Pi^{(j)}$ is an arrangement where the components have been assigned from the 1st column to the jth column for $j = 1, 2, \ldots, n$. Note that $\Pi^{(j)} = \Pi$ if $j = n$. Furthermore, let $\boldsymbol{g}'$ be an $(m-r+1)$-dimensional integer vector $(g'_1, g'_2, \ldots, g'_{m-r+1})$, where $g'_k \in \{0, 1, \ldots, s-1\}$ for $k = 1, 2, \ldots, m-r+1$. The states of the components on the end of a Lin/(r, s)/$(m, j-1)$:F system are given by $\boldsymbol{g}'$.

We can compute the reliability by using the following recursive equation.

Proposition 4.3 (Yamamoto and Miyakawa [58])**.** (a) For $j = 1, 2, \ldots, n$ and $\boldsymbol{g} \in S$,

$$R^L((r, s), (m, j), \boldsymbol{p}; \Pi^{(j)}; \boldsymbol{g}) = \begin{cases} 0, & \min\limits_{1 \le k \le m-r+1} \{g_k\} \le 0 \text{ or } \boldsymbol{g} \in E. \\ F_j(\boldsymbol{d}) \sum\limits_{\boldsymbol{g}' \in \Theta(\boldsymbol{g})} R^L((r, s), (m, j-1), \boldsymbol{p}; \Pi^{(j-1)}; \boldsymbol{g}'), & \\ & \text{otherwise}, \end{cases} \tag{4.14}$$

where

$$\Theta(\boldsymbol{g}) = \left\{ \boldsymbol{g}' \;\middle|\; \begin{array}{ll} g'_l \in \{0, 1, \ldots, s-1\}, & \text{if } g_k = 0, \\ g'_l \in \{g_l - 1\}, & \text{if } g_k > 0 \end{array} \right\}, \tag{4.15}$$

and for $k = 1, 2, \ldots, m - r + 1$,

$$d_k = \begin{cases} 0, & \text{if } g_k = 0, \\ 1, & \text{if } g_k > 0. \end{cases}$$

As the boundary condition, for $j = 0$,

$$R^L((r,s),(m,j),\boldsymbol{p};\Pi^{(j)};\boldsymbol{g}) = \begin{cases} 1, & \text{if } \boldsymbol{g} = (0,\ldots,0), \\ 0, & \text{otherwise.} \end{cases} \tag{4.16}$$

(b)

$$R^L((r,s),(m,n),\boldsymbol{p};\Pi) = \sum_{\boldsymbol{g}\in S\setminus E} R^L((r,s),(m,n),\boldsymbol{p};\Pi^{(n)};\boldsymbol{g}), \tag{4.17}$$

where the sets S and E are given by Eqs. (4.11) and (4.12).

In solving the CAP, Proposition 4.3 is very useful in terms of computing the system reliability. The reasons are as follows. When solving an optimal arrangement problem, we require iterative computations of the system reliability. If $R((r,s),(m,j),\boldsymbol{p};\Pi^{(j)};\boldsymbol{g})$s are memorized for $\boldsymbol{g} \in S$ in each j in the DFS tree, $R((r,s),(m,j),\boldsymbol{p};\Pi^{(j)};\boldsymbol{g})$s do not need to be computed again because we can utilize the memorized values, which enables us to avoid redundant computations. Therefore, the proposed algorithm can be accelerated with the help of the recursive equation [58]. An example of this computation of the system reliability is given later.

Finally, an algorithm can be accelerated with the help of the upper bound of the system reliability. It is evident from Proposition 4.3 that the system reliability is non-increasing in the number of components. To put it differently, even if further components are assigned to the arrangement, a solution is not as good as the current best solution. Thus, we can eliminate unnecessary arrangements by computing the upper bound of the system reliability under a partial arrangement and then comparing it with the maximum system reliability obtained so far. More precisely, if the system

reliability under a partial arrangement is lower than the maximum system reliability, then we perform the pruning. Thus, we derive the following condition.

Condition 3. When a component is assigned to position (m, j) for $j = 1, 2, \ldots, n$, if the following condition is not satisfied, then we perform the pruning.

$$\mathbf{Rmax} \leq \sum_{\boldsymbol{g} \in \{1,2,\ldots,s\}^{m-r+1}} R((r,s),(m,j),\boldsymbol{p};\Pi^{(j)};\boldsymbol{g}), \tag{4.18}$$

where **Rmax** is the maximum system reliability obtained so far.

4.2.2 Algorithm and Illustrative Example

In this subsection, we propose an algorithm based on the B&B method to search for finding the optimal arrangement of the Lin/(r, s)/(m, n):F system. The algorithm first begins by setting parameters: r, s, m, n, and $\boldsymbol{p}$, and then, we set **Rmax** to be 0. Next, Proposition 4.1 is used in a preprocessing step. To be precise, if $2r > m$ and $2s > n$, then we assign the $(2r-m)(2s-n)$ most reliable components to area C. It is noteworthy that the components assigned to area C are determined according to the magnitude of component reliabilities but not their actual values, and hence, we can identify the components without any computations of the system reliability. Proposition 4.1 is attractive, especially when $2r-m$ and $2s-n$ are large. After that, we assign the components in order of positions $(1,1),(2,1),\ldots,(m,1),(1,2),\ldots,(m,2),(1,3),\ldots,(m,n)$ in the DFS tree. Note that if a component has already assigned to the position by Proposition 4.1, we skip assigning a component at the position. Letting α be the number of assigned components, then we get

$$\alpha = \begin{cases} mn-(2r-m)(2s-n), & \text{if } 2r > m \text{ and } 2s > n, \\ mn, & \text{otherwise.} \end{cases} \tag{4.19}$$

As an example, Fig. 4.4 shows the assignment order of α (= 20) components in the Lin/$(3, 4)$/$(4, 6)$:F system. In this figure, a black cell means a position where a component has been assigned to advance, and a number in a cell means the assignment order of components in the proposed algorithm.

Next, let USED(τ) denote an index function that takes 1 if component τ has been assigned; otherwise, 0 for $\tau = 1, 2, \ldots, mn$. When we assign the components in order

1	5	9	11	13	17
2	6			14	18
3	7			15	19
4	8	10	12	16	20

Figure 4.4: Example of assignment order of components in the proposed algorithm.

of positions $(1,1),(2,1),\ldots,(m,1),(1,2),\ldots,(m,2),(1,3),\ldots,(m,n)$ in the DFS tree, DFS(k) is denoted by a routine to assign a component to the kth position for $k = 1,2,\ldots,\alpha$. Figure 4.5 shows a flow chart of DFS(k). The procedure of DFS(k) is given in the following steps.

STEP 0: Set $\tau = 0$.

STEP 1: Update $\tau = \tau + 1$. If $\tau > \alpha$, then stop DFS(k).

STEP 2: If component τ is used already, that is, USED(τ) $= 1$, then go back to **STEP 1**.

STEP 3: If Condition 1, 2, or Proposition 4.2 is not satisfied, then we perform the pruning, that is, go back to **STEP 1**.

STEP 4: If a component is assigned to position (m,j), for $j = 1,2,\ldots,n$, then we compute the system reliability under a partial arrangement by Proposition 4.3; otherwise, go to **STEP 7**.

STEP 5: If Condition 3 is not satisfied, then we perform the pruning, that is, go back to **STEP 1**.

STEP 6: If all the components have been assigned, then memorize the arrangement and update **Rmax**, and then, stop DFS(k); otherwise, go to **STEP 7**.

STEP 7: Go to DFS($k+1$).

After enumerating all arrangements, an arrangement with **Rmax** becomes the optimal arrangement.

We give an example to illustrate how to implement the proposed algorithm for finding the optimal arrangement to make the proposed algorithm more understandable. As

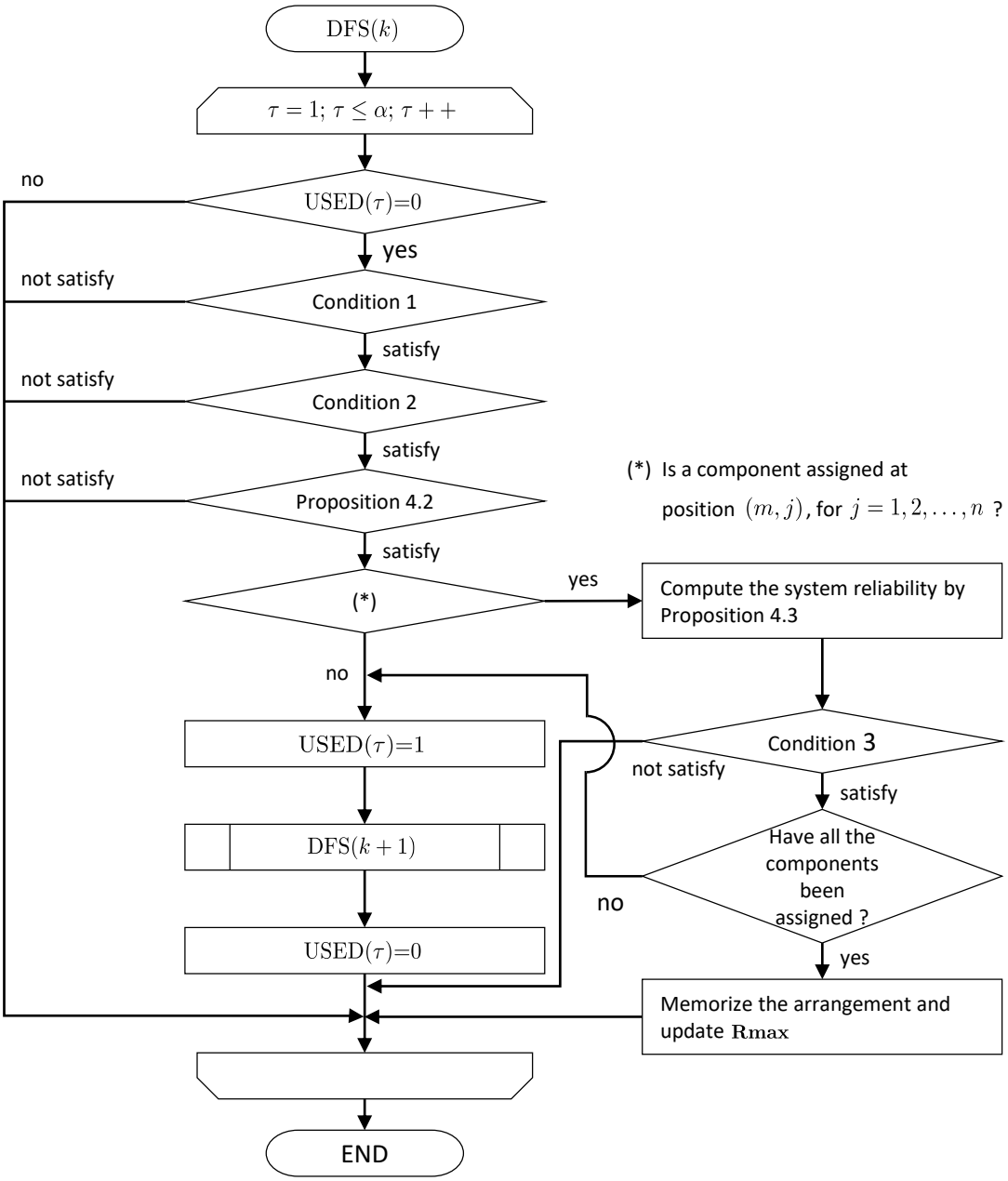


Figure 4.5: Flow chart of DFS(k).

an example, let us consider a Lin/$(2,3)$/$(3,4)$:F system. First of all, as a preprocessing step, components 11 and 12 are assigned to positions $(2,3)$ and $(2,4)$ according to Proposition 4.1. A tree for enumerating arrangements of this system is shown in Fig. 4.6. Note that a black cell means a position where a component has already been assigned, and a number in a cell means the index of components. In the following, for the sake of conciseness, we explain the procedure for each layer, although the actual algorithm uses the DFS strategy.

In the first layer (DFS(1)), if component 1 is assigned to position $(1,1)$, go to DFS(2). In contrast, if the other components are assigned to position $(1,1)$, we perform the pruning according to Condition 2 because, in the case of $2r \geq m$ and $2s \geq n$, these arrangements do not satisfy Condition 2 (i) ($\pi(1,1) = 1$). In the second layer (DFS(2)), components $2, 3, \ldots, 10$ are assigned to position $(2,1)$. Here, we focus on the case where component 6 is assigned.

In the third layer (DFS(3)), if components $2, \ldots, 5$ are assigned to position $(3,1)$, we compute the reliability of the system with the partial arrangements by Proposition 2. In contrast, if the other components are assigned to position $(3,1)$, we perform the pruning according to Condition 1 because, these arrangements do not satisfy the Condition 1 (iii) ($\pi(2,1) > \pi(3,1)$), namely, Lemma 4.1, and thus, go back to DFS(2).

In the seventh layer (DFS(7)), the first, third, and forth arrangements from the left do not satisfy Condition 1 (i) ($\pi(3,2) < \pi(3,3)$). Thus, we perform the pruning and then go back to DFS(6). In contrast, we compute the reliabilities of the systems with the other arrangements by Proposition 2. We denote $R^L((2,3),(3,j),\boldsymbol{p};\Pi^{(j)};\boldsymbol{g})$ as $R_j(\boldsymbol{g})$ to simplify the expression. When computing the reliability of the Lin/$(2,3)$/$(3,4)$:F system, we use the following recursive equations. For $j = 1, 2, 3, 4$,

$$
\begin{aligned}
R_j((2,2)) =& (1 - p_{\pi(1,j)}q_{\pi(2,j)}q_{\pi(3,j)} + q_{\pi(1,j)}q_{\pi(2,j)}p_{\pi(3,j)} + q_{\pi(1,j)}q_{\pi(2,j)}q_{\pi(3,j)}) \\
& \times [R_{j-1}((2,2)) + R_{j-1}((2,1)) + R_{j-1}((1,2)) + R_{j-1}((1,1))], \\
R_j((2,1)) =& p_{\pi(1,j)}q_{\pi(2,j)}q_{\pi(3,j)}[R_{j-1}((2,2)) + R_{j-1}((1,2))], \\
R_j((1,2)) =& q_{\pi(1,j)}q_{\pi(2,j)}p_{\pi(3,j)}[R_{j-1}((2,2)) + R_{j-1}((2,1))], \\
R_j((1,1)) =& q_{\pi(1,j)}q_{\pi(2,j)}q_{\pi(3,j)}R_{j-1}((2,2)),
\end{aligned}
$$

which are obtained from Eq. (4.14). The above equations show that $R_j(\boldsymbol{g})$s are computed from $R_{j-1}(\boldsymbol{g})$s for $\boldsymbol{g} \in \{(2,2),(2,1),(1,2),(1,1)\}$.

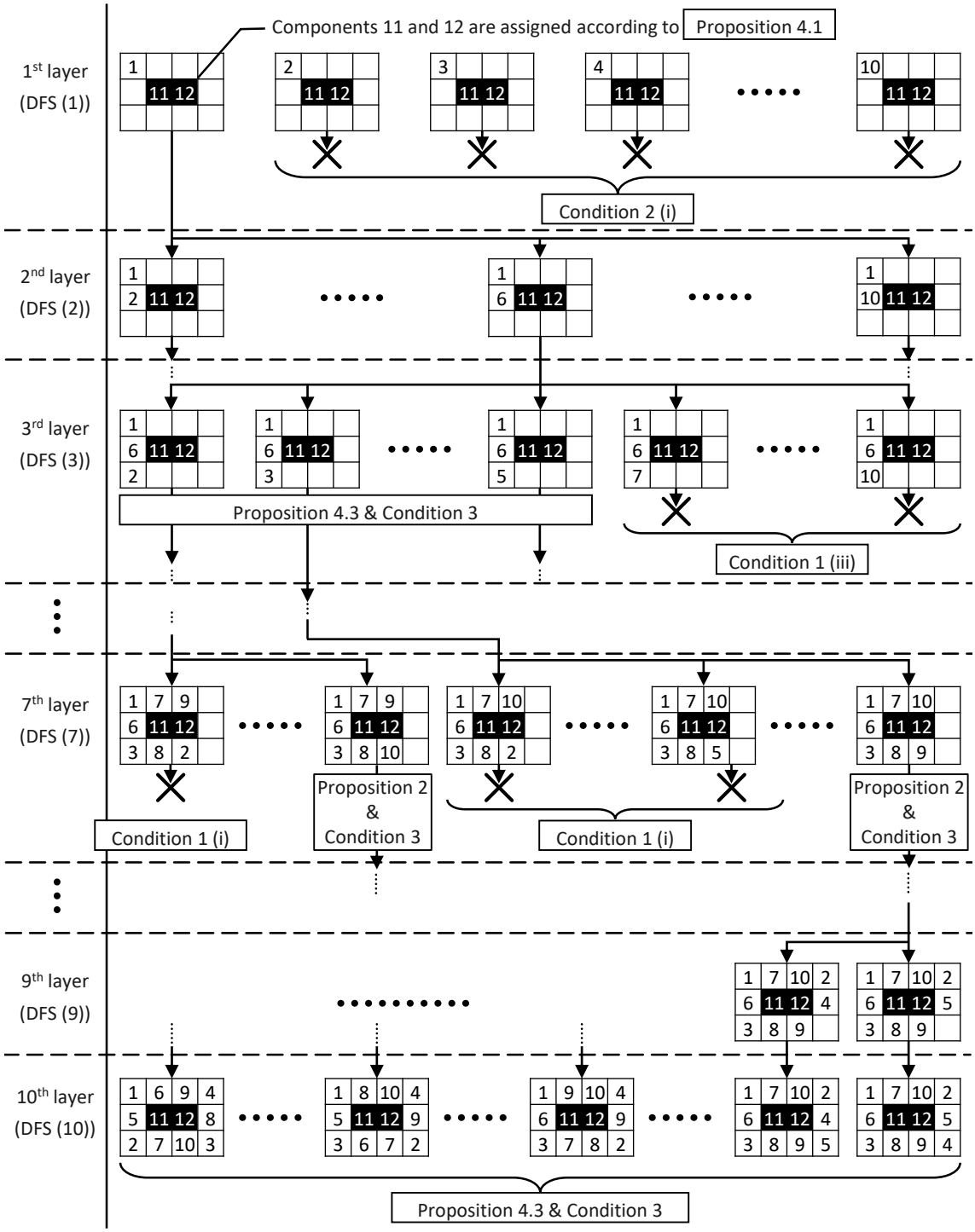


Figure 4.6: Tree for enumerating arrangements of the Lin/$(2,3)$/$(3,4)$:F system.

Here, we define the following partial arrangements $\Pi_A^{(3)}$ and $\Pi_B^{(3)}$.

$$\Pi_A^{(3)} = \left(\begin{array}{cc|c|c} 1 & 7 & 9 & \cdot \\ 6 & 11 & 12 & \cdot \\ 3 & 8 & 10 & \cdot \end{array}\right), \tag{4.20}$$

and

$$\Pi_B^{(3)} = \left(\begin{array}{cc|c|c} 1 & 7 & 10 & \cdot \\ 6 & 11 & 12 & \cdot \\ 3 & 8 & 9 & \cdot \end{array}\right), \tag{4.21}$$

where " $\cdot$ " in the matrices means that a component has not been assigned to the position. Note that Eqs. (4.20) and (4.21) correspond to the second and fifth arrangements from the left in the seventh layer of Fig. 4.6, respectively. As shown in Eqs. (4.20) and (4.21), both systems have common arrangement in the first and second columns. When computing the reliability of the system with arrangement $\Pi_A^{(3)}$, we memorize $R_2(\boldsymbol{g})$ for $\boldsymbol{g} \in \{(2,2),(2,1),(1,2),(1,1)\}$. Then, when computing the reliability of the system with $\Pi_B^{(3)}$, we utilize these memorized values. Consequently, we can avoid redundant computations, which enables us to compute the reliability of the system with a small amount of computation.

After that, the computed system reliabilities are compared with the maximum system reliability obtained so far (that is, **Rmax**) according to Condition 3. If it is smaller than **Rmax**, we perform the pruning, and then, go back to DFS(6), and otherwise go to DFS(8).

In the tenth layer (DFS(10)), we compute the system reliability for each arrangement. If the system reliability with an arrangement is higher than **Rmax**, we memorize the arrangement and update **Rmax**. If all the arrangements have already been enumerated, eventually, the arrangement with **Rmax** becomes optimal, and otherwise, go back to DFS(9).

4.3 Algorithm for Solving the CAP in the Case of $r = m - 1$ and $s = n - 1$

When we find the optimal arrangement of a large Lin/(r, s)/(m, n):F system, even if we use the proposed algorithm, it may be time-consuming, sometimes impossible to find the optimal arrangement. In this section, to enhance the efficiency, we develop an algorithm for finding the optimal arrangement of a Lin/(r, s)/(m, n):F system, which is specialized for the case of $r = m - 1$ and $s = n - 1$. This algorithm is based on a B&B method, which is common to the algorithm proposed in Section 4.2, but the way of assigning the components is completely different. As a result, the algorithm according to the structure of the problem can make a considerable reduction in computational efforts.

Recall that a set of positions is called "area" in this paper. As shown in Fig. 4.7, we consider nine areas in a Lin/$(m-1, n-1)$/(m, n):F system. The areas are expressed as

$$
\begin{aligned}
&A_{TL} = \{(1,1)\}, && B_L = \{(i,1) \mid 2 \le i \le m-1\},\\
&A_{BL} = \{(m,1)\}, && B_B = \{(m,j) \mid 2 \le j \le n-1\},\\
&A_{BR} = \{(m,n)\}, && B_R = \{(i,n) \mid 2 \le i \le m-1\},\\
&A_{TR} = \{(1,n)\}, && B_T = \{(1,j) \mid 2 \le j \le n-1\},\\
& && C = \{(i,j) \mid 2 \le i \le m-1, 2 \le j \le n-1\}.
\end{aligned}
$$

4.3.1 Conditions for the Pruning

From Proposition 4.1, the $(m-2)(n-2)$ most reliable components are assigned to area C. Also, from Condition 2, the optimal arrangement satisfies $\pi(1,1) = 1$, which means that the least reliable component is assigned to position $(1,1)$. Thus, $(m-2)(n-2)+1$ components have been assigned to the system in advance. After that, we assign the remaining $2(m+n)-5$ components to areas A_{BL}, A_{BR}, A_{TR}, B_L, B_B, B_R, and B_T. In our proposed algorithm, when enumerating all possible candidates for the optimal arrangement, we assign the components in the order of positions $(2,1), (3,1), \ldots, (m,1), (m,2), \ldots, (m,n), (m-1,n), \ldots, (1,n), (1,n-1), \ldots, (1,2)$ in the DFS tree as shown in Fig. 4.8. For efficiently finding the optimal arrangement, we

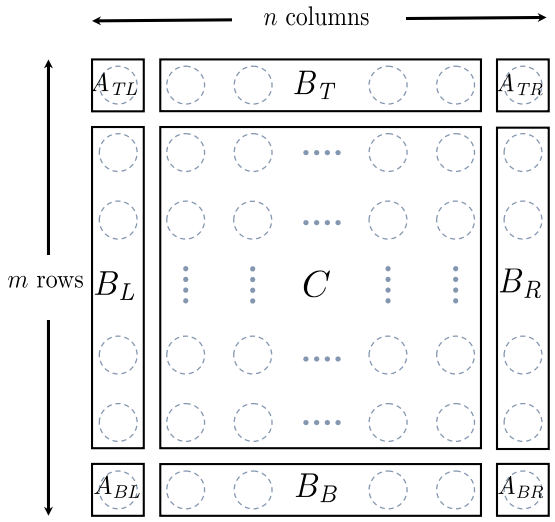


Figure 4.7: System divided into nine areas.

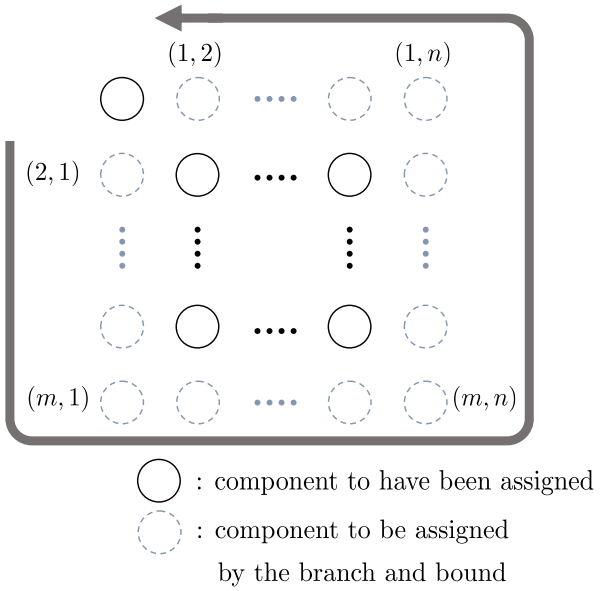


Figure 4.8: Order for assigning components.

should perform the pruning as many times as possible. In this subsection, we derive three conditions for the pruning of the DFS tree.

First, we provide the following condition.

Condition 4. When a component is assigned to position (i, j), if the following condition is not satisfied, then we perform the pruning.

(i) For $i = 3, 4, \ldots, m-1$, $\pi(i-1, 1) < \pi(i, 1)$.

(ii) For $j = 2, 3, \ldots, n-1$, $\pi(m, j-1) < \pi(m, j)$.

(iii) For $i = 2, 3, \ldots, m-1$, $\pi(i+1, n) < \pi(i, n)$.

(iv) For $j = 2, 3, \ldots, n-1$, $\pi(1, j+1) < \pi(1, j)$.

This condition can be proved from Lemmas 4.1 and 4.3 directly.

From Lemma 4.1, the least reliable component in each minimal cut is assigned to the corners in the optimal arrangement. Accordingly, we give Condition 5.

Condition 5. If the following condition is not satisfied, then we perform the pruning.

(i) When a component is assigned to position $(m, 1)$, $\pi(m, 1) < \pi(2, 1)$.

(ii) When a component is assigned to position (m, n), $\pi(m, n) < \pi(m, 2)$.

(iii) When a component is assigned to position $(1, n)$, $\pi(1, n) < \pi(m-1, n)$.

Next, we explain the equivalent arrangements due to symmetry. In Lin/$(m-1, n-1)$/(m, n):F systems, the relative relationship of component arrangements determines the system reliability, and hence, there exist several equivalent arrangements. More specifically, if $m = n$, which means that the system is square, there exist eight equivalent arrangements. To delete all but one of the equivalent arrangements, we enumerate the arrangements that satisfy the following conditions.

Condition 6. If $m = n$, when a component is assigned to position $(1, n)$, if the following condition is not satisfied, then we perform the pruning.

$$\pi(m, 1) < \pi(1, n).$$

This condition can be straightforward derived from Eq. (4.5). Condition 6 eliminates the arrangement obtained by rotating around the axis connecting positions $(m, 1)$ and $(1, n)$ of an arrangement. Note that the other equivalent arrangement due to structural symmetry has been eliminated because the least reliable component is assigned to position $(1, 1)$.

We consider the reliability of a Lin/$(m-1,n-1)$/(m,n):F system, which is denoted by $R^L((m-1,n-1),(m,n),P)$. Then, we have

$$R^L((m-1,n-1),(m,n),P) = \Pr\left\{\bigcap_{k=1}^{2}\bigcap_{l=1}^{2}\left\{\prod_{u=k}^{k+r-2}\prod_{v=l}^{l+s-2} Z_{uv} = 0\right\}\right\}. \tag{4.22}$$

We define the following events:

$$\begin{aligned}
\mathscr{C} &= \left\{\prod_{u=2}^{m-1}\prod_{v=2}^{n-1} Z_{uv} = 0\right\},\\
\mathscr{Z}_1 &= \left\{\prod_{v=2}^{n-1} Z_{1v} \times Z_{1,1} \times \prod_{u=2}^{m-1} Z_{u1} = 0\right\},\\
\mathscr{Z}_2 &= \left\{\prod_{u=2}^{m-1} Z_{u1} \times Z_{m1} \times \prod_{v=2}^{n-1} Z_{mv} = 0\right\},\\
\mathscr{Z}_3 &= \left\{\prod_{v=2}^{n-1} Z_{mv} \times Z_{mn} \times \prod_{u=2}^{m-1} Z_{un} = 0\right\},\\
\mathscr{Z}_4 &= \left\{\prod_{u=2}^{m-1} Z_{un} \times Z_{1n} \times \prod_{v=2}^{n-1} Z_{1v} = 0\right\}.
\end{aligned}$$

Note that the event $\{\mathscr{Z}_1 \cup \mathscr{C}\}$ means that at least one component works in the $m-1 \times n-1$ rectangle with four corners $(1,1)$, $(m-1,1)$, $(m-1,n-1)$, and $(1,n-1)$. Equation (4.22) can then be rewritten as follows:

$$\begin{aligned}
R^L((r,s),(m,n),P) &= \Pr\left\{\bigcap_{i=1}^{4}\{\mathscr{Z}_i \cup \mathscr{C}\}\right\},\\
&= \Pr\left\{\bigcap_{i=1}^{4}\{\mathscr{Z}_i \cap \mathscr{C}^c\} \cup \mathscr{C}\right\},\\
&= \Pr\{\mathscr{C}\} + \Pr\left\{\bigcap_{i=1}^{4}\{\mathscr{Z}_i\}\right\}(1 - \Pr\{\mathscr{C}\}),
\end{aligned} \tag{4.23}$$

which states as follows: Because all the minimal cuts contain area C, if at least one component in area C works, then the Lin/$(m-1,n-1)$/(m,n):F system works (with probability 1). If all the components in area C fail, and all the components in

- $B_T \cup A_{TL} \cup B_L$,

- $B_L \cup A_{BL} \cup B_B$,
- $B_B \cup A_{BR} \cup B_R$, or
- $B_R \cup A_{TR} \cup B_T$

fail, then the Lin/$(m-1, n-1)$/(m, n):F system fails. Consequently, we can get

$$\begin{aligned}\Pr\{\mathscr{C}\} &= 1 - \prod_{(i,j)\in C} \big(1 - p_{\pi(i,j)}\big), \\ &= 1 - \prod_{\tau=2(m+n)-3}^{mn} (1 - p_\tau)\,. \qquad \text{(from Lemma 4.1)} \end{aligned} \tag{4.24}$$

Since the components that are assigned to area C and position $(1, 1)$ can be identified, to find the optimal arrangement of a Lin/$(m-1, n-1)$/(m, n):F system, we only have to find a component arrangement such that $\Pr\big\{\bigcap_{i=1}^{4}\{\mathscr{Z}_i\}\big\}$ is maximized from Eq. (4.23). In other words, the CAP of a Lin/$(m-1, n-1)$/(m, n):F system can be simplified to a problem of finding a component arrangement such that $\Pr\big\{\bigcap_{i=1}^{4}\{\mathscr{Z}_i\}\big\}$ is maximized, which can be relatively easily handled.

Next, we provide a method for computing the probability $\Pr\big\{\bigcap_{i=1}^{4}\{\mathscr{Z}_i\}\big\}$. We first regard the components contained in B_L, B_B, B_R, and B_T as one individual component. As a result, a system obtained by eliminating the components in set C from a Lin/$(m-1, n-1)$/(m, n):F system can be reduced to a system with eight components, as shown in Fig. 4.9. The individual components are numbered as components $0, 1, \ldots, 7$ (see Fig. 4.9). For $l = 0, 1, \ldots, 7$, the reliability of component l, denoted by P_l, is given as follows:

$$\begin{aligned} P_0 &= p_{\pi(1,1)}, & P_1 &= 1 - \prod_{(i,j)\in B_L} \big(1 - p_{\pi(i,j)}\big), \\ P_2 &= p_{\pi(m,1)}, & P_3 &= 1 - \prod_{(i,j)\in B_B} \big(1 - p_{\pi(i,j)}\big), \\ P_4 &= p_{\pi(m,n)}, & P_5 &= 1 - \prod_{(i,j)\in B_R} \big(1 - p_{\pi(i,j)}\big), \\ P_6 &= p_{\pi(1,n)}, & P_7 &= 1 - \prod_{(i,j)\in B_T} \big(1 - p_{\pi(i,j)}\big). \end{aligned}$$

Moreover, we define the unreliability of components l by $Q_l = 1 - P_l$ for $l = 0, 1, \ldots, 7$.

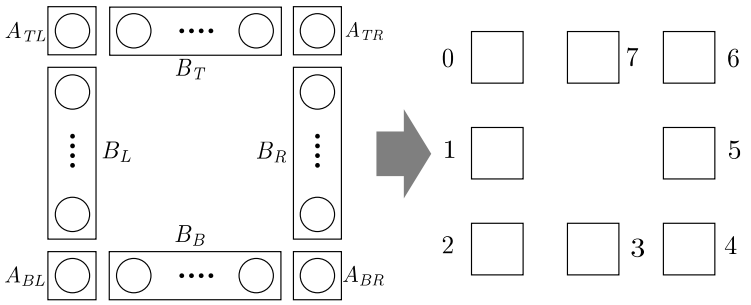


Figure 4.9: Circular sub-system.

Also, this system with the eight components is called "circular sub-system" in this chapter. Note that the circular sub-system has the minimal cuts {0,1,7}, {1,2,3}, {3,4,5}, and {5,6,7}.

We present a recursive method for efficiently computing the reliability of the circular sub-system. First, we employ the event decomposition approach [22] to compute the reliability. Applying this approach to the circular sub-system, we can decompose "the event that the circular sub-system works" disjointly into

(α) the event that the circular sub-system works in which component 0 works,

(β) the event that the circular sub-system works in which component 0 fails and component 1 works, and

(γ) the event that the circular sub-system works in which components 0 and 1 fail.

Because these events are disjoint, the summation of the probabilities of these events becomes the probability of the event that the circular sub-system works, that is, the reliability of the circular sub-system.

Next, we define the additional notation. For $\phi \in \{\alpha, \beta, \gamma\}$, $k \in \{0, 1, 2, 3\}$ and $l = 1, 2, 3, \ldots, 7$, $L^{(\phi)}(k, l)$ is defined as the probability that a circular sub-system consisting of components $0, 1, \ldots, l$ satisfies the following three conditions:

(a) the system is working.

(b) $\begin{cases} \text{component 0 works,} & \text{if } \phi = \alpha. \\ \text{component 0 fails and component 1 works,} & \text{if } \phi = \beta. \\ \text{components 0 and 1 fail,} & \text{if } \phi = \gamma. \end{cases}$

(c) components $l-k+1, l-k+2, \ldots, l$ fail. Note that if $k=0$, component l works.

Using the above notation, we can compute the reliability of the circular sub-system by the following theorem:

Theorem 4.1. (a) For $\phi \in \{\alpha, \beta, \gamma\}$ and $l = 2, 3, \ldots, 7$,

$$\begin{cases} L^{(\phi)}(0,l) = P_l \left(\displaystyle\sum_{k\in\{0,1,2,3\}} L^{(\phi)}(k, l-1) \right), \\ L^{(\phi)}(1,l) = (1-P_l) L^{(\phi)}(0, l-1), \\ L^{(\phi)}(2,l) = (1-P_l) L^{(\phi)}(1, l-1), \\ L^{(\phi)}(3,l) = \begin{cases} 0 & \text{if } l \in \{3,5,7\}, \\ (1-P_l) L^{(\phi)}(2, l-1) & \text{if } l \in \{2,4,6\}. \end{cases} \end{cases} \tag{4.25}$$

As the boundary condition, for $l = 1$,

$$L^{(\phi)}(k,1) = \begin{cases} P_0 P_1, & \text{if } \phi = \alpha \text{ and } k = 0, \\ P_0(1-P_1), & \text{if } \phi = \alpha \text{ and } k = 1, \\ (1-P_0)P_1, & \text{if } \phi = \beta \text{ and } k = 0, \\ (1-P_0)(1-P_1), & \text{if } \phi = \gamma \text{ and } k = 2, \\ 0, & \text{otherwise.} \end{cases} \tag{4.26}$$

(b) The reliability of the circular sub-system is given by

$$\Pr\left\{ \bigcap_{i=1}^{4} \{\mathscr{Z}_i\} \right\} = \sum_{k\in\{0,1,2\}, \phi\in\{\alpha,\beta\}} L^{(\phi)}(k,7) + L^{(\gamma)}(0,7). \tag{4.27}$$

Note that the minimal cut sets of the circular sub-system in the case of (γ) are $\{1,2,3\}, \{3,4,5\}, \{7\}$. Because the failure of component 7 leads to the system failure, we have

$$L^{(\gamma)}(1,7) = L^{(\gamma)}(2,7) = 0.$$

Recall that we assign the components in order of positions $(2,1), (3,1), \ldots, (m,1), (m,2), \ldots, (m,n), (m-1,n), \ldots, (1,n), (1,n-1), \ldots, (1,2)$. Let us consider two component arrangements in which we assign the same components

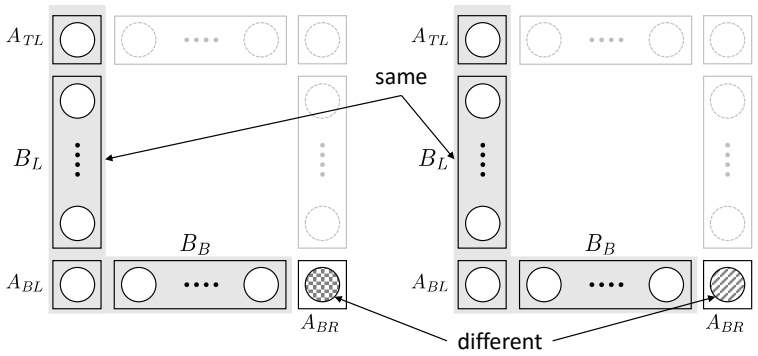


Figure 4.10: Idea of computing the system reliability effectively.

to A_{TL}, B_L, A_{BL}, B_B, the different component to A_{BR}, and no components to the other areas as shown in Fig. 4.10. When we compute the reliabilities of two circular sub-systems, areas $A_{\mathrm{TL}}, B_L, A_{\mathrm{BL}}, B_B$ are common in both arrangements. Accordingly, if we obtain the reliability of the system under the arrangement, then, by memorizing $L^{(\phi)}(k,3)$ ($\phi \in \{A,B,C\}$ and $k \in \{0,1,2,3\}$), we can easily obtain the reliability of the circular sub-system under the other arrangements. Therefore, the proposed algorithm can be accelerated with the help of Theorem 4.1. An example of this computation of the system reliability is given later.

Finally, we derive a condition for the pruning based on system reliability. It is evident from Theorem 4.1 that the system reliability is non-increasing in parameter l. To put it differently, the system reliability under a partial arrangement becomes an upper bound for the system reliability when the partial arrangement is fixed. Thus, we can eliminate unnecessary arrangements by computing the upper bound of the system reliability under a partial arrangement and then comparing it with the maximum system reliability obtained so far. Thus, we derive the following condition.

Condition 7. When a component is assigned to position (m,j) for $j = 1,2,\ldots,n$, if the following condition is not satisfied, then we perform the pruning.

$$\mathbf{Rmax} \leq \sum_{k\in\{0,1,2\},\phi\in\{A,B\}} L^{(\phi)}(k,l), \tag{4.28}$$

where **Rmax** is the maximum system reliability obtained so far.

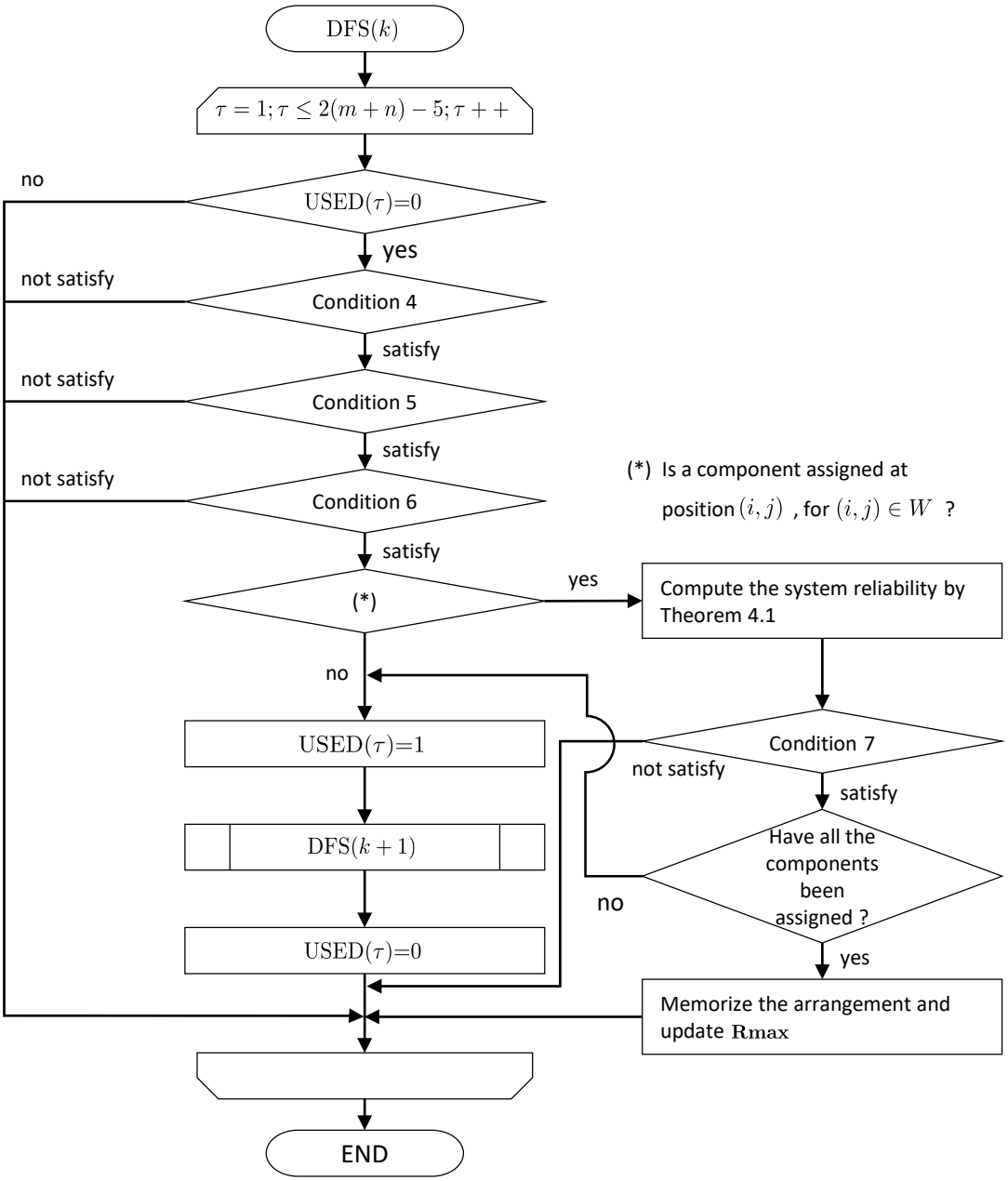


Figure 4.11: Flow chart of DFS(k).

4.3.2 Algorithm and Illustrative Example

In this subsection, we propose an algorithm based on the B&B method to search for finding the optimal arrangement of the Lin/$(m-1,n-1)$/(m,n):F system. The algorithm first begins by setting parameters: m, n, and $\boldsymbol{p}$, and then, we set **Rmax** to be 0. Next, from Proposition 4.1, the $(m-2)(n-2)$ most reliable components are assigned to area C as a preprocessing step. As a result, the Lin/$(m-1,n-1)$/(m,n):F system can be regarded as a circular sub-system. Also, the least reliable component (component 1) is assigned to position $(1,1)$ from Condition 1 (i). After that, we assign the other components to the other areas in order of positions $(2,1),(3,1),\ldots,(m,1),(m,2),\ldots,(m,n),(m-1,n),\ldots,(1,n),(1,n-1),\ldots,(1,2)$ (counter-clockwise) in the DFS tree.

Next, let USED(τ) denote an index function that takes 1 if component τ has been assigned; otherwise, 0 for $\tau = 1,2,\ldots,2(m+n)-5$. When we assign the components in order of positions $(2,1),(3,1),\ldots,(m,1),(m,2),\ldots,(m,n),(m-1,n),\ldots,(1,n),(1,n-1),\ldots,(1,2)$ in the DFS tree, DFS(k) is denoted by a routine to assign a component to the kth position for $k = 1,2,\ldots,2(m+n)-5$.

Figure 4.11 shows a flow chart of DFS(k). The procedure of DFS(k) is given in the following steps.

STEP 0: Set $\tau = 0$.

STEP 1: Update $\tau = \tau + 1$. If $\tau > 2(m+n)-5$, then stop DFS(k).

STEP 2: If component τ is used already, that is, USED(τ) $= 1$, then go back to **STEP 1**.

STEP 3: If Condition 4, 5, or 6 is not satisfied, then we perform the pruning, that is, go back to **STEP 1**.

STEP 4: If a component is assigned to position (i,j), for $(i,j) \in W$, then we compute the system reliability under a partial arrangement by Theorem 4.1; otherwise, go to **STEP 7**, where

$$W = \{(m-1,1),(m,1),(m,n-1),(m,n),(2,n),(1,n),(1,2)\}.$$

STEP 5: If Condition 7 is not satisfied, then we perform the pruning, that is, go back to **STEP 1**.

STEP 6: If all the components have been assigned, then memorize the arrangement and update **Rmax**, and then, then stop DFS(k); otherwise, go to **STEP 7**.

STEP 7: Go to DFS($k+1$).

After enumerating all arrangements, an arrangement with **Rmax** becomes the optimal arrangement.

We give an example to illustrate how to implement the proposed algorithm for finding the optimal arrangement to make the proposed algorithm more understandable. As an example, let us consider a Lin/$(2,3)$/$(3,4)$:F system similarly to Subsection 3.2. First of all, as a preprocessing step, components 11 and 12 are assigned to positions $(2,3)$ and $(2,4)$ according to Proposition 4.1, and component 1 is assigned to position $(1,1)$ according to Condition 1 (i). A tree for enumerating arrangements of this system is shown in Fig. 4.12. Note that a black cell means a position where a component has already been assigned, and a number in a cell means the index of components. In the following, for the sake of conciseness, we explain the procedure for each layer, although the actual algorithm uses the DFS strategy.

In the first layer (DFS(1)), if components $2, 3, \ldots, 10$ is assigned to position $(2,1)$, we compute the reliability of the system with the partial arrangements by Theorem 4.1, and then go to DFS(2). Here, we focus on the case where component 6 is assigned. In the second layer (DFS(2)), components $2, 3, 4, 5, 7 \ldots, 10$ are assigned to position $(3,1)$. The first and second arrangements from the right do not satisfy Condition 5 (i) ($\pi(2,1) < \pi(3,1)$), and thus, go back to DFS(1). For the other arrangements, we compute the reliability of the system with the partial arrangements by Theorem 4.1. Here, we focus on the case where component 3 is assigned. In the third layer (DFS(3)), components $2, 4, 5, \ldots, 10$ are assigned to position $(3,2)$. The first arrangement from the left does not satisfy Condition 4 (ii) ($\pi(3,1) < \pi(3,2)$), and thus, go back to DFS(2). In the fifth layer (DFS(5)), components $2, 4, 5, 9, 10$ are assigned to position $(3,4)$. The first and second arrangements from the right do not satisfy Condition 5 (ii) ($\pi(3,2) < \pi(3,4)$), and thus, go back to DFS(4). For the other arrangements, we compute the reliability of the system with the partial arrangements by Theorem 4.1.

In the ninth layer (DFS(9)), we compute the system reliability for each arrangement. If the system reliability with an arrangement is higher than **Rmax**, we memorize the

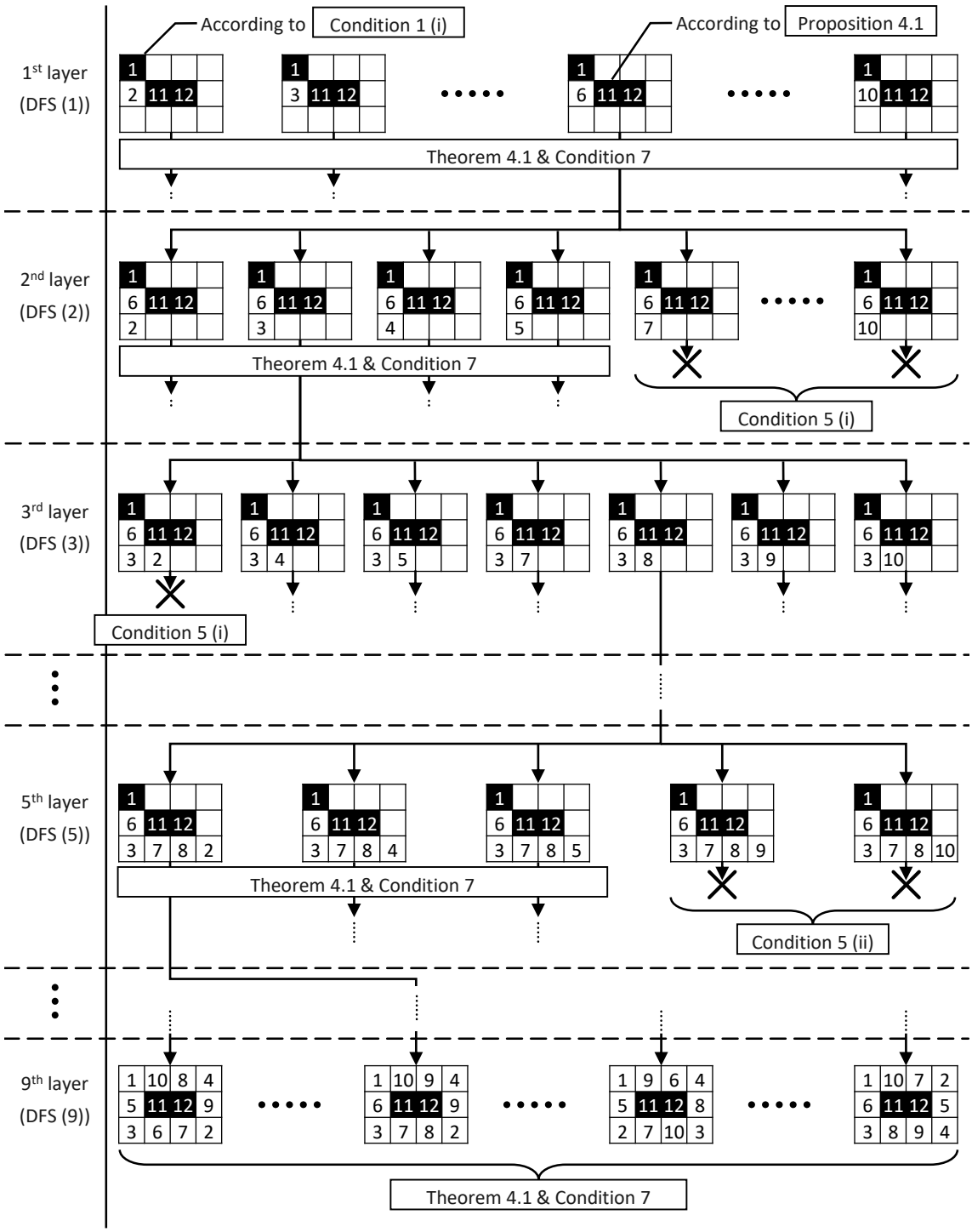


Figure 4.12: Tree for enumerating arrangements of the Lin/$(2,3)$/$(3,4)$:F system.

arrangement and update **Rmax**. If all the arrangements have already been enumerated, eventually, the arrangement with **Rmax** becomes optimal, and otherwise, go back to DFS(8).

4.4 Efficiency Investigation

In this section, we investigate the efficiency of the proposed algorithms for finding the optimal arrangement of Lin/(r,s)/(m,n):F systems. Computation time plays an important role in the measure of the computational efficiency of different algorithms, and thus, we compare the algorithms in terms of the computation time necessary to find the optimal arrangement. All algorithms were programmed in Microsoft Visual C 2017 and were implemented on a computer with Intel Core i7 2.4 GHz CPU, 16.0 GB memory, Microsoft Windows 8 OS.

As mentioned before, Omura *et al.* [93] developed an algorithm for the CAP for Lin/(r,s)/(m,n):F systems. Here, we explain the algorithm developed by Omura *et al.* [93], which is compared with the proposed algorithms to investigate their efficiency. To find the optimal arrangement, we assign the components in order of positions $(2,1),(3,1),\ldots,(m,1),(m,2),\ldots,(m,n),(m-1,n),\ldots,(1,n),(1,n-1),\ldots,(1,2)$ in the DFS tree. Finally, we compute the system reliability after all the components are assigned by the recursive equation [58]. We can find the optimal arrangement after enumerating all the arrangements. Although this algorithm is specialized for the case of $r=m-1$ and $2s>n$, it can be extended to be applied to general Lin/(r,s)/(m,n):F systems with some modifications. The algorithm proposed in Section 4.2 that does not incorporate Condition 2, preprocessing step based on Lemma 4.1, "eliminating redundant computations" and Condition 3 becomes the algorithm of Omura *et al.* [93].

In this section, three numerical experiments, Ex 1, Ex 2, and Ex 3, are carried out to aid in investigating the efficiency of the proposed algorithms. In Ex 1, we compare the efficiency of the proposed algorithms with that of the existing algorithm [93]. In Ex 2, we investigate the efficiency of the preprocessing step based on Lemma 4.1, Conditions 1 and 2 in the algorithm in Section 4.2. In Ex 3, we investigate the efficiency of "eliminating redundant computations" and "Condition 3" in the algorithm in Section 4.2.

First, we compare the efficiency of the algorithm proposed in Section 4.2 (PA (a)), the algorithm proposed in Section 4.3 (PA (b)) the existing algorithm [93] (EA). Ta-

1	**9**	12	**13**	3
5	15	16	17	**6**
8	18	19	20	**7**
2	10	11	14	4

PA (a)

1	**13**	12	**9**	3
5	15	16	17	**7**
8	18	19	20	**6**
2	10	11	14	4

PA (b)

1	**9**	12	**13**	3
5	15	16	17	**6**
8	18	19	20	**7**
2	10	11	14	4

EA

Figure 4.13: Optimal arrangements of Lin/$(3,4)$/$(4,5)$:F system when the component reliabilities are generated from a uniform distribution on $[0.10, 0.50]$.

1	7	9	13	15	16	18	2
6	19	20	21	22	23	24	5
3	8	10	11	12	14	17	4

(a) when component reliability are generated from a uniform distribution on $[0.25, 0.35]$.

1	8	9	10	14	16	17	2
6	19	20	21	22	23	24	5
3	7	11	12	13	15	18	4

(b) when component reliability are generated from a uniform distribution on $[0.10, 0.50]$.

Figure 4.14: Optimal arrangements of the Lin/$(2,7)$/$(3,8)$:F systems when the component reliabilities are generated from a uniform distribution on $[0.25, 0.35]$ and $[0.10, 0.50]$.

ble 4.1 shows the computation time required for each algorithm to find the optimal arrangement when the component reliabilities are generated from a uniform distribution on, for example, $[0.10, 0.50]$ and $[0.10, 0.30]$, and Fig. 4.13 shows the optimal arrangements of the Lin/$(3,4)$/$(4,5)$:F system. As can be seen from Fig. 4.13, PA (a) and EA provided the same optimal arrangement. Meanwhile, PA (a) and PA (b) provided different optimal arrangements, where the values denoted in boldface in Fig. 4.13 mean the difference between PA (a) (EA) and PA (b). However, even if we exchange components $\pi(1,2)$ and $\pi(1,4)$, the system reliability does not change, which is proved immediately from Lemma 4.2. Hence, the reliabilities of the optimal arrangements obtained by PA (a) (EA) and PA (b) are the same, and thus, these optimal arrangements are essentially the same. For the other systems, we confirmed that each algorithm provided the same optimal arrangements in the numerical experiment. When compar-

ing PA (a) and EA, we found from Table 4.1 that PA (a) was more efficient than EA because its computation times were shorter than those of EA. For example, for the Lin/$(4,4)$/$(5,5)$:F system, PA (a) took 40.839 seconds, whereas EA could not find the optimal arrangement within the time limit of one day (86,400 seconds). If $r \neq m-1$ or $s \neq n-1$, then PA (b) is inapplicable for finding the optimal arrangement. However, if a Lin/(r,s)/(m,n):F system satisfies $r = m-1$ and $s = n-1$, PA (b) outperforms PA (a) in terms of the computation time. Accordingly, PA (b), which depends on the special structure of the system, can make a considerable reduction in computational efforts for finding the optimal arrangement. The main reason for PA (b) to be efficient is that the method in Theorem 4.1 can compute the system reliability efficiently. In summary, compared with EA, we conclude that PA (a) and PA (b) are more practical when considering relatively large systems.

Second, we investigate the efficiency of the preprocessing step based on Proposition 4.1, Conditions 1 and 2 in PA (a). Table 4.2 gives the results of the numerical experiment when the component reliabilities are generated from a uniform distribution on $[0.10, 0.50]$. Recall that if $2r > m$ and $2s > n$, Proposition 4.1 can reduce the number of the components that we should assign. Table 4.2 shows that the computation times are constant for the case of $2r \leq m$ or $2s \leq n$ even if we do not apply the preprocessing step. However, for the case of $2r > m$ and $2s > n$, as $m - r$ and $n - s$ are small, the preprocessing step reduces the number of the components that we should assign, and as a result, the computation times also decrease. Moreover, it is obvious from Table 4.2 that Condition 1 reduces the computation time because we can reduce the number of candidates for the optimal arrangement by the pruning according to Condition 1. As can be seen from Table 4.2, we found that Condition 2 also reduced the computation time. We experimentally demonstrated that the preprocessing step based on Proposition 4.1, Conditions 1 and 2 reduced the number of enumerated arrangements, and consequently, the performance of the proposed algorithm was significantly enhanced.

Finally, we investigate the efficiency of "eliminating redundant computations" and "Condition 3." Table 4.3 shows the required computation time and the number of enumerated arrangements for finding the optimal arrangement. The symbol (a) in the table means the proposed algorithm, which eliminates redundant computations by memorizing some values and performs the pruning according to Condition 3; the symbol (b) means an algorithm in which "the system reliability is computed after all the components are assigned" and "Condition 3 is not used." It is obvious from

Table 4.3 that the computation times of (a) are shorter than those of (b). From the result, we found that memorizing the specific values and utilizing them were useful in terms of computing the system reliability efficiently. Table 4.3 indicated that when the component reliabilities are generated from a uniform distribution on $[0.25, 0.35]$, (a) enumerated the same number of arrangements as (a), and thus, we found that the pruning according to Condition 3 was not performed. On the other hand, we can observe that when the component reliabilities are generated from a uniform distribution on $[0.10, 0.50]$, the number of arrangements that (b) enumerates is small compared with the case of $[0.25, 0.35]$, which means that the pruning according to Condition 3 were performed. From the results, we found that the components with different reliabilities affected the performance of the algorithm. Specifically, as the interval of component reliabilities becomes larger, the number of enumerated arrangements decreases because the number of pruning according to Condition 3 increases. Note that the interval of component reliabilities generated from a uniform distribution on $[0.10, 0.50]$ is larger than the interval of component reliabilities generated from a uniform distribution on $[0.25, 0.35]$. In conclusion, we see that memorizing the specific values and utilizing them enhance the performance of the algorithm, and the performance depends on the component reliabilities within the scope of the experiment.

Also, Fig. 4.14 shows the two optimal arrangements of Lin/$(2, 7)$/$(3, 8)$:F systems with the component reliabilities generated from uniform distributions on $[0.25, 0.35]$ and $[0.10, 0.50]$. From Fig. 4.14, it was found that these optimal arrangements are different. Consequently, we confirmed that the optimal arrangement depended on the values of the component reliabilities, that is, the variant optimal arrangement.

In summary, within the scope of the experiment, we obtained the following findings. If $r = m - 1$ and $s = n - 1$, we suggest using the algorithm proposed in Section 4.3 to efficiently find the optimal arrangement. In contrast, if $r \neq m - 1$ or $s \neq n - 1$, we suggest using the algorithm proposed in Section 4.2, which is more efficient than the existing algorithm [93].

Table 4.1: Comparison of the proposed algorithms and the existing algorithm (sec.).

(r, s)	$(2,2)$	$(3,2)$	$(3,3)$	$(3,3)$	$(3,4)$	$(3,3)$	$(4,4)$	$(4,5)$	$(5,4)$	$(5,5)$
(m, n)	$(4,4)$	$(4,4)$	$(4,4)$	$(4,5)$	$(4,5)$	$(5,4)$	$(5,5)$	$(5,6)$	$(6,5)$	$(6,6)$
Component reliability	$[0.1, 0.5]$	$[0.1, 0.5]$	$[0.1, 0.5]$	$[0.1, 0.5]$	$[0.1, 0.5]$	$[0.1, 0.5]$	$[0.1, 0.5]$	$[0.1, 0.3]$	$[0.1, 0.3]$	$[0.1, 0.3]$
PA (a)	408.814	52.841	0.081	3686.043	2.302	4548.243	40.839	1144.770	1532.192	19645.676
PA (b)	—	—	0.008	—	0.199	—	1.874	74.520	42.371	598.857
EA	462.768	72.450	7.234	19529.636	6066.075	41857.860	N/A	N/A	N/A	N/A
System reliability	0.328	0.782	0.960	0.943	0.989	0.943	0.998	0.992	0.992	0.997

Table 4.2: Efficiency of preprocessing step based on Proposition 4.1, Conditions 1 and 2 in the proposed algorithm (sec.).

Preprocessing step	Condition 1	Condition 2	$(2,2)$	$(3,2)$	$(3,3)$	$(3,3)$	$(3,4)$	$(4,4)$	(r, s)
			$(4,4)$	$(4,4)$	$(4,4)$	$(4,5)$	$(4,5)$	$(5,5)$	(m, n)
applied	applied	applied	408.814	52.841	0.081	3686.043	2.302	40.839	
non-applied	applied	applied	408.883	52.806	4.383	16542.187	1633.039	N/A	
applied	non-applied	applied	N/A	N/A	8.64	N/A	N/A	N/A	
applied	applied	non-applied	1642.952	160.231	0.208	11096.041	5.733	97.071	
System reliability			0.329	0.782	0.960	0.943	0.989	0.998	

Table 4.3: Computation time and the number of enumerated arrangements for finding the optimal arrangement in the case of $(r, s) = (2, n-1)$ (sec.).

Component reliability		$[0.25, 0.35]$				$[0.10, 0.50]$			
(m, n)		$(3, 7)$	$(3, 8)$	$(3, 9)$	$(3, 10)$	$(3, 7)$	$(3, 8)$	$(3, 9)$	$(3, 10)$
(a)	# of enumerated arrangements	3863574	26223691	165640384	990936868	2896874	17731266	99299442	650309311
	Computation time	11.063	129.857	1411.915	15510.505	9.973	118.007	1293.085	14749.091
(b)	# of enumerated arrangements	3863574	26223691	165640384	990936868	3863574	26223691	165640384	990936868
	Computation time	27.584	290.891	2214.942	18507.930	27.777	290.713	2206.873	18499.429
System reliability		0.970	0.985	0.993	0.996	0.990	0.996	0.998	0.999

4.5 Summary

This chapter addressed the CAP in a Lin/(r, s)/(m, n):F system. The main contribution of this chapter is to develop two algorithms for efficiently finding the optimal arrangement of a Lin/(r, s)/(m, n):F system. One algorithm is for a general Lin/(r, s)/(m, n):F system. It is noteworthy that we developed the algorithm available for the general case for the first time. The other algorithm specializes to the case of $r = m-1$ and $s = n-1$. The main principle of these algorithms is the incorporation of the conditions for pruning in the standard B&B framework to reduce the number of candidates for the optimal arrangement. Besides, memorizing the specific values and utilizing them can enhance the performance of the algorithm, which eliminates redundant computations. The algorithm that specializes to the case of $r = m-1$ and $s = n-1$ was developed by the idea that the CAP of a Lin/$(m-1, n-1)$/(m, n):F system can be simplified to a problem of finding the optimal arrangement of a simple system. To investigate the efficiency of the proposed algorithm, we performed a numerical experiment. The result showed that the proposed algorithms were more efficient than the existing algorithm [93] in terms of computation time. Also, we provided the guideline for appropriately selecting which algorithm should be used. Specifically, if $r = m-1$ and $s = n-1$, we suggest using the algorithm proposed in Section 4.3 to efficiently find the optimal arrangement, whereas if $r \neq m-1$ or $s \neq n-1$, we suggest using the algorithm proposed in Section 4.2.

Chapter 5

Conclusions

This chapter summarizes the main contributions of this thesis and describes various future perspectives.

5.1 Summary of the Thesis

This thesis targeted the connected-**X**-out-of-(m, n):F lattice system, and we provided the methods and algorithms for evaluating and increasing system reliability. The contributions of this thesis are summarized as follows.

Chapter 2 focused on the system reliability evaluation, that is, to compute the system reliability when the component reliabilities are given. The system reliability evaluation is a fundamental step in all reliability studies. Most researches have been devoted to the studies of the linear-type systems, whereas no study dealt with the toroidal-type systems. Hence, we considered the computation of the reliability of the Tor/(r, s)/(m, n):F system. First, we provided a recursive equation approach, and, for efficiently computing the system reliability, we proposed two kinds of algorithms. One algorithm was incorporated with the idea that the elements of a set are enumerated beforehand and stored to increase efficiency. It was theoretically shown that the algorithm with the above idea required extra memory space but had better time complexity compared to the other one. Also, the numerical experiments have shown the effectiveness and efficiency of the algorithm with the idea. Furthermore, we obtained the B-importance for the Tor/(r, s)/(m, n):F system by the proposed algorithm as a numerical example. Although this thesis dealt with the Tor/(r, s)/(m, n):F system, the

research on this system is quite limited, and other problems for this system have not been discussed. Therefore, the further study for the Tor/(r,s)/(m,n):F system would be of value to the research field of reliability, e.g., solving the CAP and establishing the stochastic comparison.

Two types of methods are common and popular for efficiently computing the reliability of the consecutive-k systems: the recursive equation approach and the FMCIA. However, there were no studies comparing the efficiencies of the recursive equation approach and the FMCIA. Hence, this thesis proposed both the recursive equation approach and the FMCIA for computing the reliability of Lin, Cir, and Tor/$(1,2)$-or-$(2,1)$/(m,n):F systems. Note that the recursive equation approach for a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system has already been proposed by Yamamoto *et al.* [64]. One of the contributions of Chapter 2 was to describe them in a unified manner. Then, we compared the efficiency of the recursive equation approach and the FMCIA, which is the first-ever attempt to compare both methods. In conclusion, we found that their efficiency depended on the situation. Specifically, for a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system, in the INID case, the recursive equation approach was more efficient. In the IID case, if the number of columns was large, the FMCIA outperformed the recursive equation approach. On the other hand, for a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system, within the scope of the experiment, we can conclude that the FMCIA outperforms the recursive equation approach. This result enables us to select which exact method should be used for efficiently computing the system reliability. A further study of a comprehensive comparison of the efficiency of both methods for other consecutive-k systems should be conducted.

As the size of a system becomes large, it is cumbersome to obtain the exact system reliability. Accordingly, it would be beneficial to use appropriate upper and lower bounds if we do not need to obtain the exact system reliability. Hence, we derived the upper and lower bounds for the reliability of a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system based on the idea of using the upper and lower bounds of the difference. Moreover, we conducted numerical experiments to evaluate the derived bounds. From the results, within the scope of the experiment, it was found that the derived bounds were tighter than the best of the existing bounds although they required more computation time. Therefore, we concluded that we obtained the tighter bounds at the expense of the computational effort compared with the existing bounds.

Chapter 3 dealt with the system signature of a Lin/(r,s)/(m,n):F system and a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system. The system signature is connected to many other

well-known reliability concepts and has various theoretical applications. For example, the system signature is used to establish stochastic comparisons among some systems. Also, the reliability polynomial can be easily expressed in terms of the system signature, which means that once a system signature is obtained, various reliability indices are easy to derive. Despite the fact that the system signatures of consecutive-k-out-of-n:F systems have been studied, no study discussed the system signature of a connected-$\mathbf{X}$-out-of-(m,n):F lattice system. Thus, we developed the methods for computing the system signatures of a Lin/(r,s)/(m,n):F system and a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system. The numerical experiments have shown the efficiency of the proposed algorithms. Therefore, the proposed algorithms can be recommended for computing the system signatures of a Lin/(r,s)/(m,n):F system and a Lin/$(1,2)$-or-$(2,1)$/(m,n):F system. Also, we presented the stochastic orderings of some connected-$\mathbf{X}$-out-of-(m,n):F lattice systems, and the orderings could determine the system that works properly for a longer time. The comparison results can find the optimal parameters, which would be helpful for designing a system with high reliability. Therefore, the utility of the system signature became evident in the context of the comparisons of competing systems.

Chapter 4 addressed the CAP in a Lin/(r,s)/(m,n):F system. Methods for solving the CAP can be broadly classified into two approaches: the exact method and the approximate method (heuristic and meta-heuristic methods). Basically, the advantage of approximate methods is that they usually provide good solutions in a relatively short time, but the drawbacks are that they cannot guarantee global optimality. In contrast, exact methods may be particularly advantageous when the system is not large. More importantly, such methods can be used to measure the performance of approximate methods. Thus, this thesis focused on getting the optimal arrangement by the exact method. For a Lin/(r,s)/(m,n):F system, the existing studies developed the two algorithms for finding the optimal arrangement, but these methods are limited to the special cases. Hence, this thesis developed an algorithm for finding the optimal arrangement of a general Lin/(r,s)/(m,n):F system. The main principle of this algorithm is the incorporation of the conditions for pruning in the standard B&B framework to reduce the number of candidates for the optimal arrangement. Besides, memorizing the specific values and utilizing them can enhance the performance of the algorithm. It is noteworthy that we developed the algorithm available for the general case for the first time. In addition, we also proposed an algorithm that specializes to the case of $r = m - 1$ and $s = n - 1$. This algorithm is incorporated with the idea that the CAP

of a Lin/$(m-1, n-1)$/(m, n):F system can be simplified to a problem of finding the optimal arrangement of a simple system. To investigate the efficiency of the proposed algorithms, we performed a numerical experiment. Also, we provided the guideline for appropriately selecting which algorithm should be used.

In summary, this thesis provided the methods and the algorithms for the reliability evaluation and the optimal design of connected-**X**-out-of-(m, n):F lattice systems. These methods and algorithms can be used in "computing the reliability of a system from the component reliabilities" and "optimizing reliability in the design phase" and will be useful for appropriately evaluating and enhancing the reliability of practical systems that can be expressed as connected-**X**-out-of-(m, n):F lattice systems. We expect that they lead to designing practical systems that satisfy the performance requirements desired by customers. In addition, as mentioned in Chapter 1, the connected-**X**-out-of-(m, n):F lattice system may be an oversimplification in some cases. However, this system can be regarded as a special case of a general system. Thus, this study would also be a clue for giving methods for appropriately evaluating and enhancing the system reliability.

5.2 Future Work

This section describes various interesting topics for possible future developments and research.

(A) Upper and lower bounds for the reliability of various connected-X-out-of-(m, n):F lattice systems This thesis proposed the upper and lower bounds for the system reliability, which leads to a notable reduction in the computational cost and results in very tight bounds. The main idea of the bounds is, as mentioned before, to employ the upper and lower bounds of the difference. Further studies will apply the idea used in this thesis to various connected-**X**-out-of-(m, n):F lattice systems in order to derive upper and lower bounds. Furthermore, this idea would be used to derive bounds for the reliability of coherent systems.

(B) Stochastic comparison of systems with multiple types of components This thesis considered the system signature of a connected-**X**-out-of-(m, n):F lattice system in the IID case. In other words, the components are regarded as single types

of components. However, in a practical situation, a system may consist of multiple types of components. Coolen and Coolen-Maturi [127] extended the original system signature to a survival signature, which is an appropriate tool in studying systems with multiple types of components. Also, Eryilmaz [128] discussed the reliability of systems with multiple types of components. The study for a connected-**X**-out-of-(m, n):F lattice system with more than two types of components would be of value in the research field of reliability. The methods proposed in this thesis could be extended to such systems, and then, we will propose a method for efficiently computing the signature of such systems. Then, the obtained signature can establish the stochastic comparison, which makes adjustment of the system's parameters more flexible.

(C) CAP for various consecutive-k systems This thesis addressed the CAP for a Lin/(r, s)/(m, n):F system. Because a practical engineering system is expected to be reliable and work satisfactorily for a long time, finding an optimal arrangement is very important to design new systems economically. Thus, studies of developing an efficient and systematic algorithm for the optimal arrangement is of vital importance. There have been few reports about the CAP for a connected-**X**-out-of-(m, n):F lattice system except for a Lin/(r, s)/(m, n):F system. In the future, we plan to develop efficient algorithms for efficiently solving the CAP for other connected-**X**-out-of-(m, n):F lattice system, which are based on the algorithm proposed in this thesis. For example, we will develop an algorithm for a Lin/$(1, 2)$-or-$(2, 1)$/(m, n):F system if necessary conditions that the optimal arrangement must satisfy can be derived. The research aimed at developing such efficient algorithms would be quite worthwhile.

(D) Meta-heuristic methods for solving the CAP This thesis proposed the algorithms for the CAP. A possible limitation of the algorithms is that it may not be efficient enough for large systems, which is a common limitation for many exact methods. Often, approximate methods can obtain near-optimal solutions at a relatively low computational cost. As part of future work plans, we aim to develop meta-heuristic methods, such as genetic algorithm, simulated annealing, and ant colony optimization, for solving the CAP. The key idea of the algorithms proposed in this thesis is pruning a branch of a DFS tree and efficient computation of the system reliability. Some of the successful ideas in the exact methods can be incorporated into meta-heuristic methods. As a result, we expect that a high-performance meta-heuristic method using the problem

structure will be developed. In recent years, B-importance-based methods have received great attention, which is explained in Chapter 1. The previous studies employed the B-importance to guide for assigning components. To enhance the efficacy and efficiency, we may incorporate the B-importance into a meta-heuristic method.

(E) Maintenance problem Another problem in the reliability area is the maintenance problem. Maintenance involves preventive (planned) and corrective (unplanned) actions carried out to retain a system in or restore it to an acceptable operating condition, and optimal maintenance policies aim to provide optimum system reliability and safety performance at the lowest possible maintenance costs [129]. Frequent maintenance will achieve a stable operation of a system, whereas too frequently maintenance actions may cause high maintenance costs. Hence, the essential thing in the maintenance problem is determining how and when to perform the maintenance. Readers are referred to Yun and Endharta [15] for the details of the maintenance problem for consecutive-k systems. The maintenance problem is also needed to be discussed in more detail for connected-**X**-out-of-(m, n):F lattice systems. In the maintenance problem, the system signature and the survival signature may help to derive the maintenance policy [130–132].

(F) Problem of combining the CAP and the maintenance problem Most studies have been discussed the CAP and the maintenance problem, separately. However, combining the advantages of the CAP and the maintenance problem may improve system reliability cost-effectively, and thus, we would be able to obtain better optimization results. A few papers have attempted to study such a problem [18]. Motivated by the above fact, we will address the problem of combining the CAP and the maintenance problem, which will be is an important topic for later research.

There are still several promising directions worthy of future research.

Throughout this thesis, the components were assumed to be independent. Although this assumption simplifies the computations, it is inevitable for components to be dependent on each other because they are influenced by common production and operating environment [25]. Accordingly, it remains a challenge for future research to discuss connected-**X**-out-of-(m, n):F lattice systems with dependent components. For example, the dependence might be modeled by copula functions, which are powerful tools

for modeling dependence.

In many previous studies, both systems and components can only take two possible states: the perfectly working to the complete failure. However, many practical systems may take many other performance levels because of complicated system structures. Such systems can be modeled by multi-state systems in which systems and/or components can have more than two states: from the perfect working state to the complete failure state. Because the complexity of practical engineering systems has increased, this seems to be a very fruitful area for future research.

Finally, I would like to bridge the gap between theory and practice. In the area of consecutive-k systems, most papers discussed the theoretical study, and the contributions of this thesis are also mainly theoretical. As already mentioned in Chapter 1, a connected-**X**-out-of-(m, n):F lattice system can be used, for example, to evaluate a wireless sensor network, which is essential in the IoT society. Hence, reliability optimization enables us to design a reliable wireless sensor network. Therefore, studies for this system would be very valuable to fill the gap. I hope that the theoretical results will provide useful insights into industrial manufacturing and contributes to the stable operation of practical systems.

Appendix

Proof of Theorem 2.1

Proof. First, from the definition of $A(i,l;h_l;g_l)$ in Eq. (2.2), we can get

$$A(i,l;h_l;g_l)=\begin{cases} A(i-1,l;h_l;g_l-1)\cap\{Y_{il}(1)=1\}, & \text{if } 2\le i\le h_l \text{ and } g_l=i,\\ A(i-1,l;h_l;i-1)\cap\{Y_{il}(1)=0\}, & \text{if } i=h_l+1 \text{ and } g_l=0,\\ A(i-1,l;h_l;g_l-1)\cap\{Y_{il}(1)=1\}, & \text{if } i\ge h_l+2 \text{ and } g_l>0,\\ \left\{\bigcup_{v=0}^{r-1}A(i-1,l;h_l;v)\right\}\cap\{Y_{il}(1)=0\}, & \text{if } i\ge h_l+2 \text{ and } g_l=0,\\ \emptyset, & \text{otherwise,}\end{cases} \tag{A.1}$$

for $i=1,2,\ldots,m$ and $l=1,2,\ldots,n$. Also, from Eq. (2.3), the reliability of a Cir/(r,s)/(i,n):F system in which the states of the component on both ends are given by $\boldsymbol{h}$ and $\boldsymbol{g}$ can be rewritten as follows:

$$\begin{aligned} R^C(m;\boldsymbol{h};\boldsymbol{g}) &= \Pr\left\{\bigcap_{l=1}^{n}A(i,l;h_l;g_l)\right\},\\ &= \Pr\left\{\bigcap_{l=1}^{n}\left\{\bigcup_{g_l'\in L(h_l,g_l)}A(i-1,l;h_l;g_l')\right\}\bigcap\left\{\bigcap_{l=1}^{n}\{Y_{il}(1)=d_l\}\right\}\right\}, \end{aligned} \tag{A.2}$$

where

$$L(h_l,g_l)=\begin{cases}\{g_l-1\}, & \text{if } (2\le i\le h_l \text{ and } g_l=i) \text{ or } (i\ge h_l+2 \text{ and } g_l>0),\\ \{h_l\}, & \text{if } i=h_l+1 \text{ and } g_l=0,\\ \{0,1,\ldots,r-1\}, & \text{if } i\ge h_l+2 \text{ and } g_l=0,\\ \emptyset, & \text{otherwise,}\end{cases}$$

and for $l = 1, 2, \ldots, n$,

$$d_l = \begin{cases} 0, & \text{if } g_l = 0, \\ 1, & \text{if } g_l > 0. \end{cases}$$

Substituting Eq. (A.1) into Eq. (A.2) yields

$$\begin{aligned} R^C(m; \boldsymbol{h}; \boldsymbol{g}) &= \Pr\left\{ \bigcup_{\boldsymbol{g}' \in \Theta(i;\boldsymbol{h};\boldsymbol{g})} \left\{ \bigcap_{l=1}^{n} A(i-1, l; h_l; g'_l) \right\} \bigcap \left\{ \bigcap_{l=1}^{n} \{Y_{il}(1) = d_l\} \right\} \right\}, \\ &= \Pr\left\{ \bigcap_{l=1}^{n} \{Y_{il}(1) = d_l\} \right\} \times \sum_{\boldsymbol{g}' \in \Theta(i;\boldsymbol{h};\boldsymbol{g})} \Pr\left\{ \bigcap_{l=1}^{n} A(i-1, l; h_l; g'_l) \right\}, \\ &= F_i(\boldsymbol{d}) \sum_{\boldsymbol{g}' \in \Theta(i;\boldsymbol{h};\boldsymbol{g})} R^C(i-1; \boldsymbol{h}; \boldsymbol{g}'). \end{aligned} \tag{A.3}$$

where $\Theta(i; \boldsymbol{h}; \boldsymbol{g})$ is given in Eq. (2.15). Thus, Eq. (2.14) holds.

Also, the reliability of the Tor$/(r, s)/(m, n)$:F system can be obtained by summing up $R^C(m; \boldsymbol{h}; \boldsymbol{g})$s for $\boldsymbol{h} \in S \setminus C_h$ and $\boldsymbol{g} \in T(\boldsymbol{h}) \setminus C_g$ because the events that the Cir$/(r, s)/(m, n)$:F systems work are disjoint. Thus, Eq. (2.18) holds, and the proof of Theorem 2.1 is completed. □

Proof of Theorem 2.4

Proof. Let us first consider the case of $j \geq 2$ and $\boldsymbol{x}, \boldsymbol{y} \in W^L$, and we will denote the event

$$K(j) = \left\{ \prod_{i=1}^{m} \prod_{k=2}^{j} \left(1 - Y_{ik}^{(1,2)}\right) \times \prod_{i=2}^{m} \prod_{k=1}^{j} \left(1 - Y_{ij}^{(2,1)}\right) = 1 \right\},$$

for simplicity. From Eq. (2.34), we have

$$\begin{aligned} R^L(j; \boldsymbol{x}; \boldsymbol{y}) &= \Pr\left\{ K(j) \bigcap \left\{ \bigcap_{i=1}^{m} \{Z_{i,1} = x_i\} \right\} \bigcap \left\{ \bigcap_{i=1}^{m} \{Z_{ik} = y_i\} \right\} \right\}, \\ &= \Pr\left\{ \bigcup_{\boldsymbol{y}' \in \Omega^L(\boldsymbol{y})} K(j-1) \bigcap \left\{ \bigcap_{i=1}^{m} \{Z_{i,1} = x_i\} \right\} \bigcap \left\{ \bigcap_{i=1}^{m} \{Z_{i,j-1} = y_i\} \right\} \right. \\ &\qquad \left. \bigcap \left\{ \bigcap_{i=1}^{m} \{Z_{ij} = y_i\} \right\} \right\}. \end{aligned} \tag{A.4}$$

Since the states of the $2m$ components in the $(j-1)$th and jth columns are given by $\boldsymbol{y}'$ and $\boldsymbol{y}$, respectively, if $\boldsymbol{y}' \in \Omega^L(\boldsymbol{y})$, there exists no failure pattern $(1,2)$ in the $(j-1)$th and jth columns. Substituting Eqs. (2.34) and (2.22) into Eq. (A.4) yields

$$R^L(j;\boldsymbol{x};\boldsymbol{y}) = R^L(j-1;\boldsymbol{x};\boldsymbol{y}') \times G_j(\boldsymbol{y}). \tag{A.5}$$

In the case of $j = 1$ and $\boldsymbol{x}, \boldsymbol{y} \in W^L$, we have

$$R^L(1;\boldsymbol{x};\boldsymbol{y}) = G_1(\boldsymbol{y}), \tag{A.6}$$

as the boundary condition.

In the case of $\boldsymbol{x} \notin W^L$ or $\boldsymbol{y} \notin W^L$, there exist two or more consecutive failed components, and thus, it follows that

$$R^L(j;\boldsymbol{x};\boldsymbol{y}) = 0, \tag{A.7}$$

from the definition of $R^L(j;\boldsymbol{x};\boldsymbol{y})$ in Eq. (2.34). From Eqs. (A.4), (A.5), and (A.6), it follows that Eq. (2.35).

By applying the event decomposition approach [22], the reliability of the Cir/$(1,2)$-or-$(2,1)$/(m,n):F system can be obtained by summing up $R^L(n;\boldsymbol{x};\boldsymbol{y})$s for $\boldsymbol{x} \in W^L$ and $\boldsymbol{y} \in \Omega^L(\boldsymbol{x})$. Thus, we can get Eq. (2.36), and this completes the proof. □

Proof of Theorem 2.9

Proof. The main idea of the proposed approach is that we make a cut in a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system between the 1st circle and the mth circle and subsequently bond the 1st circle and the mth circle. Consequently, the reliability of a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system can be obtained. Here, in order to bond the 1st circle and the mth circle later, we virtually add components $(0,1),(0,2),\ldots,(0,n)$ as dummy components in 0th circle.

First, we define $C(m;\boldsymbol{h};\boldsymbol{g})$ to be the conditional probability that a Cir/$(1,2)$-or-$(2,1)$/(m,n):F system in which the states of the n components in the mth circle are given by $\boldsymbol{g}$ works given that the states of the n dummy components in the 0th circle

are given by $\boldsymbol{h}$. Note that $\boldsymbol{h}, \boldsymbol{g} \in \{0,1\}^n$. Then, we obtain $C(m;\boldsymbol{h};\boldsymbol{g})$ as follows:

$$C(m;\boldsymbol{h};\boldsymbol{g}) = \boldsymbol{\pi}_0(\boldsymbol{h}) \left(\prod_{i=1}^{m} M^C(i) \right) \boldsymbol{\pi}_m(\boldsymbol{g})^{\mathsf{T}}, \tag{A.8}$$

where $\boldsymbol{\pi}_0(\boldsymbol{h})$ is a $|W^C|$-dimensional vector in which an element corresponding to $\boldsymbol{h}$ is unity, and the others are zero; $\boldsymbol{\pi}_m(\boldsymbol{g})$ is a $|W^C|$-dimensional vector in which an element corresponding to $\boldsymbol{g}$ is unity, and the others are zero. Note that

$$W^C = \left\{ \boldsymbol{h} \;\middle|\; \sum_{j=2}^{n} h_{j-1}h_j = 0 \right\}.$$

When we bond the 1st circle and the mth circle, the states of the n components in the 0th circle should be coincident with the states of the n components in the mth circle. Hence, we have

$$\begin{aligned} R^T((m,n),P) &= \sum_{\boldsymbol{h}\in\{0,1\}^n} C(m;\boldsymbol{h};\boldsymbol{h}), \\ &= \sum_{\boldsymbol{h}\in W^C} C(m;\boldsymbol{h};\boldsymbol{h}), \\ &= \sum_{\boldsymbol{h}\in W^C} \boldsymbol{\pi}_0(\boldsymbol{h}) \left(\prod_{i=1}^{m} M^C(i) \right) \boldsymbol{\pi}_m(\boldsymbol{h})^{\mathsf{T}}, \qquad \text{(from Eq. (A.8))} \end{aligned} \tag{A.9}$$

where the second equality is obtained by

$$C(m;\boldsymbol{h};\boldsymbol{h}) = 0,$$

for $\boldsymbol{h} \in \{0,1\}^n \setminus W^C$ because there exist two consecutive failed components.

In Eq. (A.9), $\boldsymbol{\pi}_0(\boldsymbol{h})$ and $\boldsymbol{\pi}_m(\boldsymbol{h})$ are the same for $\boldsymbol{h} \in W^C$. This means that Eq. (A.9) shows the summation of all the elements on the main diagonal (i.e., the diagonal from the upper left to the lower right) of the matrix, which is so-called *trace*, is the reliability of a Tor/$(1,2)$-or-$(2,1)$/(m,n):F system. Thus, we can get

$$R^T((m,n),P) = \mathrm{Tr}\left(\prod_{i=1}^{m} M^C(i) \right).$$

Theorem 2.9 has been proved. □

Proof of Theorem 2.10

Proof. Letting UB_{diff} and LB_{diff} be the upper and lower bounds for $D_b([1,m],n)$, respectively, then from Eq. (2.62), we obtain

$$LB_D = R^L((b,n),P_{[1,b]}) \times R^L((m-b,n),P_{[b+1,m]}) - UB_{diff}, \tag{A.10}$$

and

$$UB_D = R^L((b,n),P_{[1,b]}) \times R^L((m-b,n),P_{[b+1,m]}) - LB_{diff}. \tag{A.11}$$

Now, UB_{diff} and LB_{diff} can be computed by the following equations:

$$UB_{diff} = R^L((l_1-1,n),P_{[1,l_1-1]}) \times D_b([l_1,l_2],n) \times R^L((m-l_2,n),P_{[l_2+1,m]}), \tag{A.12}$$

and

$$LB_{diff} = S^{(l_1)}((l_1,n),P_{[1,l_1]}) \times D_b([l_1,l_2],n) \times S^{(l_2)}((m-l_2+1,n),P_{[l_2,m]}), \tag{A.13}$$

where Eq. (A.12) is established in a similar manner as Eq. (1.18), and Eq. (A.13) is established in a similar manner as Eq. (1.17). Besides, Eq. (2.66) can be obtained directly from Eq. (2.63) Thus, the proof is completed. □

Proof of Theorem 3.1

Proof. If $0 \le z \le r-1$, a Lin/(r,s)/(m,j):F system does not have r consecutive failed components in the jth column, and hence $\boldsymbol{g} = (0,\ldots,0)$. Thus, we obtain

$$\alpha(j;z;\boldsymbol{g}) = \begin{cases} \binom{mj}{z}, & \text{if } 0 \le z \le r-1 \text{ and } \boldsymbol{g} = (0,\ldots,0), \\ 0, & \text{otherwise,} \end{cases} \tag{A.14}$$

for $j = 1,2,\ldots,n$ and $\boldsymbol{g} \in S$.

If $z > mj$, by the definition of $\alpha(j; z; \boldsymbol{g})$ in Eq. (3.14), then we can get

$$\alpha(j; z; \boldsymbol{g}) = 0, \tag{A.15}$$

for $j = 1, 2, \ldots, n$ and $\boldsymbol{g} \in S$.

If $r \leq z \leq mj$ for $j = 1, 2, \ldots, n$, we obtain $\alpha(j; z; \boldsymbol{g})$ for $\boldsymbol{g} \in S \setminus E$ by using the event decomposition approach [22]. Specifically, by enumerating of the states of m components in the jth column of the Lin/(r, s)/(m, j):F system, $\alpha(j; z; \boldsymbol{g})$ can be expressed as a summation of the number of path sets of the Lin/(r, s)/(m, j):F system that satisfies the following conditions:

(a) The system has exactly z failed components.

(b) The system fulfills the additional condition, namely, $A_j(\boldsymbol{g}) = 1$.

(c) The states of the m components in the jth column are given by a vector $\boldsymbol{y}$,

where $y_i = 0$ (1) means that the ith component in a column works (fails) for $i = 1, 2, \ldots, m$. That is,

$$\alpha(j; z; \boldsymbol{g}) = \sum_{\boldsymbol{y} \in \{0,1\}^m} \left| h(j; z; \boldsymbol{g}; \boldsymbol{y}) \right|, \tag{A.16}$$

where

$$h(j; z; \boldsymbol{g}; \boldsymbol{y}) = \left\{ (\boldsymbol{x}_1^{\mathsf{T}}, \ldots, \boldsymbol{x}_j^{\mathsf{T}}) \;\middle|\; \begin{array}{l} \phi(\boldsymbol{x}_1^{\mathsf{T}}, \ldots, \boldsymbol{x}_j^{\mathsf{T}}) = 0, \sum_{a=1}^{j} N(\boldsymbol{x}_a) = z, \\ A_j(\boldsymbol{g}) = 1, \text{ and } x_{ij} = y_i, \ (i = 1, 2, \ldots, m) \end{array} \right\}.$$

From the definition of $A_j(\boldsymbol{g})$ in Eq. (3.13), if $g_k \geq 1$, components $(k, j), (k + 1, j), \ldots, (k + r - 1, j)$ fail (that is, $\prod_{u=k}^{k+r-1} y_u = 1$); if $g_k = 0$, at least one component works in components $(k, j), (k+1, j), \ldots, (k+r-1, j)$ (that is, $\prod_{u=k}^{k+r-1} y_u = 0$) as shown in Fig. A.1. Hence, there exists no Lin/(r, s)/(m, j):F system such that $\boldsymbol{y}$ does not satisfy

$$\prod_{u=k}^{k+r-1} y_u = \begin{cases} 1, & \text{if } g_k \geq 1, \\ 0, & \text{if } g_k = 0. \end{cases} \tag{A.17}$$

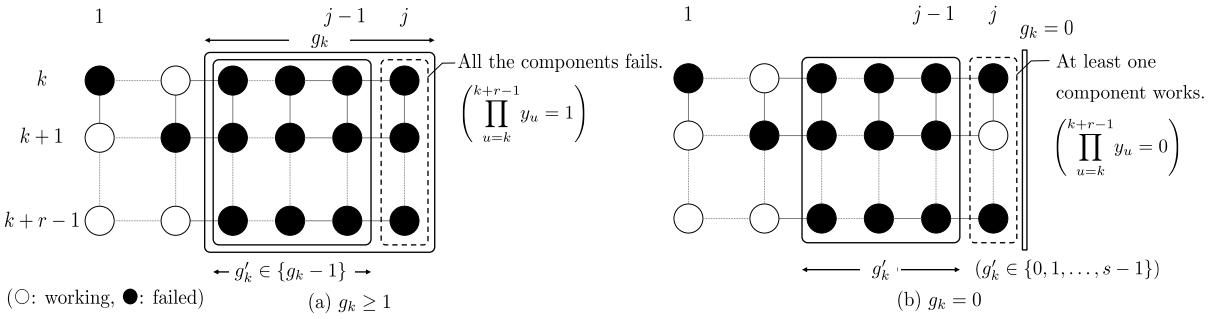


Figure A.1: Relationship among $\boldsymbol{g}$, $\boldsymbol{g}'$, and $\boldsymbol{y}$.

Thus, $h(j;z;\boldsymbol{g};\boldsymbol{y})=0$, and then, we have

$$\alpha(j;z;\boldsymbol{g})=\sum_{\boldsymbol{y}\in U(\boldsymbol{g})}\left|h(j;z;\boldsymbol{g};\boldsymbol{y})\right|, \tag{A.18}$$

where $U(\boldsymbol{g})$ is given in Eq. (3.19). Let $\boldsymbol{g}'$ be an $(m-r+1)$-dimensional vector $(g'_1,g'_2,\ldots,g'_{m-r+1})$, where $g'_k\in\{0,1,\ldots,s\}$ for $k=1,2,\ldots,m-r+1$. Also, $\alpha(j-1;z';\boldsymbol{g}')$ denotes the number of path sets of a Lin/(r,s)/$(m,j-1)$:F system with exactly z' failed components and the additional condition given by $\boldsymbol{g}'$. Then, if $\prod_{u=k}^{k+r-1}y_u=1$, then g'_k can take only g_k-1 (see Fig. A.1 (a)); if $\prod_{u=k}^{k+r-1}y_u=0$, then g'_k can take $0,1,\ldots,s-1$ (see Fig. A.1 (b)). Also, if the Lin/(r,s)/(m,j):F system has exactly z failed components, and the states of "components in the jth column" are given by a vector $\boldsymbol{y}$, then the Lin/(r,s)/$(m,j-1)$:F system has exactly $z-N(\boldsymbol{y})$ failed components. Thus, using $\Theta(\boldsymbol{g})$ in Eq. (3.20), we obtain

$$\begin{aligned}\alpha(j;z;\boldsymbol{g})&=\sum_{\boldsymbol{y}\in U(\boldsymbol{g})}\sum_{\boldsymbol{g}'\in\Theta(\boldsymbol{g})}\left|\left\{(\boldsymbol{x}_1^{\mathsf{T}},\ldots,\boldsymbol{x}_{j-1}^{\mathsf{T}})\;\middle|\;\begin{array}{l}\phi^{RS}(\boldsymbol{x}_1^{\mathsf{T}},\ldots,\boldsymbol{x}_j^{\mathsf{T}})=0,\\ \displaystyle\sum_{a=1}^{j-1}N(\boldsymbol{x}_a)=z-N(\boldsymbol{y}),\\ \text{and } A_{j-1}(\boldsymbol{g}')=1\end{array}\right\}\right|,\\ &=\sum_{\boldsymbol{y}\in U(\boldsymbol{g})}\sum_{\boldsymbol{g}'\in\Theta(\boldsymbol{g})}\alpha(j-1;z-N(\boldsymbol{y});\boldsymbol{g}').\end{aligned} \tag{A.19}$$

Therefore, Eq. (3.18) holds from the definition of $\alpha(j;z;\boldsymbol{g})$ in Eq. (3.14).

Next, from Eq. (3.15), $\alpha_z((r,s),(m,n))$ can be obtained by summing up $\alpha(n;z;\boldsymbol{g})$s for $\boldsymbol{g}\in S\setminus E$, and thus, Eq. (3.22) holds. Finally, from Eq. (3.4), we can get Eq. (3.23) directly. The proof of Theorem 3.1 is completed. □

Proof of Theorem 3.2

Proof. If $z < N(\boldsymbol{y})$ or $z > N(\boldsymbol{y}) + \left\lceil \frac{m(j-1)}{2} \right\rceil$, by the definition of $\beta(j; z; \boldsymbol{y})$ in Eq. (3.27), then

$$\beta(j; z; \boldsymbol{y}) = 0. \tag{A.20}$$

If $N(\boldsymbol{y}) \leq z \leq N(\boldsymbol{y}) + \left\lceil \frac{m(j-1)}{2} \right\rceil$, then $\beta(j; z; \boldsymbol{y})$ can be computed by summing up over all $\boldsymbol{x} \in \{0,1\}^m$, and thus,

$$\beta(j; z; \boldsymbol{y}) = \sum_{\boldsymbol{x} \in \{0,1\}^m} \beta(j-1; z - N(\boldsymbol{y}); \boldsymbol{x}), \tag{A.21}$$

holds. Now, using the definition of $\beta(j; z; \boldsymbol{y})$, for all $\boldsymbol{x} \in \{0,1\}^m \setminus \Omega(\boldsymbol{y})$,

$$\beta(j; z; \boldsymbol{y}) = 0, \tag{A.22}$$

holds because there exist two consecutive failed components on column $j-1$ or the rows of columns $j-1$ and j, that is, $\phi^{OR}(\boldsymbol{x}_1^{\mathsf{T}}, \ldots, \boldsymbol{x}_j^{\mathsf{T}}) = 0$. Accordingly, we can get

$$\beta(j; z; \boldsymbol{y}) = \sum_{\boldsymbol{x} \in \Omega(\boldsymbol{y})} \beta(j-1; z - N(\boldsymbol{y}); \boldsymbol{x}). \tag{A.23}$$

This completes the proof. □

Permutation Importance

To prove Lemmas 4.1 and 4.2, we first present the concept of permutation importance, which was introduced by Boland *et al.* [124]. First, according to Kuo and Zhu [86], we define the following notation. Let us consider a coherent system consisting of n components. A state vector of this system is given by $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, where for $i \in \{1, 2, \ldots, n\}$, $x_i = 0$ if the ith component works, and $x_i = 1$ if it fails. For $i \in \{1, 2, \ldots, n\}$, let p_i be the reliability of the ith component and $\boldsymbol{p} = (p_1, p_2, \ldots, p_n)$. Let $R(\boldsymbol{p})$ denote the system reliability under component reliability vector $\boldsymbol{p}$. For $i, j \in \{1, 2, \ldots, n\}$ $(i \neq j)$, the symbol $(\alpha_i, \beta_j, \boldsymbol{x}^{(ij)})$ denotes a state vector with $x_i = \alpha$ and $x_j = \beta$ and the other elements are given by $\boldsymbol{x}^{(ij)}$, where $\alpha, \beta \in \{0, 1\}$ and $\boldsymbol{x}^{(ij)}$ is a

vector obtained by deleting x_i and x_j from $\boldsymbol{x}$.

Boland *et al.* [124] defined the permutation importance as follows:

Definition A.1 (Boland *et al.* [124])**.** For $i, j \in \{1, 2, \ldots, n\}$ $(i \neq j)$, the ith component is more permutation important than the jth component, denoted by $i >_{pe} j$, for structure function ϕ if

$$\phi(0_i, 1_j, \boldsymbol{x}^{(ij)}) \geq \phi(1_i, 0_j, \boldsymbol{x}^{(ij)}), \tag{A.24}$$

holds for all $\boldsymbol{x}^{(ij)}$ and the strict inequality holds for some $\boldsymbol{x}^{(ij)}$. If the equality holds for all $\boldsymbol{x}^{(ij)}$, the ith component and the jth component are said to be permutation equivalent, denoted by $i =_{pe} j$.

Several properties for the permutation importance were reported. Koutras *et al.* [90] provided a transitivity property of the permutation importance as follows:

Lemma A.1 (Koutras *et al.* [90])**.** If $i >_{pe} j$ and $j >_{pe} k$, then $i >_{pe} k$.

For the proof of Lemma A.1, readers are referred to Koutras *et al.* [90].

Boland *et al.* [124] revealed the relationship between the permutation importance and the system reliability as follows:

Lemma A.2 (Boland *et al.* [124])**.** For $i, j \in \{1, 2, \ldots, n\}$ $(i \neq j)$,

(a) $i >_{pe} j$ if and only if $R(\beta_i, \alpha_j, \boldsymbol{p}^{(ij)}) \geq R(\alpha_i, \beta_j, \boldsymbol{p}^{(ij)})$ for all $\boldsymbol{p}^{(ij)}$ and all $0 < \alpha < \beta < 1$ with the strict inequality for some $\boldsymbol{p}^{(ij)}$, α and β.

(b) $i =_{pe} j$ if and only if $R(\beta_i, \alpha_j, \boldsymbol{p}^{(ij)}) = R(\alpha_i, \beta_j, \boldsymbol{p}^{(ij)})$ for all $\boldsymbol{p}^{(ij)}$ and all $0 < \alpha < \beta < 1$.

For the proof of Lemma A.2, readers are referred to Boland *et al.* [124].

Note that $(\alpha_i, \beta_j, \boldsymbol{p}^{(ij)})$ is a vector with $p_i = \alpha$, $p_j = \beta$ and the other elements are given by $\boldsymbol{p}^{(ij)}$ for $i, j \in \{1, 2, \ldots, n\}$ $(i \neq j)$, where $\boldsymbol{p}^{(ij)}$ is a vector obtained by deleting p_i and p_j from $\boldsymbol{p}$. Let consider a system with an arrangement such that $p_i < p_j$. If $i >_{pe} j$ and the components assigned to positions i and j are exchanged, Lemma A.2 [124] (a) states that the exchanged system is more reliable than the original system. Hence, to improve the system reliability, we should assign a reliable component to not position j but position i. Also, Lemma A.2 [124] (b) states that if $i =_{pe} j$, even when the

components assigned to positions i and j are exchanged, the system reliability does not change.

In general, we have difficulty in identifying the permutation importance from Definition A.1 due to the computational complexity. Boland *et al.* [124] and Meng [125] provided a theorem for identifying the permutation importance from minimal cuts.

Lemma A.3 (Boland *et al.* [124], Meng [125])**.** For $i, j \in \{1, 2, \ldots, n\}$ $(i \neq j)$,

(a) If $\mathbb{C}_j$ is a proper subset of $\mathbb{C}_i$ $(\mathbb{C}_j \subset \mathbb{C}_i)$, then $i >_{pe} j$.

(b) If $\mathbb{C}_j = \mathbb{C}_i$, then $i =_{pe} j$.

Note that, for $i \in \{1, 2, \ldots, n\}$, $\mathbb{C}_i$ is a set of minimal cuts containing the ith component. For the proof of Lemma A.3, readers are referred to Boland *et al.* [124] and Meng [125].

Proof of Lemma 4.1

Using the permutation importance, we will now prove Lemma 4.1. Because it is easy to find the minimal cuts of a Lin/(r, s)/(m, n):F system, we use Lemma A.3 [124, 125] to identify the permutation importance. First, to indicate minimal cuts of the system, we need some additional notation. A set of all components is denoted by

$$I = \left\{ \; (i, j) \;\middle|\; 1 \leq i \leq m \text{ and } 1 \leq j \leq n \; \right\},$$

For $u = 1, 2, \ldots, m - r + 1$ and $v = 1, 2, \ldots, n - s + 1$, a minimal cut with component $\pi(u, v)$ at the upper left corner is given by

$$K_{(u,v)} = \left\{ \; (i, j) \in I \;\middle|\; u \leq i \leq u + r - 1 \text{ and } v \leq j \leq v + s - 1 \; \right\},$$

and the family of the system's minimal cut set is

$$\mathbb{C} = \left\{ \; K_{(u,v)} \;\middle|\; 1 \leq u \leq m - r + 1 \text{ and } 1 \leq v \leq n - s + 1 \; \right\}.$$

A set $\mathbb{C}_{(i,j)}$ is also defined, $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$, as

$$\mathbb{C}_{(i,j)} = \left\{ K_{(u,v)} \in \mathbb{C} \;\middle|\; \begin{array}{l} \max\{1, i-r+1\} \le u \le \min\{m-r+1, i\} \text{ and} \\ \max\{1, j-s+1\} \le v \le \min\{n-s+1, j\} \end{array} \right\}, \quad \text{(A.25)}$$

which shows a set of minimal cuts including position (i, j).

Proof. Let us now give the proof of Lemma 4.1 (a). From $2 \le j \le \min\{s, n-s+1\}$, we can get

$$j - s + 1 \ge 2 - s + 1 = 3 - s.$$

From $s \ge 2$, $\max\{1, j-s+1\} = 1$. If $s \ge n-s+1$, then $2 \le j \le n-s+1$, and thus, we have $\min\{n-s+1, j\} = j$. Also, if $s < n-s+1$, then $2 \le j \le s < n-s+1$, and thus, we have $\min\{n-s+1, j\} = j$. If $2 \le j \le \min\{s, n-s+1\}$, for $i = 1, 2, \ldots, m$, from Eq. (A.25), we can get

$$\mathbb{C}_{(i,j)} = \left\{ K_{(u,v)} \in \mathbb{C} \;\middle|\; \alpha \le u \le \beta \text{ and } 1 \le v \le j \right\},$$
$$\mathbb{C}_{(i,j-1)} = \left\{ K_{(u,v)} \in \mathbb{C} \;\middle|\; \alpha \le u \le \beta \text{ and } 1 \le v \le j-1 \right\},$$

where $\alpha = \max\{1, i-r+1\}$ and $\beta = \min\{m-r+1, i\}$. Hence, $\mathbb{C}_{(i,j-1)} \subset \mathbb{C}_{(i,j)}$, and thus $(i, j) >_{pe} (i, j-1)$ from Lemma A.3 [124, 125] (a). Therefore, by Lemma A.1 [90] and Lemma A.2 [124] (a), Lemma 4.1 (a) is completely proved. □

The rest of Lemma 4.1 can also be proved in a similar way.

Proof of Lemma 4.2

Proof. We prove Lemma 4.2 (a). Since $m - r + 1 \le i \le r$, from

$$i - r + 1 \ge r - r + 1 = r,$$

we can get $\max\{1, i-r+1\} = 1$. Also, from $i \ge m-r+1$, we can get $\min\{m-r+1, i\} = m - r + 1$. If $2r \le m$, then there exists no i such that $m - r + 1 \le i \le r$. If $2r > m$, for $j = 1, 2, \ldots, n$, a set is defined by

$$A_j = \{(i, j) \mid m - r + 1 \le i \le r\}.$$

For $(i_1, j), (i_2, j) \in A_j$ $(j = 1, 2, \ldots, n)$, from Eq. (A.25), we have

$$\mathbb{C}_{(i_1,j)} = \mathbb{C}_{(i_2,j)},$$

because $\max\{1, i-r+1\} = 1$ and $\min\{m-r+1, i\} = m-r+1$ for all i $(m-r+1 \leq i \leq r)$. Thus, $(i_1, j) =_{pe} (i_2, j)$ from Lemma A.2 [124] (b). Therefore, by Lemma A.1 [90] and Lemma A.2 [124] (b), Lemma 4.2 (a) is completely proved. □

Lemma 4.2 (b) can also be proved in a similar way.

References

[1] M. J. Zuo, Z. Tian, and H.-Z. Huang, "Neural networks for reliability-based optimal design," in *Computational Intelligence in Reliability Engineering: New Metaheuristics, Neural and Fuzzy Techniques in Reliability* (G. Levitin, ed.), pp. 175–196, Springer, 2007.

[2] A. Birolini, *Reliability Engineering: Theory and Practice*. Springer, 2017.

[3] R. E. Barlow and F. Proschan, *Mathematical Theory of Reliability*. John Wiley, 1965.

[4] R. K. Sharma, D. Kumar, and P. Kumar, "Modeling system behavior for risk and reliability analysis using KBARM," *Quality and Reliability Engineering International*, vol. 23, no. 8, pp. 973–998, 2007.

[5] B. K. Lad, M. S. Kulkarni, and K. B. Misra, "Optimal reliability design of a system," in *Handbook of Performability Engineering* (K. B. Misra, ed.), pp. 499–519, Springer, 2008.

[6] A. Lisnianski and G. Levitin, *Multi-State System Reliability: Assessment, Optimization and Applications*, vol. 6. World Scientific, 2003.

[7] W. Kuo and V. R. Prasad, "An annotated overview of system-reliability optimization," *IEEE Transactions on Reliability*, vol. 49, no. 2, pp. 176–187, 2000.

[8] W. Kuo and R. Wan, "Recent advances in optimal reliability allocation," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, vol. 37, no. 2, pp. 143–156, 2007.

[9] D. W. Coit and E. Zio, "The evolution of system reliability optimization," *Reliability Engineering and System Safety*, vol. 192, pp. 1–11, 2018.

[10] X. Huang, F. P. Coolen, and T. Coolen-Maturi, "A heuristic survival signature based approach for reliability-redundancy allocation," *Reliability Engineering and System Safety*, vol. 185, pp. 511–517, 2019.

[11] Q. Yao, X. Zhu, and W. Kuo, "A Birnbaum-importance based genetic local search algorithm for component assignment problems," *Annals of Operations Research*, vol. 212, no. 1, pp. 185–200, 2014.

[12] J. M. Kontoleon, "Reliability determination of a r-successive-out-of-n:F system," *IEEE Transactions on Reliability*, vol. 29, no. 5, p. 437, 1980.

[13] D. T. Chiang and S.-C. Niu, "Reliability of consecutive-k-out-of-n:F system," *IEEE Transactions on Reliability*, vol. 30, no. 1, pp. 87–89, 1981.

[14] I. S. Triantafyllou, "Consecutive-type reliability systems: An overview and some applications," *Journal of Quality and Reliability Engineering*, vol. 2015, pp. 1–20, 2015.

[15] W. Y. Yun and A. J. Endharta, "Preventive maintenance of consecutive multi-unit systems," in *Advances in Reliability and System Engineering* (M. Ram and J. P. Davim, eds.), pp. 27–51, Springer, 2017.

[16] S. D. Dafnis, F. S. Makri, and A. N. Philippou, "The reliability of a generalized consecutive system," *Applied Mathematics and Computation*, vol. 359, pp. 186–193, 2019.

[17] R. Peng, D. Wu, and K. Gao, "Reliability of a dual linear consecutive system with three failure modes," in *Stochastic Models in Reliability, Network Security and System Safety* (Q.-L. Li, J. Wang, and H.-B. Yu, eds.), pp. 259–269, Springer, 2019.

[18] J. Zhao, S. Si, and Z. Cai, "A multi-objective reliability optimization for reconfigurable systems considering components degradation," *Reliability Engineering and System Safety*, vol. 183, pp. 104–115, 2019.

[19] J. Shen and L. Cui, "Reliability and Birnbaum importance for sparsely connected circular consecutive-k systems," *IEEE Transactions on Reliability*, vol. 64, no. 4, pp. 1140–1157, 2015.

[20] M. T. Chao, J. C. Fu, and M. V. Koutras, "Survey of reliability studies of consecutive-k-out-of-n:F & related systems," *IEEE Transactions on Reliability*, vol. 44, no. 1, pp. 120–127, 1995.

[21] G. J. Chang, L. Cui, and F. K. Hwang, *Reliabilities of Consecutive-k Systems*, vol. 4. Kluwer Academic Publishers, 2000.

[22] W. Kuo and M. J. Zuo, *Optimal Reliability Modeling: Principles and Applications*. John Wiley and Sons, 2003.

[23] J.-C. Chang and F. K. Hwang, "Reliabilities of consecutive-k systems," in *Handbook of Reliability Engineering* (H. Pham, ed.), ch. 2, pp. 37–59, Springer, 2003.

[24] S. Eryilmaz, "Review of recent advances in reliability of consecutive k-out-of-n and related systems," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 224, no. 3, pp. 225–237, 2010.

[25] L. Cui and Q. Dong, "Consecutive k and related models — a survey," in *Stochastic Models in Reliability, Network Security and System Safety* (Q.-L. Li, J. Wang, and H.-B. Yu, eds.), pp. 3–18, Springer, 2019.

[26] A. A. Salvia and W. C. Lasher, "2-dimensional consecutive-k-out-of-n:F models," *IEEE Transactions on Reliability*, vol. 39, no. 3, pp. 382–385, 1990.

[27] T. K. Boehme, A. Kossow, and W. Preuss, "A generalization of consecutive-k-out-of-n:F systems," *IEEE Transactions on Reliability*, vol. 41, no. 3, pp. 451–457, 1992.

[28] C. Lin, L. Cui, D. W. Coit, and M. Lv, "Reliability modeling on consecutive-k_r-out-of-n_r:F linear zigzag structure and circular polygon structure," *IEEE Transactions on Reliability*, vol. 65, no. 3, pp. 1509–1521, 2016.

[29] C.-Y. Chong and S. P. Kumar, "Sensor networks: evolution, opportunities, and challenges," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1247–1256, 2003.

[30] T. Antoniou, A. Boukerche, I. Chatzigiannakis, G. Mylonas, and S. Nikoletseas, "A new energy efficient and fault-tolerant protocol for data propagation in smart dust networks using varying transmission range," in *Proceedings of the 37th Annual Simulation Symposium*, pp. 1–15, 2004.

[31] Q. Liu, "k-barrier coverage reliability evaluation for wireless sensor networks using two-dimensional k-within-consecutive-$r \times s$-out-of-$m \times n$:F system," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 233, no. 6, pp. 1029–1039, 2019.

[32] T. Yuge, M. Dehare, and S. Yanagi, "Reliability and availability of a repairable lattice system," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E83-A, no. 5, pp. 782–786, 2000.

[33] C. Lin, Z. Zeng, Y. Zhou, M. Xu, and Z. Ren, "A lower bound of reliability calculating method for lattice system with non-homogeneous components," *Reliability Engineering and System Safety*, vol. 188, pp. 36–46, 2019.

[34] K. Noguchi, M. Sasaki, S. Yanagi, and T. Yuge, "System reliability of a sensor system considering the detectable area," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences (in Japanese)*, vol. J79-A, no. 8, pp. 1444–1453, 1996.

[35] T. Akiba and H. Yamamoto, "Reliability of a 2-dimensional k-within-consecutive-$r \times s$-out-of-$m \times n$:F system," *Naval Research Logistics*, vol. 48, no. 7, pp. 625–637, 2001.

[36] F. S. Makri and Z. M. Psillakis, "Bounds for reliability of k-within connected-(r, s)-out-of-(m, n) failure systems," *Microelectronics Reliability*, vol. 37, no. 8, pp. 1217–1224, 1997.

[37] H. Yamamoto, "Reliability of a connected-(r_1, s_1)-or-(r_2, s_2)-or-. . .-or-(r_k, s_k)-out-of-(m, n):F lattice system," *Microelectronics Reliability*, vol. 36, no. 2, pp. 151–168, 1996.

[38] F. S. Makri and Z. M. Psillakis, "Bounds for reliability of k-within two-dimensional consecutive-r-out-of-n failure systems," *Microelectronics Reliability*, vol. 36, no. 3, pp. 341–345, 1996.

[39] M. J. Zuo, D. Lin, and Y. Wu, "Reliability evaluation of combined k-out-of-n:F, consecutive-k-out-of-n:F and linear connected-(r, s)-out-of-(m, n):F system structures," *IEEE Transactions on Reliability*, vol. 49, no. 1, pp. 99–104, 2000.

[40] H. Yamamoto and M. Miyakawa, "The reliability of the adjacent triangle:F triangular lattice system," *The Journal of Reliability Engineering Association of Japan (in Japanese)*, vol. 7, no. 18, pp. 97–105, 1996.

[41] M. Boushaba and Z. Azouz, "Reliability bounds of a 3-dimensional consecutive-k-out-of-n:F system," *International Journal of Reliability, Quality and Safety Engineering*, vol. 18, no. 1, pp. 51–59, 2011.

[42] T. Akiba, "Improvement ideas of the reliability calculation algorithm for the 3-dimensional consecutive-k system," in *Proceedings of the 8th Asia-Pacific International Symposium on Advanced Reliability and Maintenance Modeling*, no. 1, pp. 361–366, 2018.

[43] T. Akiba, N. Takahashi, and H. Yamamoto, "Reliability of a 3-dimensional consecutive-k system," in *Proceedings of the 9th International Conference on Mathematical Methods in Reliability*, pp. 1144–1151, 2015.

[44] M. G. Kulkarni and A. S. Kashikar, "Signature and reliability of conditional three-dimensional consecutive-(s, s, s)-out-of-(s, s, m):F system," *International Journal of Reliability, Quality and Safety Engineering*, vol. 21, no. 2, pp. 1–13, 2014.

[45] S. Cowell, "A formula for the reliability of a d-dimensional consecutive-k-out-of-n:F system," *International Journal of Combinatorics*, vol. 2015, pp. 1–6, 2015.

[46] A. P. Godbole, L. K. Potter, and J. K. Sklar, "Improved upper bounds for the reliability of d-dimensional consecutive-k-out-of-n:F systems," *Naval Research Logistics*, vol. 45, no. 2, pp. 219–230, 1997.

[47] T. Akiba, H. Yamamoto, and Y. Kainuma, "Reliability of a 3-dimensional adjacent triangle:F triangular lattice system," in *Proceedings of the 5th Asia Pacific Industrial Engineering and Management Systems*, pp. 1–10, 2004.

[48] H. Yamamoto and T. Akiba, "Survey of reliability studies of multi-dimensional consecutive-k-out-of-n:F systems," *Reliability Engineering Association of Japan*, vol. 25, no. 8, pp. 783–796, 2003.

[49] T. Akiba, T. Nakamura, X. Xiao, and H. Yamamoto, "Evaluation methods for reliability of consecutive-k systems," in *Systems Engineering: Reliability Analysis*

Using k-out-of-n Structures (M. Ram and T. Dohi, eds.), ch. 1, pp. 1–24, CRC Press, 2019.

[50] X. Zhu, Q. Yao, and W. Kuo, "Patterns of the Birnbaum importance in linear consecutive-k-out-of-n systems," *IIE Transactions*, vol. 44, no. 4, pp. 277–290, 2012.

[51] F. K. Hwang, "Fast solutions for consecutive-k-out-of-n:F system," *IEEE Transactions on Reliability*, vol. 31, no. 5, pp. 447–448, 1982.

[52] M.-S. Lin, "An $O(k^2 \log(n))$ algorithm for computing the reliability of consecutive-k-out-of-n:F systems," *IEEE Transactions on Reliability*, vol. 53, no. 1, pp. 3–6, 2004.

[53] L. Cui, Y. Xu, and X. Zhao, "Developments and applications of the finite Markov chain imbedding approach in reliability," *IEEE Transactions on Reliability*, vol. 59, no. 4, pp. 685–690, 2010.

[54] J. C. Fu, "Reliability of consecutive-k-out-of-n:F systems with $(k-1)$-step Markov dependence," *IEEE Transactions on Reliability*, vol. 35, no. 5, pp. 602–606, 1986.

[55] J. C. Fu and B. Hu, "On reliability of a large consecutive-k-out-of-n:F system with $(k-1)$-step Markov dependence," *IEEE Transactions on Reliability*, vol. 36, no. 1, pp. 75 – 77, 1987.

[56] J. C. Fu and M. V. Koutras, "Distribution theory of runs: A Markov chain approach," *Journal of the American Statistical Association*, vol. 89, no. 427, pp. 1050–1058, 1994.

[57] X. Zhao and L. Cui, "On the accelerated scan finite Markov chain imbedding approach," *IEEE Transactions on Reliability*, vol. 58, no. 2, pp. 383–388, 2009.

[58] H. Yamamoto and M. Miyakawa, "Reliability of a linear connected-(r, s)-out-of-(m, n):F lattice system," *IEEE Transactions on Reliability*, vol. 44, no. 2, pp. 333–336, 1995.

[59] K. Noguchi, M. Sasaki, S. Yanagi, and T. Yuge, "Reliability of connected-(r, s)-out-of-(m, n):F system," *Electronics and Communications in Japan*, vol. 80, no. 1, pp. 954–962, 1997.

[60] X. Zhao, L. Cui, W. Zhao, and F. Liu, "Exact reliability of a linear connected-(r, s)-out-of-(m, n):F system," *IEEE Transactions on Reliability*, vol. 60, no. 3, pp. 689–698, 2011.

[61] H. Yamamoto and M. Miyakawa, "Reliability of a circular connected-(r, s)-out-of-(m, n):F lattice system," *Journal of the Operations Research Society of Japan*, vol. 39, no. 3, pp. 389–406, 1996.

[62] H. Yamamoto and T. Akiba, "A recursive algorithm for the reliability of a circular connected-(r, s)-out-of-(m, n):F lattice system," *Computers and Industrial Engineering*, vol. 49, no. 1, pp. 21–34, 2005.

[63] Y. Higashiyama, "An algorithm for exact reliability of consecutive 2-out-of-(m, n):F system with un-equal component probability," in *Proceedings of the IEEE Region 10 Conference*, vol. 2, pp. 990–993, 1999.

[64] H. Yamamoto, T. Akiba, H. Nagatsuka, and Y. Moriyama, "Recursive algorithm for the reliability of a connected-$(1, 2)$-or-$(2, 1)$-out-of-(m, n):F lattice system," *European Journal of Operational Research*, vol. 188, no. 3, pp. 854–864, 2008.

[65] Y. Higashiyama, "A recursive method for computing the reliability of circular consecutive 2-out-of-(m, n):F system," *International Journal of Reliability, Quality and Safety Engineering*, vol. 9, no. 3, pp. 229–235, 2002.

[66] J. D. Esary and F. Proschan, "Coherent structures of non-identical components," *Technometrics*, vol. 5, no. 2, pp. 191–209, 1963.

[67] M. V. Boutsikas and M. V. Koutras, "Generalized reliability bounds for coherent structures," *Journal of Applied Probability*, vol. 37, no. 3, pp. 778–794, 2000.

[68] J. Fu and M. Koutras, "Reliability bounds for coherent structures with independent components," *Statistics and Probability Letters*, vol. 22, no. 2, pp. 137–148, 1995.

[69] H. Yamamoto, "Reliability of connected-**X**-out-of-(m, n):F lattice systems," *Proceedings of the Institute of Statistical Mathematics (in Japanese)*, vol. 44, no. 2, pp. 201–210, 1996.

[70] Y. C. Hsieh and T. C. Chen, "Reliability lower bounds for two-dimensional consecutive-k-out-of-n:F systems," *Computers and Operations Research*, vol. 31, no. 8, pp. 1259–1272, 2004.

[71] V. Beiu and L. Daus, "Reliability bounds for two dimensional consecutive systems," *Nano Communication Networks*, vol. 6, no. 3, pp. 145–152, 2015.

[72] S. Si, Z. Cai, S. Sun, and S. Zhang, "Integrated importance measures of multi-state systems under uncertainty," *Computers and Industrial Engineering*, vol. 59, no. 4, pp. 921–928, 2010.

[73] F. Ohi, "Importance measures for a binary state system," in *Reliability Modeling With Computer and Maintenance Applications* (S. Nakamura, C. H. Qian, and T. Nakagawa, eds.), ch. 6, pp. 103–137, World Scientific, 2017.

[74] K. P. Amrutkar and K. K. Kamalja, "An overview of various importance measures of reliability system," *International Journal of Mathematical, Engineering and Management Sciences*, vol. 2, no. 3, pp. 150–171, 2017.

[75] Z. W. Birnbaum, "On the importance of different components in a multicomponent system," in *Multivariate Analysis II* (P. R. Krishnaiah, ed.), pp. 581–592., Academic Press, 1969.

[76] K. K. Kamalja, "Birnbaum importance for consecutive-k systems," *International Journal of Reliability, Quality and Safety Engineering*, vol. 19, no. 4, pp. 1–25, 2012.

[77] H. Dui and S. Si, "Component importance-driven reliability optimization for linear consecutive k out of n systems," in *Stochastic Models in Reliability, Network Security and System Safety* (Q.-L. Li, J. Wang, and H.-B. Yu, eds.), pp. 259–269, Springer, 2019.

[78] F. J. Samaniego, *System Signatures and their Applications in Engineering Reliability*. Springer, 2007.

[79] F. J. Samaniego, "On closure of the IFR class under formation of coherent systems," *IEEE Transactions on Reliability*, vol. 34, no. 1, pp. 69–72, 1985.

[80] I. B. Gertsbakh, Y. Shpungin, and R. Vaisman, "Reliability of a network with heterogeneous components," in *Recent Advances in Multi-state Systems Reliability* (A. Lisnianski, I. Frenkel, and A. Karagrigoriou, eds.), pp. 3–18, Springer, 2018.

[81] P. J. Boland and F. J. Samaniego, "Stochastic ordering results for consecutive k-out-of-n:F systems," *IEEE Transactions on Reliability*, vol. 53, no. 1, pp. 7–10, 2004.

[82] I. S. Triantafyllou and M. V. Koutras, "On the signature of coherent systems and applications," *Probability in the Engineering and Informational Sciences*, vol. 22, no. 1, pp. 19–35, 2008.

[83] S. Eryilmaz and J. Navarro, "Failure rates of consecutive k-out-of-n systems," *Journal of the Korean Statistical Society*, vol. 41, no. 1, pp. 1–11, 2012.

[84] S. Eryilmaz, "The number of failed components in a coherent system with exchangeable components," *IEEE Transactions on Reliability*, vol. 61, no. 1, pp. 203–207, 2012.

[85] S. Reed, M. Löfstrand, and J. Andrews, "An efficient algorithm for computing exact system and survival signatures of K-terminal network reliability," *Reliability Engineering and System Safety*, vol. 185, pp. 429–439, 2019.

[86] W. Kuo and X. Zhu, *Importance Measures in Reliability, Risk, and Optimization: Principles and Applications*. John Wiley and Sons, 2012.

[87] M. J. Zuo and W. Kuo, "Design and performance analysis of consecutive-k-out-of-n structure," *Naval Research Logistics*, vol. 37, no. 2, pp. 203–230, 1990.

[88] T. Hanafusa and H. Yamamoto, "Branch and bound algorithm for the optimal assignment problem in linear consecutive-k-out-of-n:F system," *Reliability Engineering Association of Japan (in Japanese)*, vol. 22, no. 7, pp. 641–661, 2000.

[89] K. Shingyochi and H. Yamamoto, "An efficient algorithm for solving optimal component arrangement problem in circular consecutive-k-out-of-n:F system," *Journal of Japan Industrial Management Association*, vol. 58, no. 6, pp. 423–429, 2008.

[90] M. V. Koutras, G. K. Papadopoulos, and S. G. Papastavridis, "Note: Pairwise rearrangements in reliability structures," *Naval Research Logistics*, vol. 41, no. 5, pp. 683–687, 1994.

[91] T. Omura, T. Akiba, H. Yamamoto, and X. Xiao, "Algorithm for obtaining optimal arrangement of a connected-(r, s)-out-of-(m, n):F system — the case of $m = r$ and $s = 2$ —," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E98-A, no. 10, pp. 2018–2024, 2015.

[92] D. M. Malon, "Optimal consecutive-2-out-of-n:F component sequencing," *IEEE Transactions on Reliability*, vol. 33, no. 5, pp. 414–418, 1984.

[93] T. Omura, H. Yamamoto, X. Xiao, and T. Akiba, "Algorithm for obtaining optimal arrangement of a connected-(r, s)-out-of-(m, n):F system — the case of $m = r + 1$ and $2s > n$ —," in *Proceedings of the 16th Asia Pacific Industrial Engineering and Management Systems Conference*, pp. 652–659, 2015.

[94] F. H. Lin and W. Kuo, "Reliability importance and invariant optimal allocation," *Journal of Heuristics*, vol. 8, no. 2, pp. 155–171, 2002.

[95] Q. Yao, X. Zhu, and W. Kuo, "Heuristics for component assignment problems based on the Birnbaum importance," *IIE Transactions*, vol. 43, no. 9, pp. 633–646, 2011.

[96] K. Shingyochi, H. Yamamoto, Y. Tsujimura, and Y. Kambayashi, "Genetic algorithm for solving optimal component arrangement problem of circular consecutive-k-out-of-n:F system," in *Proceedings of the 2nd Asian International Workshop*, pp. 176–184, 2006.

[97] K. Shingyochi and H. Yamamoto, "Efficient genetic algorithm for optimal arrangement in a linear consecutive-k-out-of-n:F system," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E92-A, no. 7, pp. 1578–1584, 2009.

[98] Z. Cai, S. Si, S. Sun, and C. Li, "Optimization of linear consecutive-k-out-of-n system with a Birnbaum importance-based genetic algorithm," *Reliability Engineering and System Safety*, vol. 152, pp. 248–258, 2016.

[99] S. Zhang, J. B. Zhao, W. J. Zhu, and L. Du, "Reliability optimization of linear consecutive k-out-of-n:F systems with Birnbaum importance-based quantum genetic algorithm," *Advances in Mechanical Engineering*, vol. 11, no. 4, pp. 1–16, 2019.

[100] K. Shingyochi, H. Yamamoto, Y. Tsujimura, and T. Akiba, "Proposal of simulated annealing algorithms for optimal arrangement in a circular consecutive-k-out-of-n:F system," *Quality Technology and Quantitative Management*, vol. 7, no. 4, pp. 395–405, 2010.

[101] K. Shingyochi, H. Yamamoto, and H. Yamachi, "Comparative study of several simulated annealing algorithms for optimal arrangement problems in a circular consecutive-k-out-of-n:F system," *Quality Technology and Quantitative Management*, vol. 9, no. 3, pp. 295–303, 2012.

[102] Z. Q. Cai, W. Wang, S. Zhang, and Z. Y. Jiang, "Ant colony optimization for component assignment problems in circular consecutive-k-out-of-n systems," in *Proceedings of the 2017 IEEE International Conference on Industrial Engineering and Engineering Management*, pp. 954–958, 2017.

[103] W. Wang, Z. Cai, J. Zhao, and S. Si, "Optimization of linear consecutive-k-out-of-n systems with Birnbaum importance based ant colony optimization algorithm," *Journal of Shanghai Jiaotong University (Science)*, pp. 1–8, 2019.

[104] X. Zhu, Y. Fu, T. Yuan, and X. Wu, "Birnbaum importance based heuristics for multi-type component assignment problems," *Reliability Engineering and System Safety*, vol. 165, pp. 209–221, 2017.

[105] I. I. H. Nashwan, "Failure function of the connected-$(1,2)$-or-$(2,1)$-of-(m,n):F linear & circular lattice system," in *Proceedings of the 2019 IEEE International Conference on Computer and Information Sciences*, pp. 1–6, 2019.

[106] D. Redekop, B. Xu, and Y. Zhang, "Stability of a toroidal fluid-containing shell," *International Journal of Pressure Vessels and Piping*, vol. 76, no. 9, pp. 575–581, 1999.

[107] H. S. Tzou and D. W. Wang, "Micro-sensing characteristics and modal voltages of linear/non-linear toroidal shells," *Journal of Sound and Vibration*, vol. 254, no. 2, pp. 203–218, 2002.

[108] S. Kochar, H. Mukerjee, and F. J. Samaniego, "The "signature" of a coherent system and its application to comparisons among systems," *Naval Research Logistics*, vol. 46, no. 5, pp. 507–523, 1999.

[109] G. Da, L. Xia, and T. Hu, "On computing signatures of k-out-of-n systems consisting of modules," *Methodology and Computing in Applied Probability*, vol. 16, no. 1, pp. 223–233, 2014.

[110] G. Da, P. S. Chan, and M. Xu, "On the signature of complex system: A decomposed approach," *European Journal of Operational Research*, vol. 265, no. 3, pp. 1115–1123, 2018.

[111] J. L. Marichal, P. Mathonet, and F. Spizzichino, "On modular decompositions of system signatures," *Journal of Multivariate Analysis*, vol. 134, pp. 19–32, 2015.

[112] X. Jia, J. Shen, F. Xu, R. Ma, and X. Song, "Modular decomposition signature for systems with sequential failure effect," *Reliability Engineering and System Safety*, vol. 189, pp. 435–444, 2019.

[113] P. J. Boland, F. J. Samaniego, and E. M. Vestrup, "Linking dominations and signatures in network reliability theory," in *Mathematical and Statistical Methods in Reliability* (K. A. Doksum and B. Lindqvist, eds.), ch. 7, pp. 89–103, World Scientific, 2003.

[114] G. Da, M. Xu, and P. S. Chan, "An efficient algorithm for computing the signatures of systems with exchangeable components and applications," *IISE Transactions*, vol. 50, no. 7, pp. 584–595, 2018.

[115] P. J. Boland, "Signatures of indirect majority systems," *Journal of Applied Probability*, vol. 38, no. 2, pp. 597–603, 2001.

[116] S. Eryilmaz, "Mixture representations for the reliability of consecutive-k systems," *Mathematical and Computer Modelling*, vol. 51, no. 5-6, pp. 405–412, 2010.

[117] S. Eryilmaz and A. Tuncel, "Computing the signature of a generalized k-out-of-n system," *IEEE Transactions on Reliability*, vol. 64, no. 2, pp. 766–771, 2015.

[118] E. M. El-Sayed, "Algorithm for reliability of $(1, 2)$ or $(2, 1)$-out-of $(n, 2)$:F systems," *Journal of the Egyptian Mathematical Society*, vol. 6, no. 2, pp. 169–173, 1998.

[119] T. Ishikawa, T. Shinzato, T. Nakamura, and H. Yamamoto, "Expansion on ratio $\frac{q}{p}$ for reliability function of linear connected-$(1, 2)$-or-$(2, 1)$-out-of-(m, n):F lattice system," in *Proceedings of the 7th Asia-Pacific International Symposium on Advanced Reliability and Maintenance Modeling*, pp. 162–169, 2016.

[120] M. Shaked and J. G. Shanthikumar, *Stochastic Orders*. Springer Science and Business Media, 2007.

[121] M. A. Arcones, P. H. Kvam, and F. J. Samaniego, "Nonparametric estimation of a distribution subject to a stochastic precedence constraint," *Journal of the American Statistical Association*, vol. 97, no. 457, pp. 170–182, 2002.

[122] J. Navarro, F. J. Samaniego, N. Balakrishnan, and D. Bhattacharya, "On the application and extension of system signatures in engineering reliability," *Naval Research Logistics*, vol. 55, no. 4, pp. 313–327, 2008.

[123] M. Hollander and F. J. Samaniego, "The use of stochastic precedence in the comparison of engineered systems," in *Advances in Mathematical Modeling for Reliability* (T. Bedford, J. Quigley, and L. Walls, eds.), pp. 129–137, 2008.

[124] P. J. Boland, F. Proschan, and Y. L. Tong, "Optimal arrangement of components via pairwise rearrangements," *Naval Research Logistics*, vol. 36, no. 6, pp. 807–815, 1989.

[125] F. C. Meng, "Comparing criticality of nodes via minimal cut (path) sets for coherent systems," *Probability in the Engineering and Informational Sciences*, vol. 8, no. 1, pp. 79–87, 1994.

[126] H. Yamamoto, A. Suzuki, and T. Nakamura, "A branch and bound algorithm for solving facility layout problems using symmetries," *Journal of Japan Industrial Management Association (in Japanese)*, vol. 55, no. 5, pp. 293–301, 2004.

[127] F. P. Coolen and T. Coolen-Maturi, "Generalizing the signature to systems with multiple types of components," in *Complex Systems and Dependability* (W. Zamojski, J. Mazurkiewicz, J. Sugier, T. Walkowiak, and J. Kacprzyk, eds.), ch. 8, pp. 115–130, Springer, 2013.

[128] S. Eryilmaz, "Reliability of systems with multiple types of dependent components," *IEEE Transactions on Reliability*, vol. 65, no. 2, pp. 1022–1029, 2016.

[129] H. Wang and H. Pham, "Optimal imperfect maintenance models," in *Handbook of Reliability Engineering* (H. Pham, ed.), ch. 21, pp. 397–414, Springer, 2003.

[130] S. Eryilmaz, "The number of failed components in a k-out-of-n system consisting of multiple types of components," *Reliability Engineering and System Safety*, vol. 175, pp. 246–250, 2018.

[131] M. Hashemi, M. Asadi, and S. Zarezadeh, "Optimal maintenance policies for coherent systems with multi-type components," *Reliability Engineering and System Safety*, vol. 195, pp. 1–11, 2020.

[132] S. Zarezadeh and M. Asadi, "Coherent systems subject to multiple shocks with applications to preventative maintenance," *Reliability Engineering and System Safety*, vol. 185, pp. 124–132, 2019.

Publications

Journal Papers

[1] T. Nakamura, H. Yamamoto, and T. Akiba, "Reliability of a toroidal connected-(r, s)-out-of-(m, n):F lattice system," *Journal of Risk and Reliability*, 10 pages, 2020 (Early Access).

[2] T. Nakamura and H. Yamamoto, "Algorithm for solving optimal arrangement problem in connected-(r, s)-out-of-(m, n):F lattice system," *IEEE Transactions on Reliability*, 13 pages, 2019 (Early Access).

[3] T. Nakamura, H. Yamamoto, and T. Akiba, "Fast algorithm for optimal arrangement in connected-$(m-1, n-1)$-out-of-(m, n):F lattice system," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E101-A, no. 12, pp. 2446–2453, 2018.

[4] T. Nakamura, H. Yamamoto, and X. Xiao, "Fast calculation methods for reliability of connected-(r, s)-out-of-(m, n):F lattice system in special cases," *International Journal of Mathematical, Engineering and Management Sciences*, vol. 3, no. 2, pp. 113–122, 2018.

[5] T. Nakamura, H. Yamamoto, T. Shinzato, X. Xiao, and T. Akiba, "Reliability of a circular connected-$(1, 2)$-or-$(2, 1)$-out-of-(m, n):F lattice system with identical components," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E100-A, no. 4, pp. 1029–1036, 2017.

International Conference Papers

[1] T. Nakamura and H. Yamamoto, "Recursive algorithm for the reliability of the toroidal connected-(r, s)-out-of-(m, n):F lattice system," in *Proceedings of the 3rd International Symposium on Stochastic Models in Reliability Engineering, Life Sciences and Operations Management*, 2019.

[2] T. Nakamura, H. Yamamoto, T. Shinzato, and T. Akiba, "Upper and lower bounds for reliability of connected-$(1, 2)$-or-$(2, 1)$-out-of-(m, n):F lattice system," in *Proceedings of the 19th Asia Pacific Industrial Engineering and Management Systems*, 6 pages, 2018.

[3] T. Nakamura, H. Yamamoto, X. Xiao, and T. Akiba, "The increasing failure rate property of connected-$(1, 2)$-or-$(2, 1)$-out-of-(m, n):F lattice systems — the Case of $m = 2, 3, 4$ —," in *Proceedings of the 8th Asia-Pacific International Symposium on Advanced Reliability and Maintenance Modeling*, pp. 45–51, 2018.

[4] T. Nakamura, H. Yamamoto, X. Xiao, T. Shinzato, and T. Akiba, "A computing method for system signatures of a connected-(r, s)-out-of-(m, n):F lattice system," in *Proceedings of the 18th Asia Pacific Industrial Engineering and Management Systems*, 6 pages, 2017.

[5] T. Nakamura, H. Yamamoto, X. Xiao, N. Takahashi, and T. Akiba, "Necessary conditions for optimal arrangement of connected-(r, s)-out-of-(m, n):F lattice system with minimal cuts overlapping," in *Proceedings of the 10th International Conference on Mathematical Methods in Reliability*, 8 pages, 2017.

[6] T. Nakamura, H. Yamamoto, T. Shinzato, X. Xiao, and T. Akiba, "Approximations for reliability of the connected-$(1, 2)$-or-$(2, 1)$-out-of-(m, n):F lattice system," in *Proceedings of the 3rd East Asia Workshop on Industrial Engineering*, 7 pages, 2016.

[7] T. Nakamura, H. Yamamoto, X. Xiao, and T. Akiba, "Reliability of a toroidal connected-$(1, 2)$-or-$(2, 1)$-out-of-(m, n):F lattice system," in *Proceedings of the 7th Asia-Pacific International Symposium on Advanced Reliability and Maintenance Modeling*, pp. 399–406, 2016.

[8] T. Nakamura, T. Akiba, X. Xiao, H. Yamamoto, and K. Shingyochi, "A study on the reliability of a connected-(r, s)-out-of-(m, n):F lattice system," in *Proceedings of the 16th Asia Pacific Industrial Engineering and Management Systems Conference*, pp. 1936–1943, 2015.

Book Chapters

[1] T. Akiba, T. Nakamura, X. Xiao, and H. Yamamoto, "Evaluation methods for reliability of consecutive-k systems," in *Systems Engineering: Reliability Analysis Using k-out-of-n Structures* (M. Ram and T. Dohi, eds.), ch. 1, pp. 1–24, CRC Press, 2019.

[2] T. Nakamura, H. Yamamoto, T. Shinzato, X. Xiao, and T. Akiba, "Proposal of calculation method for reliability of toroidal connected-$(1, 2)$-or-$(2, 1)$-out-of-(m, n):F lattice system with Markov chain," in *Reliability Modeling With Computer and Maintenance Applications* (S. Nakamura, C. H. Qian, and T. Nakagawa, eds.), ch. 7, pp. 103–137, World Scientific, 2017.

Acknowledgments

First, I would like to thank my supervisor, Prof. Yamamoto. Without his continued support, invaluable suggestions, and guidance, I could not have overcome many obstacles, and then this thesis could not have been accomplished. I appreciate his feedback and valuable comments during the writing process. He also gave me several opportunities to present my work at conferences and visit the University of Alberta and so on. I believe that these experiences will greatly aid my professional development. I would like to say thank you to Prof. Yamamoto for everything he has done for me during the past years. And, I will continue benefiting from what I have learned from him throughout my whole career.

I also would like to thank the members of my thesis committee, Prof. Aida, Prof. Masuda, and Prof. Yuge, for their helpful feedback and guidance during my committee meetings. Their thorough review of my drafts has improved this thesis so much. I am thankful, in particular, to Prof. Yuge for making an effort to come from the National Defense Academy of Japan to attend my oral examinations despite his busy schedule.

I would also like to thank Dr. Xiao for reading my thesis and for her comments. In particular, her valuable comments definitely have improved my abstract of this thesis greatly. She also gave me basic and important things in research while I was in the Yamamoto Laboratory.

I would like to thank Prof. Akiba, who has advised me a lot in my research. Mainly, I have learned programming from him, which was essential for completing this thesis. Additionally, when I met difficulties, he always encouraged and helped me with his experience and knowledge.

I would like to thank Prof. Shingyochi, Dr. Shinzato, and Dr. Takahashi, who dedicated their precious time to discuss my research topic and provide exciting comments and suggestions. I am also thankful to the other members of the Reliability Study Group.

Furthermore, I would like to express my gratitude to the academics of the Department of Management Systems Engineering for their assistance during my study.

I would like to thank Prof. Zuo, who allowed me to visit the University of Alberta. Also, I am grateful to Prof. Zuo for training me to make presentations and write research papers in English, which helped me significantly in finishing this thesis. Although my stay in Edmonton was short (three months), these experiences are quite valuable.

I am grateful for the opportunity to study with members of the Yamamoto Laboratory. Especially, I want to say thanks to all the people who have checked my thesis to eliminate text mistakes and typos before submitting it. I also want to thank everyone who helped me directly or indirectly in preparing the research contained in this thesis.

Finally, I would like to thank my family for their understanding and support.

This research was supported by Grant-in-Aid for JSPS Research Fellow Grant Number 18J12465.