7

THE *k*-OUT-OF-*n* SYSTEM MODEL

An n-component system that *works* (or is "good") if and only if at least k of the n components work (or are good) is called a *k-out-of-n:G* system. An n-component system that *fails* if and only if at least k of the n components fail is called a *k-out-of-n:F* system. Based on these two definitions, a k-out-of-n:G system is equivalent to an $(n - k + 1)$-out-of-n:F system. The term k-out-of-n system is often used to indicate either a G system or an F system or both. Since the value of n is usually larger than the value of k, redundancy is generally built into a k-out-of-n system. Both parallel and series systems are special cases of the k-out-of-n system. A series system is equivalent to a 1-out-of-n:F system and to an n-out-of-n:G system while a parallel system is equivalent to an n-out-of-n:F system and to a 1-out-of-n:G system.

The k-out-of-n system structure is a very popular type of redundancy in *fault-tolerant* systems. It finds wide applications in both industrial and military systems. Fault-tolerant systems include the multidisplay system in a cockpit, the multiengine system in an airplane, and the multipump system in a hydraulic control system. For example, it may be possible to drive a car with a V8 engine if only four cylinders are firing. However, if less than four cylinders fire, then the automobile cannot be driven. Thus, the functioning of the engine may be represented by a 4-out-of-8:G system. The system is tolerant of failures of up to four cylinders for minimal functioning of the engine. In a data processing system with five video displays, a minimum of three displays operable may be sufficient for full data display. In this case the display subsystem behaves as a 3-out-of-5:G system. In a communications system with three transmitters, the average message load may be such that at least two transmitters must be operational at all times or critical messages may be lost. Thus, the transmis-

sion subsystem functions as a 2-out-of-3:G system. Systems with spares may also be represented by the k-out-of-n system model. In the case of an automobile with four tires, for example, usually one additional spare tire is equipped on the vehicle. Thus, the vehicle can be driven as long as at least 4-out-of-5 tires are in good condition.

Among applications of the k-out-of-n system model, the design of electronic circuits such as very large scale integrated (VLSI) and the automatic repairs of faults in an on-line system would be the most conspicuous. This type of system demonstrates what is called the *voting redundancy*. In such a system, several parallel outputs are channeled through a decision-making device that provides the required system function as long as at least a predetermined number k of n parallel outputs are in agreement.

In this chapter, we provide a detailed coverage on reliability evaluation of the k-out-of-n systems. Methods for finding both the exact and the approximate system reliability values are introduced. The performance measures of both nonrepairable and repairable k-out-of-n systems are addressed. In addition, the weighted k-out-of-n system model is discussed in this chapter. In our discussions, it is assumed that the working of the components is independent of one another unless otherwise specified.

7.1 SYSTEM RELIABILITY EVALUATION

In this section, we concentrate on techniques for reliability evaluation of k-out-of-n:G systems. The k-out-of-n:G system with i.i.d. components is first studied. Several approaches for system reliability evaluation, when the components are not necessarily s-identical, are then introduced in detail. Finally, bounds on system reliability, when components are not necessarily s-independent, are discussed.

Notation

- n: number of components in the system
- k: minimum number of components that must work for the k-out-of-n:G system to work
- p_i: reliability of component i, $i = 1, 2, \ldots, n$
- p: reliability of each component when all components are i.i.d.
- q_i: unreliability of component i, $q_i = 1 - p_i$, $i = 1, 2, \ldots, n$
- q: unreliability of each component when all components are i.i.d., $q = 1 - p$
- $R_e(k, n)$: probability that exactly k out of n components are working
- $R(k, n)$: reliability of a k-out-of-n:G system or probability that at least k out of the n components are working, where $0 \leq k \leq n$ and both k and n are integers
- $Q(k, n)$: unreliability of a k-out-of-n:G system or probability that less than k out of the n components are working, where $0 \leq k \leq n$ and both k and n are integers, $Q(k, n) = 1 - R(k, n)$

7.1.1 The k-out-of-n:G System with i.i.d. Components

In a k-out-of-n:G system with i.i.d. components, the number of working components follows the binomial distribution with parameters n and p. Thus, we have

$$\Pr(\text{exactly } i \text{ components work}) = \binom{n}{i} p^i q^{n-i}, \qquad i = 0, 1, 2, \ldots, n. \tag{7.1}$$

The reliability of the system is equal to the probability that the number of working components is greater than or equal to k:

$$R(k, n) = \sum_{i=k}^{n} \binom{n}{i} p^i q^{n-i}. \tag{7.2}$$

Equation (7.2) is an explicit formula that can be used for reliability evaluation of the k-out-of-n:G system.

If we apply the pivotal decomposition to component n or directly use equation (7.26) developed by Rushdi [208], the system reliability of a k-out-of-n:G system with i.i.d. components can be expressed as

$$\begin{aligned} R(k, n) &= pR(k-1, n-1) + (1-p)R(k, n-1) \\ &= p(R(k-1, n-1) - R(k, n-1)) + R(k, n-1) \\ &= p \Pr(\text{exactly } k-1 \text{ out of } n-1 \text{ components work}) + R(k, n-1) \\ &= \binom{n-1}{k-1} p^k q^{n-k} + R(k, n-1). \end{aligned} \tag{7.3}$$

Rearranging the terms in equation (7.3), we obtain the expression

$$R(k, n) - R(k, n-1) = \binom{n-1}{k-1} p^k q^{n-k} \qquad \text{for } n \geq k. \tag{7.4}$$

Equation (7.4) represents the improvement in system reliability by increasing the number of components in the system from $n-1$ to n. As n increases, this improvement amount in system reliability will become smaller. Thus, there is an optimal design issue of determining the system size n, which will be addressed later.

Equation (7.3) can be used recursively for system reliability evaluation with the boundary condition

$$R(k, n) = 0 \qquad \text{for } n < k. \tag{7.5}$$

Using equation (7.4) and the boundary condition given in equation (7.5), we can express the reliability of a k-out-of-n:G system as follows:

$$R(k, n) = \sum_{i=k}^{n} [R(k, i) - R(k, i-1)] = p^k \sum_{i=k}^{n} \binom{i-1}{k-1} q^{i-k}. \tag{7.6}$$

From the equations for system reliability given above, we can see that the reliability of a k-out-of-n:G system with i.i.d. components is a function of n, k, and p. An increase in n or p or both or a decrease in k will increase the system's reliability. Equation (7.4) represents the increase in system reliability by increasing the number of components in the system from $n-1$ to n. In the following, we give an expression for the increase in system reliability for each unit of decrease in k:

$$\begin{aligned} R(k,n) &= \Pr(\text{at least } k \text{ components work}) \\ &= \Pr(\text{at least } k-1 \text{ components work}) \\ &\quad - \Pr(\text{exactly } k-1 \text{ components work}) \\ &= R(k-1,n) - \binom{n}{k-1} p^{k-1} q^{n-k+1}. \end{aligned} \tag{7.7}$$

Or equivalently, we have

$$R(k-1,n) - R(k,n) = \binom{n}{k-1} p^{k-1} q^{n-k+1}. \tag{7.8}$$

With the various expressions of $R(k,n)$ derived so far, we can easily find the expressions of the unreliability $Q(k,n)$ of the k-out-of-n:G system. For example, the following is obvious from equation (7.2):

$$Q(k,n) = 1 - R(k,n) = \sum_{i=0}^{k-1} \binom{n}{i} p^i q^{n-i}. \tag{7.9}$$

To find the expression for the sensitivity of system reliability on component reliability in this i.i.d. case, we can take the first derivative of $R(k,n)$ with respect to p. Using equation (7.6), we have

$$\frac{dR(k,n)}{dp} = k\binom{n}{k} p^{k-1} q^{n-k}. \tag{7.10}$$

Exercises

1. Verify equation (7.10).
2. Find similar expressions of $R(k,n)$ or $Q(k,n)$ and other measures for the k-out-of-n:F systems.
3. Analyze the performance of a 3-out-of-6:G system with $p_1 = 0.5$, $p_2 = 0.6$, $p_3 = 0.7$, $p_4 = 0.8$, $p_5 = 0.9$, and $p_6 = 0.95$.

7.1.2 The k-out-of-n:G System with Independent Components

For k-out-of-n:G systems with components whose reliabilities are not necessarily identical, we can use the concept of minimal path sets to evaluate system reliability. However, more efficient algorithms for reliability evaluation of such systems were

reported by Barlow and Heidtmann [20] and Rushdi [208]. These two algorithms have the same complexity as $O(k(n-k+1))$. The use of Markov chain imbeddable structures confirms the same result. Belfore [26] uses fast Fourier transform (FFT) and proposes an $O(n(\log_2 n)^2)$ algorithm for reliability evaluation of k-out-of-n:G systems. In this section, we illustrate the use of minimal path sets in system reliability evaluation. In addition, we illustrate these other approaches to deriving efficient algorithms for reliability evaluation of k-out-of-n:G systems.

Minimal Path Sets or Minimal Cut Sets Approach As discussed earlier, the reliability of any system is equal to the probability that at least one of the minimal path sets works. The unreliability of the system is equal to the probability that at least one minimal cut set is failed. For a minimal path set to work, each component in the set must work. For a minimal cut set to fail, all components in the set must fail. In a k-out-of-n:G system, there are $\binom{n}{k}$ minimal path sets and $\binom{n}{n-k+1}$ minimal cut sets. Each minimal path set contains exactly k different components and each minimal cut set contains exactly $n-k+1$ components. Thus, all minimal path sets and minimal cut sets are known. The question remaining to be answered is how to find the probability that at least one of the minimal path sets contains all working components or the probability that at least one minimal cut set contains all failed components.

The IE method can be used for reliability evaluation of a k-out-of-n:G system since all the minimal path sets and minimal cut sets are known. The IE method has the disadvantage of involving many canceling terms. Heidtmann [92] and McGrady [165] provide improved versions of the IE method for reliability evaluation of the k-out-of-n:G system. In their improved algorithms, the canceling terms are eliminated. However, both algorithms are still enumerative in nature. For example, the formula provided by Heidtmann [92] using minimal path sets is as follows:

$$R(k,n) = \sum_{i=k}^{n} (-1)^{i-k} \binom{i-1}{k-1} \sum_{j_1<j_2<\cdots<j_i} \prod_{\ell=1}^{i} p_{j_\ell}. \tag{7.11}$$

In this equation, for each fixed i value, the inner summation term gives us the probability that i components are working properly regardless of whether the other $n-i$ components are working or not. The total number of terms to be summed together in the inner summation series is equal to $\binom{n}{i}$. If all the components are i.i.d., equation (7.11) gives another formula for reliability evaluation of a k-out-of-n:G system with i.i.d. components:

$$R(k,n) = \sum_{i=k}^{n} \binom{n}{i} \binom{i-1}{k-1} (-1)^{i-k} p^i. \tag{7.12}$$

Equation (7.12) is apparently not as efficient as those given in Section 7.1.1.

The SDP method can also be used for reliability evaluation of the k-out-of-n:G systems. Like the improved IE method given in equation (7.11), the SDP method is also easy to use for the k-out-of-n:G systems. However, we will see later that there are much more efficient methods than the IE (and its improved version) and the SDP

method for evaluating k-out-of-n:G systems. In the following, we present an example to illustrate the use of minimal path sets with the IE method, Heidtmann's improved IE method, and the SDP method for reliability evaluation of a 2-out-of-4:G system.

Example 7.1 Evaluate the reliability of a 2-out-of-4:G system with $p_1 = 0.91$, $p_2 = 0.92$, $p_3 = 0.93$, and $p_4 = 0.94$. The number of minimal path sets is equal to $\binom{4}{2} = 6$. We will use S_i to represent the ith minimal path set as listed below:

$$S_1 = x_1x_2, \qquad S_2 = x_1x_3, \qquad S_3 = x_1x_4,$$
$$S_4 = x_2x_3, \qquad S_5 = x_2x_4, \qquad S_6 = x_3x_4.$$

With the IE method, we can calculate system reliability as follows:

$$\begin{aligned}
R(2,4) &= \Pr(S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5 \cup S_6) \\
&= \Pr(S_1) + \Pr(S_2) + \Pr(S_3) + \Pr(S_4) + \Pr(S_5) + \Pr(S_6) \\
&\quad - \Pr(S_1S_2) - \Pr(S_1S_3) - \Pr(S_1S_4) - \Pr(S_1S_5) - \Pr(S_1S_6) - \Pr(S_2S_3) \\
&\quad - \Pr(S_2S_4) - \Pr(S_2S_5) - \Pr(S_2S_6) - \Pr(S_3S_4) - \Pr(S_3S_5) - \Pr(S_3S_6) \\
&\quad - \Pr(S_4S_5) - \Pr(S_4S_6) - \Pr(S_5S_6) + \Pr(S_1S_2S_3) + \Pr(S_1S_2S_4) \\
&\quad + \Pr(S_1S_2S_5) + \Pr(S_1S_2S_6) + \Pr(S_1S_3S_4) + \Pr(S_1S_3S_5) \\
&\quad + \Pr(S_1S_3S_6) + \Pr(S_1S_4S_5) + \Pr(S_1S_4S_6) + \Pr(S_1S_5S_6) \\
&\quad + \Pr(S_2S_3S_4) + \Pr(S_2S_3S_5) + \Pr(S_2S_3S_6) + \Pr(S_2S_4S_5) \\
&\quad + \Pr(S_2S_4S_6) + \Pr(S_2S_5S_6) + \Pr(S_3S_4S_5) + \Pr(S_3S_4S_6) \\
&\quad + \Pr(S_3S_5S_6) + \Pr(S_4S_5S_6) - \Pr(S_1S_2S_3S_4) - \Pr(S_1S_2S_3S_5) \\
&\quad - \Pr(S_1S_2S_3S_6) - \Pr(S_1S_2S_4S_5) - \Pr(S_1S_2S_4S_6) - \Pr(S_1S_2S_5S_6) \\
&\quad - \Pr(S_1S_3S_4S_5) - \Pr(S_1S_3S_4S_6) - \Pr(S_1S_3S_5S_6) - \Pr(S_1S_4S_5S_6) \\
&\quad - \Pr(S_2S_3S_4S_5) - \Pr(S_2S_3S_4S_6) - \Pr(S_2S_3S_5S_6) - \Pr(S_2S_4S_5S_6) \\
&\quad - \Pr(S_3S_4S_5S_6) + \Pr(S_1S_2S_3S_4S_5) + \Pr(S_1S_2S_3S_4S_6) \\
&\quad + \Pr(S_1S_2S_3S_5S_6) + \Pr(S_1S_2S_4S_5S_6) + \Pr(S_1S_3S_4S_5S_6) \\
&\quad + \Pr(S_2S_3S_4S_5S_6) - \Pr(S_1S_2S_3S_4S_5S_6) \\
&\approx 0.998467.
\end{aligned}$$

With equation (7.11), we have

$$\begin{aligned}
R(2,4) &= \sum_{i=2}^{4} (-1)^{i-2} \binom{i-1}{1} \sum_{j_1<j_2<\cdots<j_i} \prod_{\ell=1}^{i} p_{j_\ell} \\
&= (p_1p_2 + p_1p_3 + p_1p_4 + p_2p_3 + p_2p_4 + p_3p_4)
\end{aligned}$$

$$- 2(p_1p_2p_3 + p_1p_2p_4 + p_1p_3p_4 + p_2p_3p_4) + 3p_1p_2p_3p_4$$
$$\approx 0.998441.$$

With the SDP method, we have

$$\begin{aligned} R(2,4) &= \Pr(S_1 \cup S_2 \cup S_3 \cup S_4 \cup S_5 \cup S_6) \\ &= \Pr(S_1) + \Pr(\overline{S}_1 S_2) + \Pr(\overline{S}_1 \overline{S}_2 S_3) + \Pr(\overline{S}_1 \overline{S}_2 \overline{S}_3 S_4) \\ &\quad + \Pr(\overline{S}_1 \overline{S}_2 \overline{S}_3 \overline{S}_4 S_5) + \Pr(\overline{S}_1 \overline{S}_2 \overline{S}_3 \overline{S}_4 \overline{S}_5 S_6) \\ &= \Pr(x_1x_2) + \Pr(x_1\overline{x}_2x_3) + \Pr(x_1\overline{x}_2\,\overline{x}_3x_4) + \Pr(\overline{x}_1x_2x_3) \\ &\quad + \Pr(\overline{x}_1x_2\overline{x}_3x_4) + \Pr(\overline{x}_1\,\overline{x}_2x_3x_4) \\ &\approx 0.998441. \end{aligned}$$

It is clear that the IE method involves much more calculation than either the improved IE method or the SDP method. Because there are many canceling terms in the IE method, the round-off errors are obvious in its final result.

Generating Function Approach by Barlow and Heidtmann Barlow and Heidtmann [20] present two BASIC programs for reliability evaluation of k-out-of-n:G systems with independent components. The first program uses the following generating function and its expanded form:

$$g_n(z) = \prod_{i=1}^{n}(q_i + p_i z) = \sum_{i=0}^{n} R_e(i,n)z^i, \tag{7.13}$$

where z is a dummy variable. As we have defined in the notation, $R_e(i,n)$ represents the probability that there are exactly i working components in the system. Through examination of the expanded form of $g_n(z)$, we find that $R_e(i,n)$ also represents the coefficient of z^i in the generating function. The BASIC program computes all $R_e(i,j)$ entries recursively. Rushdi [208] provides better explanations of this algorithm. In fact, the algorithm relies on the equation

$$R(k,n) = \sum_{i=k}^{n} R_e(i,n), \tag{7.14}$$

which is obvious based on the definition of a k-out-of-n:G system. The algorithm obtains $R_e(i,n)$ through the recursive relation

$$R_e(i,j) = q_j R_e(i,j-1) + p_j R_e(i-1,j-1), \qquad 0 \le i \le n, \qquad 0 \le j \le n, \tag{7.15}$$

with the boundary conditions

$$R_e(-1, j) = R_e(j+1, j) = 0 \qquad \text{for } j = 0, 1, 2, \ldots . \tag{7.16}$$

$$R_e(0, 0) = 1. \tag{7.17}$$

To derive this recursive relation, first construct the following generating function:

$$g_{j-1}(z) = \prod_{i=1}^{j-1} (q_i + p_i z) = \sum_{i=0}^{j-1} R_e(i, j-1) z^i. \tag{7.18}$$

Since $g_j(z) = (q_j + p_j z) g_{j-1}(z)$, a comparison of the coefficients of z^i in both sides of the equation

$$\begin{aligned} \sum_{i=0}^{j} R_e(i, j) z^i &= (q_j + p_j z) \sum_{i=0}^{j-1} R_e(i, j-1) z^i \\ &= \sum_{i=0}^{j} \left[q_j R_e(i, j-1) + p_j R_e(i-1, j-1) \right] z^i \end{aligned} \tag{7.19}$$

leads to equation (7.15).

To find out the computational complexity of this algorithm, we examine the number of entries, $R_e(i, j)$, that should be calculated with equation (7.15) utilizing boundary conditions in equations (7.16) and (7.17). As shown in Figure 7.1, the total number of entries is equal to

$$(n-k+1)(k+1) - 1 + \tfrac{1}{2}(n-k)^2.$$

Each such entry requires three basic arithmetic operations (two multiplications and one addition). We then need to use equation (7.14) to find the system reliability, which requires $n-k$ basic arithmetic operations. As a result, the total number of basic arithmetic operations required is equal to

$$\begin{aligned} &3\left[(n-k+1)(k+1) - 1 + \tfrac{1}{2}(n-k)^2 \right] + n - k \\ &\quad = (n-k)(1.5n + 1.5k + 4) + 3k. \end{aligned}$$

From this expression, we can see that the computational complexity of the algorithm is $O(n^2)$ when k is small (close to 1) and $O(n)$ when k is large (close to n). Generally speaking, the complexity of this algorithm is $O(n^2)$.

The number of arithmetic operations required for system reliability evaluation can be reduced by noting that we are only interested in finding the probability that at least k components are working. Thus, the calculation of $R_e(i, j)$ can be avoided. The second BASIC program by Barlow and Heidtmann [20] avoids calculating $R_e(i, j)$ and requires only $3k(n-k+1)$ arithmetic operations. This computational complexity is also achieved by the algorithm proposed by Rushdi [208]. We will present Rushdi's algorithm in the following section.

i \ j	0	1	2	3	...	k	k+1	...	n-k	n-k+1	n-k+2	...	...	...	n
-1	0	0	0	0	...	0	0	...	0						
0	1	(0,1)	(0,2)	(0,3)	...	(0,k)	(0,k+1)	...	(0,n-k)	(0,n-k+1)					
1	0	(1,1)	(1,2)	(1,3)	...	(1,k)	(1,k+1)	...	(1,n-k)	(1,n-k+1)	(1,n-k+2)				
2		0	(2,2)	(2,3)	...	(2,k)	(2,k+1)	...	(2,n-k)	(2,n-k+1)	(2,n-k+2)	...			
...			0	(3,3)	...	...	...	...	...	...	...	...	...		
...				0	...	...	...	...	...	...	...	...	...	...	
k						(k,k)	(k,k+1)	...	(k,n-k)	(k,n-k+1)	(k,n-k+2)	...	...	...	(k,n)
k+1							(k+1,k+1)	...	(k+1,n-k)	...	...	...	...	...	(k+1,n)
...								...	...	...	...	...	...	...	...
n-k									(n-k,n-k)	...	...	...	...	...	(n-k,n)
n-k+1									0	(n-k+1,n-k+1)	...	...	...	...	(n-k+1,n)
n-k+2										0	...	...	...	...	(n-k+2,n)
...											0	...	...	...	...
...												0	...	...	...
...													0	...	...
n														0	(n,n)

FIGURE 7.1 The $R_e(i, j)$ entries to be calculated with the algorithm by Barlow and Heidtmann.

Exercises

1. Consider a system with $n = 5$ components. Verify that the coefficient of z^i for $i = 0, 1, \ldots, 5$ does represent the probability that there are exactly i working components in the system.
2. Compute the reliability and unreliability of a 3-out-of-8:G system with the algorithm given in this section.
3. Use the generating function approach to derive a similar algorithm for the k-out-of-n:F system.

Symmetric Switching Function Approach by Rushdi This approach starts with an analysis of the structure function of the k-out-of-n:G system. The structure function $\phi(\mathbf{x})$ of a k-out-of-n:G system is symmetric based on Definition 4.4. It can only take two possible values (0 or 1) under the binary assumption of component and system states, like an on–off switch. This is why we name this approach the symmetric switching function approach.

In this section, x_i indicates the state of component i and $S(k, n)$, instead of $\phi(\mathbf{x})$, indicates the structure function of the system. Both x_i and $S(k, n)$ are binary variables with a value of 1 indicating the working state and 0 indicating the failed state. The complements of these variables are represented by $\overline{x}_i$ and $\overline{S}(k, n)$, respectively. Based on these definitions of $S(k, n)$ and $\overline{S}(k, n)$, we have the following expressions for system reliability and system unreliability:

$$R(k, n) = \Pr(S(k, n) = 1), \qquad Q(k, n) = \Pr(\overline{S}(k, n) = 1).$$

To find an expression of the system state, we can use pivotal decomposition on the nth component, as shown below:

$$S(k, n) = x_n S(k - 1, n - 1) + \overline{x}_n S(k, n - 1), \tag{7.20}$$

$$\overline{S}(k, n) = x_n \overline{S}(k - 1, n - 1) + \overline{x}_n \overline{S}(k, n - 1). \tag{7.21}$$

Based on these two equations, the state of a k-out-of-n:G system can be expressed as a function of the states of two subsystems with the same $n-1$ components. However, the minimum numbers of components required for these two subsystems to work are different. One requires at least k components to work while the other requires at least $k - 1$ components to work. These two subsystems with $n - 1$ components can be further decomposed on the last component, namely component $n - 1$, until we reach some boundary conditions. Thus, an iterative expression can be used to describe this decomposition process. Consider a system with j components that requires at least i components to work for the system to work. We have the following equations to express the state of such a system as a function of the states of two subsystems:

$$S(i, j) = x_j S(i - 1, j - 1) + \overline{x}_j S(i, j - 1), \tag{7.22}$$

$$\overline{S}(i, j) = x_j \overline{S}(i - 1, j - 1) + \overline{x}_j \overline{S}(i, j - 1), \tag{7.23}$$

where i may take any integer value from 1 to k and j may take values from 0 to n. The following boundary conditions are needed for equations (7.22) and (7.23):

$$S(0, j) = \overline{S}(j + 1, j) = 1, \tag{7.24}$$

$$S(j + 1, j) = \overline{S}(0, j) = 0. \tag{7.25}$$

Equations (7.22) and (7.23) are in pivotal decomposition form. Because of the assumption that the components are s-independent, they can be immediately converted to the following algebraic reliability expressions:

$$R(i, j) = p_j R(i - 1, j - 1) + q_j R(i, j - 1), \tag{7.26}$$

$$Q(i, j) = p_j Q(i - 1, j - 1) + q_j Q(i, j - 1). \tag{7.27}$$

Equations (7.26) and (7.27) are recursive relations that are valid for $1 \leq i \leq k$. Their boundary conditions can be directly obtained from equations (7.24) and (7.25) as follows:

$$R(0, j) = Q(j + 1, j) = 1, \tag{7.28}$$

$$R(j + 1, j) = Q(0, j) = 0. \tag{7.29}$$

Solutions for the reliability $R(k, n)$ or the unreliability $Q(k, n)$ is easily obtained by programming in languages that allow a program to call itself recursively. However, a closer look at the recursive relations (7.26) and (7.27) reveals that they can be easily represented by what is called a signal flow graph (SFG). As an illustration, Figure 7.2 shows the SFG for the computation of $R(3, 7)$. In Figure 7.2, a node at position (i, j) represents $R(i, j)$. The black nodes in the first row with $i = 0$ are "source" nodes with values of 1, that is, $R(0, j) = 1$. The white nodes at $i = j + 1$ are source nodes with zero values, that is, $R(j + 1, j) = 0$ for $j \geq 0$. The values at

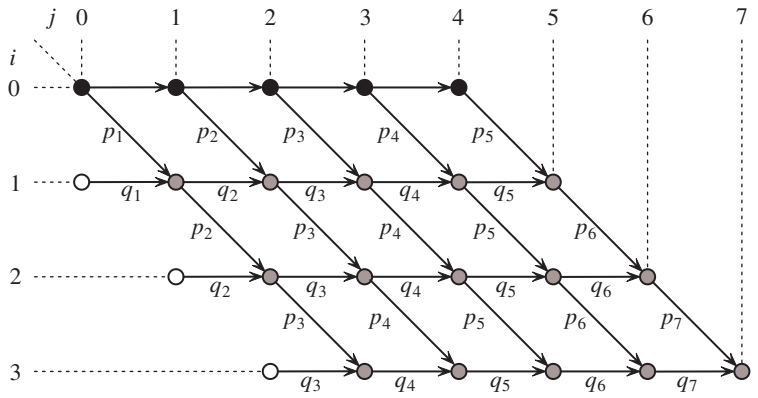

FIGURE 7.2 Signal flow graph for obtaining $R(3, 7)$ and $Q(3, 7)$.

other nodes, say (i, j), have to be calculated by adding the product of the immediate top-left entry and p_j to the product of the immediate left entry and q_j.

The same graph in Figure 7.2 can also be used for the computation of $Q(3, 7)$ provided that the graph nodes (i, j) are understood to represent the unreliabilities $Q(i, j)$ instead of the reliabilities $R(i, j)$, and the two types of source nodes interchange their values; that is, the black nodes at $i = 0$ become zero values $[Q(0, j) = 0]$ and the white nodes at $i = j + 1$ become unity values $[Q(j + 1, j) = 1]$.

The algorithm proceeds efficiently by directly constructing (i.e., computing the element values of) the parallelogram with corners $(1, 1)$, $(1, n - k + 1)$, (k, k), and (k, n). The number of elements in the parallelogram is $k(n - k + 1)$. Each element of the parallelogram requires three arithmetic operations (namely, one multiplication and two additions) for its evaluation. This can be easily seen by invoking the relation $q_j = 1 - p_j$ to simplify (7.26) and (7.27) into the following forms:

$$\begin{aligned} R(i, j) &= p_j R(i - 1, j - 1) + (1 - p_j) R(i, j - 1) \\ &= R(i, j - 1) + p_j \left(R(i - 1, j - 1) - R(i, j - 1)\right), \end{aligned} \tag{7.30}$$

$$\begin{aligned} Q(i, j) &= (1 - q_j) Q(i - 1, j - 1) + q_j Q(i, j - 1) \\ &= Q(i - 1, j - 1) + q_j \left(Q(i, j - 1) - Q(i - 1, j - 1)\right). \end{aligned} \tag{7.31}$$

This means that the algorithm by Rushdi requires $3k(n-k+1)$ arithmetic operations, and its computational complexity can be written as $O(k(n - k + 1))$.

Computation of the $R(i, j)$ or $Q(i, j)$ entries shown in Figure 7.2 can be processed by row, by column, or even diagonally. However, to minimize the memory requirements, this is done columnwise, for the $R(3, 7)$ case, with due attention paid to the parallelogram boundaries. In this case the algorithm requires a memory storage of $k+1 = 4$ scalars, in addition to the memory needed to store p_i for $1 \leq i \leq n$. The additional storage requirement for any such problem is $\min\{k + 1, n - k\}$ by calculating columnwise (if k is smaller) or rowwise (if $n-k+1$ is smaller). It is also interesting to note that this algorithm has the same computational complexity for its reliability and unreliability evaluations.

Detailed comparisons of the time and memory requirements of the algorithms by Sarje and Prasad [215], Rushdi [208], and Barlow and Heidtmann [20] are conducted by Rushdi [209]. The results are shown in Table 7.1. The time requirement is measured by the number of multiplications, additions, and array references. Table 7.1 shows that the algorithms by Barlow and Heidtmann and Rushdi are computationally more efficient and require less memory than the algorithm by Sarje and Prasad. Risse [202] and Pham and Upadhyaya [194] also give detailed comparisons of such algorithms that generally agree with this result.

Sample outputs of the algorithm by Rushdi for reliability and unreliability evaluations of a 5-out-of-8:G system are shown in Tables 7.2 and 7.3, respectively. In both tables, we have assumed that component reliabilities are $p_j = 0.9 - 0.01(j - 1)$ for $j = 1, 2, \ldots, 8$.

Another advantage of the algorithms by Barlow and Heidtmann [20] and Rushdi [208] is worth noting. All the intermediate entries needed for calculating $R(k, n)$ or

TABLE 7.1 Comparison of Time and Space Complexities

Algorithm	Sarje and Prasad [215]	Rushdi [208]	Barlow and Heidtmann [20]
Temporal complexity			
Multiplications	$4k(n-k)+4$	$k(n-k+1)$	$(k+1)(n-k+1)-1$
Additions	$2k(n-k+1)+n$	$2k(n-k+1)$	$(2k+1)(n-k+1)-2$
References to one-dimensional arrays	$4k(n-k)+k$	$4k(n-k+1)+1$	$(4k+3)(n-k+1)-3$
References to two-dimensional arrays	$4k(n-k)+2k+6$	0	0
Spatial Complexity:			
Memory requirements	$3(n-k+1)+2n$	$\min(k+1, n-k+2)$	$k+2$

Source: Rushdi [209].

TABLE 7.2 Calculating $R(5,8)$ of 5-out-of-8:G System with Symmetric Switching Function Approach

	j	0	1	2	3	4	5	6	7	8
	0	1	1	1	1					
	1	0	0.900000	0.989000	0.998680	0.999828				
i	2		0	0.801000	0.966440	0.994489	0.999081			
	3			0	0.704480	0.932437	0.985802	0.997089		
	4				0	0.613246	0.887750	0.971094	0.992930	
	5					0	0.527391	0.833697	0.949110	0.985480

TABLE 7.3 Calculating $Q(5,8)$ of 5-out-of-8:G System with Symmetric Switching Function Approach

	j	0	1	2	3	4	5	6	7	8
	0	0	0	0	0					
	1	1	0.100000	0.011000	0.001320	0.000172				
i	2		1	0.199000	0.033560	0.005511	0.000919			
	3			1	0.295120	0.067563	0.014198	0.002911		
	4				1	0.386754	0.112250	0.028906	0.007070	
	5					1	0.472609	0.166303	0.050890	0.014520

$Q(k,n)$ are meaningful numbers that represent $R(i,j)$ or $Q(i,j)$ for $1 \le i \le k$ and $i \le j \le n-k+i$. These numbers are available to the reliability engineer at no extra cost, and can enable one to make a valid economic assessment of redundancy. For example, row 5 in Table 7.2 represents the reliability $R(5,j)$, where j varies from 5 to 8. The incremental system reliability achieved by increasing system size from j to $j+1$ is

$$\Delta_j R(i,j) = R(5,j+1) - R(5,j). \tag{7.32}$$

The economic equivalence of this incremental reliability can, therefore, be estimated and compared to the cost of adding an additional component, thereby obtaining the optimal number of components for the 5-out-of-j system.

Exercises

1. Compute the reliability of a 2-out-of-6:G system with $p_i = 0.6 + 0.06i$ for $i = 1, 2, \ldots, 6$.
2. How do you use the algorithm covered in this section to evaluate the reliability of a k-out-of-n:F system?
3. What can you conclude through examination of the entries in the same column in Table 7.2 or 7.3?

MIS Technique To imbed the k-out-of-n:G system into the Markov chain following Definition 5.1, we can define the state space as $S = \{s_0, s_1, \ldots, s_k\} = \{0, 1, \ldots, k\}$, the partition as $S_i = \{i\}$ for $i = 0, 1, \ldots, k$ (here, $N = m = k$), and the Markov chain $\{Y_l,\ l \geq 0\}$ as

1. $Y_l = i$ if exactly i of the components $1, 2, \ldots, l$ are working $(0 \leq i < k)$ and
2. $Y_l = k$ if at least k of the components $1, 2, \ldots, l$ are working.

Recall that p_{ij} represents the probability for the Markov chain to make a transition from state i to state j. The transition matrix of the Markov chain is

$$\mathbf{\Lambda}_l = (p_{ij})_{(k+1)\times(k+1)} = \left[\begin{array}{ccccc|c} q_l & p_l & & & & \\ & q_l & p_l & & & \\ & & \ddots & \ddots & & \\ & & & q_l & p_l & \\ & & & & q_l & p_l \\ \hline & & & & & 1 \end{array}\right]_{(k+1)\times(k+1)},$$

where $i,\ j = 0, 1, \ldots, k$. (7.33)

Unmarked entries of the matrix are all equal to zero. This transition matrix provides the probabilities for the system with one more component, namely component l, to be in state $j\,(0 \leq j \leq k)$ given that the system with $l - 1$ components is in state $i\,(0 \leq i \leq k)$. In the k-out-of-n:G system, we are interested in knowing whether the number of working components has reached or exceeded k. That is why the system state space includes $0, 1, \ldots, k$ and represents the progressive increase in the number of working components as the system size increases. When the system size reaches n, the probability that the system is in state k is the reliability of the system.

Making use of the transition probability matrix given in equation (7.33) and noting $N = m = k$, we obtain the following recursive equations with Theorem 5.2:

$$a_0(l) = q_l a_0(l-1), \qquad l \geq 1, \tag{7.34}$$

$$a_j(l) = p_l a_{j-1}(l-1) + q_l a_j(l-1), \qquad 1 \leq j < k, \quad j \leq l \leq n, \tag{7.35}$$

$$a_k(l) = p_l a_{k-1}(l-1) + a_k(l-1), \qquad k \leq l \leq n, \tag{7.36}$$

where $a_j(l)$ is the probability that there are exactly j working components in a system with l components for $0 \leq j < k$ and $a_k(l)$ is the probability that there are at least k working components in the l-component subsystem. The following boundary conditions are immediate:

$$a_0(0) = 1, \tag{7.37}$$

$$a_j(0) = 0, \qquad j > 0, \tag{7.38}$$

$$a_j(l) = 0, \qquad l < j. \tag{7.39}$$

The reliability of the system is $R(k, n) = a_k(n)$.

The recursive equations (7.34)–(7.36) can also be represented by a signal flow diagram similar to the one shown in Figure 7.2. These recursive equations have the same iterative structure as the algorithms given by Barlow and Heidtmann [20] and Rushdi [208]. The computational complexity of the recursive equations (7.34)–(7.36) is $\sim 3k(n-k+1)$ or $O(k(n-k+1))$.

Example 7.2 Consider a 5-out-of-8:G system with component reliabilities $p_l = 0.9 - 0.01(l-1)$ for $l = 1, 2, \ldots, 8$. Table 7.4 lists the results using the MIS approach. Compare this table with Table 7.2.

From Table 7.4, we see that the MIS approach not only provides the reliability of a k-out-of-n:G system but also the probabilities that there are at least k working components in the l-component subsystem for $l = k, k+1, \ldots, n$. In addition, we also know the probabilities that there are exactly j working components in the l-component subsystems for $j = 0, 1, \ldots, k-1$ and $l = j, j+1, \ldots, j+n-k$. For example, from the column with $l = 6$ in Table 7.4, we see that in the six-component subsystem, the probability that there are exactly three working compo-

TABLE 7.4 Reliability Evaluation of 5-out-of-8:G System with MIS Approach

l		0	1	2	3	4	5	6	7	8
p_l			0.90	0.89	0.88	0.87	0.86	0.85	0.84	0.83
q_l			0.10	0.11	0.12	0.13	0.14	0.15	0.16	0.17
	0	1	0.100000	0.011000	0.001320					
	1	0	0.900000	0.188000	0.032240	0.005340				
j	2		0	0.801000	0.261560	0.062052	0.013280			
	3			0	0.704880	0.319192	0.098052	0.025996		
	4				0	0.613246	0.360360	0.137392	0.043802	
	5					0	0.527391	0.833697	0.949111	0.985482

nents is 0.025996, that there are exactly four working components is 0.137392, and that there are exactly five working components is 0.833697. From these entries, we can also find that the probability that there are at least three working components in the six-component subsystem is equal to $0.025996 + 0.137392 + 0.833697 = 0.997085$.

Fast Fourier Transform Method by Belfore Belfore [26] uses the generating function approach as developed by Barlow and Heidtmann [20] and applies the FFT in computation of the products of the generating functions. An algorithm for reliability evaluation of k-out-of-n:G systems results from such a combination that has a computational complexity of $O(n(\log_2 n)^2)$. In the following, we explain this FFT approach.

Consider the following generation function:

$$g_n(z) = \prod_{i=1}^{n}(p_i + q_i z). \tag{7.40}$$

It is a polynomial function of variable z. The coefficient of term z^i in this polynomial function represents the probability that exactly i components are failed and thus the other $n - i$ components are working. By factoring out the products of the component reliabilities, we can express equation (7.40) in the form

$$g_n(z) = P_\pi \prod_{i=1}^{n}(1 + a_i z) = P_\pi(1 + A(z)), \tag{7.41}$$

where

$$\begin{aligned} P_\pi &= p_1 p_2 \cdots p_n, \\ a_i &= \frac{q_i}{p_i} \qquad \text{for } i = 1, 2, \ldots, n, \\ A(z) &= b_1 z + b_2 z^2 + \cdots + b_n z^n, \\ b_i &= \sum_{1 \le j_1 < j_2 < \cdots j_i \le n} a_{j_1} a_{j_2} \cdots a_{j_i} \qquad \text{for } i = 1, 2, \ldots, n. \end{aligned}$$

Using the form of the generating function in equation (7.41) rather than the one in equation (7.40) results in fewer computations because the multiplications by 1 are implicit and the increases in FFT sizes are delayed since the FFT is applied to $A(z)$ [26]. Once the coefficients of $A(z)$, or b_i for $i = 1, 2, \ldots, n$, are calculated, we can find the reliability of a k-out-of-n:G system with the following equation:

$$R(k, n) = P_\pi \left(1 + \sum_{i=1}^{n-k} b_i\right). \tag{7.42}$$

Suppose, for ease of explanation, that n is a power of 2. Define $A(z)$ as an nth-order polynomial function of z. The term $1 + A(z)$ in equation (7.41) can be viewed as a product of two generating functions, $1 + A_1(z)$ and $1 + A_2(z)$, where $A_1(z)$ and $A_2(z)$ are $(n/2)$th-order polynomial functions of z. We have

$$1 + A(z) = [1 + A_1(z)][1 + A_2(z)] = 1 + A_1(z) + A_2(z) + A_1(z)A_2(z).$$

As a result,

$$A(z) = A_1(z) + A_2(z) + A_1(z)A_2(z). \tag{7.43}$$

To find $A(z)$, first we need to compute the product of two $(n/2)$th-order polynomial functions, $A_1(z)$ and $A_2(z)$. In turn $A_1(z)$ and $A_2(z)$ each is equal to the sum of two lower order [$(n/4)$th-order] polynomial functions plus their product. This process can be repeated until the order of the polynomial functions is low enough so that the exact values of its coefficients become apparent.

Finding the expression of the product of two $(n/2)$th-order polynomial functions $A_1(z)$ and $A_2(z)$ is equivalent to finding the coefficients of the resulting polynomial function. This can be achieved using FFT. First, we define a discrete function corresponding to $A_i(z)(i = 1, 2)$. This discrete function takes values of the coefficients of $A_i(z)(i = 1, 2)$ over the definition domain of $\{0, 1, 2, \ldots, (n/2) - 1\}$. If $A_1(z)$ and $A_2(z)$ are in the forms

$$A_1(z) = c_1 z + c_2 z^2 + \cdots + c_{n/2} z^{n/2},$$
$$A_2(z) = d_1 z + d_2 z^2 + \cdots + d_{n/2} z^{n/2},$$

the two discrete functions are in the forms

$$f_1(x) = \begin{cases} c_{x+1} & \text{if } x = 0, 1, 2, \ldots, n/2 - 1, \\ 0 & \text{otherwise,} \end{cases}$$

$$f_2(x) = \begin{cases} d_{x+1} & \text{if } x = 0, 1, 2, \ldots, n/2 - 1, \\ 0 & \text{otherwise.} \end{cases}$$

The convolution of these two discrete functions is given by

$$f(x) = f_1(x) * f_2(x) \equiv \sum_{y=0}^{n/2-1} f_1(x - y) f_2(y),$$

where $*$ denotes the convolution operator. We have to note that the definition domain of the resulting function, $f(x)$, is $\{0, 1, 2, \ldots, n - 1\}$. This is in agreement with the product of two $(n/2)$th-order polynomial functions being an nth-order polynomial function. The values of $f(x)$ for $x \in \{0, 1, 2, \ldots, n\}$ provide the coefficients of the resulting polynomial function, namely $A_1(z)A_2(z)$. We can then use equation (7.43)

to find the coefficients of $A(z)$. However, we need an efficient method to find the convolution of two discrete functions.

Based on the convolution theorem in Fourier theory, the Fourier transform of the convolution of two functions is equal to the product of the Fourier transforms of these two individual functions. Thus, to find the coefficients of the product of two polynomial functions, we can first find the FFT of two discrete functions corresponding to the two polynomial functions, multiply these two FFTs in the frequency domain, and finally conduct the inverse FFT on the resulting product to obtain the desired coefficients of the resulting polynomial function. For details on FFT, readers are referred to Bracewell [41].

Using the generating function form as shown in equation (7.41) and assuming n is a power of 2, the number of operations required to multiply $1 + A_1(z)$ and $1 + A_2(z)$, where $A_1(z)$ and $A_2(z)$ each is a $(n/2)$th-order polynomial function, using FFT [26] is

$$T(n) = 15n \log_2(n) + 11n - 2. \tag{7.44}$$

Apparently, large overheads are involved in the FFT approach. Thus, it is not efficient to use this approach for small n values. When n is small, we can use the algorithm provided by Barlow and Heidtmann [20] to directly find the coefficients of the generating function. Belfore [26] shows that the FFT approach is more efficient than the algorithm by Barlow and Heidtmann (BH) when n is larger than 512. The following algorithm based on Barlow and Heidtmann [20] is used to compute the coefficients of the generating functions for small n values:

```
BH(n, a[1 : n], Az[1 : n])
  integer i, j;
  Az[1] = a[1];
  For i = 2 To n By 1 Do
    Az[i] = Az[i − 1] ∗ a[i];
    For  j = i − 1 To 2 By −1 Do
      Az[j] = Az[j] + Az[j − 1] ∗ a[i];
    EndFor
    Az[1] = Az[1] + a[i];
  EndFor
  Return Az[1 : n];
End
```

In the BH algorithm shown above, $a[1 : n]$ is an array of size n holding the ratios q_i/p_i and $Az[1 : n]$ is an array of size n holding the coefficients of z_i in the resulting generating function for $i = 1, 2, \ldots, n$. The sizes of these arrays are determined by the calling algorithm through the argument n.

The following algorithm is used to calculate the coefficients of the generating function shown in equation (7.41) for large n values:

```
GF_FFT(n, a[1 : n], Az[1 : n])
  If n <= threshold Then
    Call BH(n, a, Az);
  Else
    Call GF_FFT(n/2, a[1 : n/2], Az[1 : n/2]);
    Call GF_FFT(n−n/2, a[n/2 + 1 : n], Az[n/2 + 1 : n]);
    FFT_size = 2^liub(log2 n);
    Initialize temp1, temp2;
    For i = 1 To n/2 By 1 Do
      temp1[i].real = Az[i];
    EndFor
    For i = n/2 + 1 To n By 1 Do
      temp2[i − n/2].real = Az[i];
    EndFor
    Compute FFT of temp1;
    Compute FFT of temp2;
    For i = 1 To FFT_size By 1 Do
      temp1[i] = temp1[i] * temp2[i];
    EndFor
    Compute the inverse FFT of temp1 and assign it to I_temp1;
    Az[1] = Az[1] + Az[n/2 + 1];
    For i = 2 To n/2 By 1 Do
      Az[i] = Az[i] + Az[n/2 + i] + I_temp1[i − 1];
    EndFor
    For i = n/2 + 1 To n−n/2 By 1 Do
      Az[i] = Az[n/2 + i] + I_temp1[i − 1];
    Endfor
    For i = n − n/2 + 1 To n By 1 Do
      Az[i] = I_temp1[i − 1];
    Endfor
  EndIf
  Return Az[1 : n];
End
```

In this algorithm, $n/2$ is defined to be the largest integer less than or equal to $n/2$ when n is not divisible by 2; *liub* indicates the lowest integer upper bound; *temp*1 and *temp*2 are complex variable arrays, both of size *FFT_size*; $a[n/2 + 1 : n]$ indicates that a subarray of $a[1 : n]$ with values in positions $n/2+1$ through n of $a[1 : n]$ being used; and the calling algorithm defines the sizes of the arrays in the arguments. The *threshold* value is specified by the user so that when n is smaller than the *threshold*, the BH algorithm is used for calculating the coefficients of the generating function.

The system reliability of a k-out-of-n:G system can be calculated with the following algorithm according to equation (7.42):

```
R_sys_FFT(n, k, p[1 : n], q[1 : n])
  P_pi = 1;
  For i = 1 To n By 1 Do
    P_pi = P_pi * p[i];
    a[i] = q[i]/p[i];
  EndFor
  Call GF_FFT(n, a[1 : n], Az[1 : n]);
  Rel = 1;
  For i = 1 To n − k By 1 Do
    Rel = Rel + Az[i];
  EndFor
  Rel = P_pi * Rel;
  Return Rel;
End
```

The lower and upper bounds on the complexity of the algorithm R_sys_FFT for a *threshold* value of 2 are given by Belfore [26] as

$$T_{\text{lower}}(n) = \tfrac{1}{2} \times 15n(\log_2 n)^2 + \tfrac{1}{2} \times 37n \log_2 n - 23n + 2, \tag{7.45}$$

$$T_{\text{upper}}(n) = 15n(\log_2 n)^2 + 67n \log_2 n + 6n + 2. \tag{7.46}$$

For simplicity, we can say that the FFT approach has a time complexity of

$$O(n(\log_2 n)^2).$$

7.1.3 Bounds on System Reliability

When the components in a k-out-of-n system are s-independent, the algorithms presented in Section 7.1.2 are quite efficient for evaluation of exact system reliability. However, the components in the system may not be independent in some cases. To add to the difficulty, the way in which the components are dependent on each other may not be completely understood. In this section, we provide discussion on system reliability approximation when components are not necessarily independent.

Associated Components As introduced earlier, the concept of association indicates that two random variables have nonnegative covariance. In this case, we may use the theorem given in Barlow and Proschan [22] to find the upper and lower bounds on system reliability of k-out-of-n:G systems.

Let $P_1, P_2, \ldots, P_r$ represent the minimal path sets. We have $r = \binom{n}{k}$ and there are k components in each of these minimal path sets. Let $K_1, K_2, \ldots, K_t$ represent the minimal cut sets. Then, $t = \binom{n}{n-k+1}$ and there are $n - k + 1$ components in each of these minimal cut sets. The following bounds on system reliability are given by

Barlow and Proschan [22]:

$$\max_{1\le i\le r}\prod_{j\in P_i} p_j \le R_s \le \min_{1\le i\le t}\left\{1-\prod_{j\in K_i}(1-p_j)\right\}. \tag{7.47}$$

Unspecified Dependence of Components Without making any assumptions on how components are dependent on one another, Lipow [144] provides a simple formula for the lower bound of the reliability of a k-out-of-n:G system:

$$R_s \ge \max_{1\le i\le r}\sum_{j\in P_i} p_j - k + 1, \tag{7.48}$$

where $r = \binom{n}{k}$ and P_i is the ith minimal path set. This formula was derived from the IE method for system reliability evaluation using minimal path sets. It is useful only when component reliabilities are pretty close to 1 and k is not too big.

Exercises

1. Analyze the closeness of the bounds given in (7.47) to exact system reliability for a k-out-of-n:G system. Consider cases when component reliabilities are high and low.
2. Develop an upper bound for system reliability using the IE method when component dependency is unspecified. Under what conditions will this bound be close to the exact system reliability?
3. Analyze the closeness of the bounds given in (7.48) to exact system reliability for a k-out-of-n:G system. Consider cases when component reliabilities are high and low.

7.2 RELATIONSHIP BETWEEN k-OUT-OF-n G AND F SYSTEMS

In the previous section, we illustrated different approaches for reliability evaluation of k-out-of-n:G systems. Exercises were given for following similar approaches to derive algorithms for reliability evaluation of k-out-of-n:F systems. In this section, we provide a formal discussion of the relationship between k-out-of-n G and F systems and how reliability evaluation algorithms for these two types of systems are closely related.

7.2.1 Equivalence between k-out-of-n:G and $(n-k+1)$-out-of-n:F Systems

Based on the definitions of these two types of systems, a k-out-of-n:G system is equivalent to an $(n-k+1)$-out-of-n:F system. Similarly, a k-out-of-n:F system is equivalent to an $(n-k+1)$-out-of-n:G system. This means that provided the systems have the same set of component reliabilities, the reliability of a k-out-of-n:G system

is equal to the reliability of an $(n-k+1)$-out-of-n:F system and the reliability of a k-out-of-n:F system is equal to the reliability of an $(n-k+1)$-out-of-n:G system. As a result, we can use the algorithms that have been covered in the previous section for the k-out-of-n:G systems in reliability evaluation of the k-out-of-n:F systems. The procedure is simple and is outlined below:

Procedure for Using Algorithms for the G Systems in Reliability Evaluation of the F Systems Utilizing the Equivalence Relationship

1. Given $k, n, p_1, p_2, \ldots, p_n$ for a k-out-of-n:F system.
2. Calculate $k_1 = n - k + 1$.
3. Use $k_1, n, p_1, p_2, \ldots, p_n$ to calculate the reliability of a k_1-out-of-n:G system. This reliability is also the reliability of the original k-out-of-n:F system.

7.2.2 Dual Relationship between k-out-of-n G and F Systems

Barlow and Proschan [22] provide the following definition of a dual structure.

Definition 7.1 Given a structure ϕ, its dual structure ϕ^D is given by

$$\phi^D(\mathbf{x}) = 1 - \phi(\mathbf{1} - \mathbf{x}), \tag{7.49}$$

where $\mathbf{1} - \mathbf{x} = (1 - x_1, 1 - x_2, \ldots, 1 - x_n)$.

With a simple variable substitution of $\mathbf{1}-\mathbf{x}$ for $\mathbf{x}$, we have the equation

$$\phi^D(\mathbf{1} - \mathbf{x}) = 1 - \phi(\mathbf{x}). \tag{7.50}$$

We can interpret equation (7.50) as follows. Given a primal system with component state vector $\mathbf{x}$ and the system state represented by $\phi(\mathbf{x})$, the state of the dual system is equal to $1 - \phi(\mathbf{x})$ if the component state vector for the dual system can be expressed as $\mathbf{1} - \mathbf{x}$. In the binary system context, each component and the system may only be in two possible states, either working or failed. We will say that two components with different states have *opposite states*. For example, if component 1 is in state 1 and component 2 is in state 0, components 1 and 2 have opposite states. Suppose a system (called system 1) has component state vector $\mathbf{x}$ and system state $\phi(\mathbf{x})$. Consider another system (called system 2) with the same number of components as system 1. If each component in system 2 has the opposite state of the corresponding component in system 1 and the state of system 2 becomes the opposite of the state of system 1, then system 1 and system 2 are duals of each other.

Now we examine the k-out-of-n G and F systems. Suppose that in the k-out-of-n:G system, there are exactly j *working* components and the system is working (in other words, $j \geq k$). Now assume that there are exactly j *failed* components in the k-out-of-n:F system. Since $j \geq k$, the k-out-of-n:F system must be in the failed state. If $j < k$, the k-out-of-n:G system is failed, and at the same time the k-out-of-n:F system is working. Thus, the k-out-of-n G and F systems are duals of each

other. Using the equivalence relationship described in the previous section, we can also say that the dual of a k-out-of-n:G system is an $(n-k+1)$-out-of-n:G system. Similarly, we can say that a k-out-of-n:F system is the dual of an $(n-k+1)$-out-of-n:F system. These dual and equivalence relationships between the k-out-of-n G and F systems are summarized below:

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

Using the dual relationship, we can summarize the following procedure for reliability evaluation of the dual system if the available algorithms are for the primal system:

Procedure for Using Algorithms for the G Systems in Reliability Evaluation of the F Systems Utilizing the Dual Relationship

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

Using the dual relationship, we can also obtain algorithms for k-out-of-n:F system reliability evaluation from those developed for the k-out-of-n:G systems. We only need to change reliability measures to unreliability measures and vice versa. Take the algorithm developed by Rushdi [208] as an example. The formulas for reliability and unreliability evaluation of a k-out-of-n:G system are given in equations (7.26) and (7.27) with boundary conditions in equations (7.28) and (7.29). By changing $R(i, j)$ to $Q(i, j)$, $Q(i, j)$ to $R(i, j)$, p_i to q_i, and q_i to p_i in those four equations, we obtain the following equations for reliability and unreliability evaluation of a k-out-of-n:F system:

$$Q_F(i, j) = q_j Q_F(i-1, j-1) + p_j Q_F(i, j-1), \tag{7.51}$$

$$R_F(i, j) = q_j R_F(i-1, j-1) + p_j R_F(i, j-1), \tag{7.52}$$

with the boundary conditions

$$Q_F(0, j) = R_F(j+1, j) = 1, \tag{7.53}$$

$$Q_F(j+1, j) = R_F(0, j) = 0. \tag{7.54}$$

To avoid confusion, the subscript F is added to indicate that these measures are for the F system. Similar steps can be applied to other algorithms for the G systems to derive the corresponding algorithms for the F systems. It is because of such close relationships between the k-out-of-n G and F systems that we often refer to them collectively as the k-out-of-n systems.

Example 7.3 Consider a k-out-of-n:F system with $k = 3$, $n = 7$, and $p_i = 0.8 + 0.02i$ for $i = 1, \ldots, 7$. Use the two procedures listed in Section 7.2 to evaluate the reliability of the system.

The 3-out-of-7:F system is equivalent to a 5-out-of-7:G system with the same set of components. Table 7.5 lists the calculations needed to find the reliability of the 5-out-of-7:G system. The reliability of the 5-out-of-7:G system is found to be 0.959836, which is equal to the reliability of the original 3-out-of-7:F system. Each entry in Table 7.5 can be interpreted in terms of either a k-out-of-n:G subsystem or a $(n - k + 1)$-out-of-n:F subsystem. For example, 0.853982 in the column labeled 5 and the row labeled 4 represents the reliability of a 4-out-of-5:G subsystem and at the same time the reliability of the equivalent 2-out-of-5:F subsystem.

We can also use the dual relationship between a k-out-of-n:F system and a k-out-of-n:G system. From the given p_i values, calculate all $q_i = 1 - p_i$ for $i = 1, 2, \ldots, 7$. Treat these q_i's as component reliability values and apply the formulas for the k-out-of-n:G system. Table 7.6 lists these calculations. The rightmost entry at the bottom row in Table 7.6 is the reliability of the 3-out-of-7:G system. To find the reliability of the original 3-out-of-7:F system, we need to subtract this value from 1:

$$R_F(3, 7) = 1 - 0.040163 = 0.959837.$$

TABLE 7.5 Reliability Evaluation of Equivalent 5-out-of-7:G System

k \ n	0	1	2	3	4	5	6	7
0	1	1	1					
1	0	0.820000	0.971200	0.995968				
2		0	0.688800	0.931664	0.998252			
3			0	0.592368	0.890948	0.978521		
4				0	0.521284	0.853982	0.968558	
5					0	0.469156	0.823196	0.959836

TABLE 7.6 Reliability Evaluation of Dual System: 3-out-of-7:G System

k \ n	0	1	2	3	4	5	6	7
0	1	1	1	1	1			
1	0	0.180000	0.311200	0.407632	0.478716	0.530845		
2		0	0.028800	0.068336	0.109052	0.146018	0.176804	
3			0	0.004032	0.011748	0.021479	0.031442	0.040164

Exercises

1. Derive the formulas for reliability evaluation of a k-out-of-n:F system based on the algorithm by Barlow and Heidtmann.
2. Derive the formulas for reliability evaluation of a k-out-of-n:F system based on the MIS approach.
3. Compute the reliabilities of the k-out-of-n F and G systems with $k = 2, 3, 4$ and $n = 5, 6, 7$.

7.3 NONREPAIRABLE k-OUT-OF-n SYSTEMS

In the previous sections, we have discussed the so-called *static properties* of k-out-of-n systems. Reliability has not been expressed as a function of time. But, in fact, reliability and other performance measures of any system are functions of time. Starting from this section, we will provide stochastic analyses of the k-out-of-n systems. In reality, it is sometimes impossible to repair a system until its mission is complete. In this case, the reliability of the system is a decreasing function of time. In this section, we examine the performance measures of nonrepairable k-out-of-n systems.

Notation

- T_i: lifetime of component i, a random variable
- T_s: lifetime of the system, a random variable
- $R_i(t)$: $\Pr(T_i \geq t)$, reliability function of component i
- $R(t)$: reliability function of each component when components are i.i.d.
- $F_i(t)$: $1 - R_i(t)$, CDF or unreliability function of component i
- $F(t)$: CDF or unreliability function of each component when components are i.i.d.
- $f_i(t)$: pdf of the lifetime of component i
- $f(t)$: pdf of the lifetime of each component when components are i.i.d.
- $h_i(t)$: failure rate function of component i
- $h(t)$: failure rate function of each component when components are i.i.d.
- $R_s(t)$: reliability function of the system
- $F_s(t)$: $1 - R_s(t)$, CDF or unreliability function of the system
- $f_s(t)$: pdf of the lifetime of the system
- $h_s(t)$: failure rate function of the system
- $R(t; k, n)$: reliability function of the k-out-of-n:G system
- MTTF_s: mean time to failure of the system
- $\text{MTTF}(k, n)$: mean time to failure of a k-out-of-n:G system

7.3.1 Systems with i.i.d. Components

When the components in a k-out-of-n:G system are i.i.d., the reliability function of the system can be expressed as

$$R_s(t) = \sum_{i=k}^{n} \binom{n}{i} R(t)^i F(t)^{n-i}. \tag{7.55}$$

This equation is directly obtained from equation (7.2) by replacing p with $R(t)$ and q with $F(t)$. Similarly, the CDF of the system lifetime is given by

$$F_s(t) = 1 - R_s(t) = \sum_{i=0}^{k-1} \binom{n}{i} R(t)^i F(t)^{n-i}. \tag{7.56}$$

The pdf of the system lifetime is then

$$f_s(t) = \frac{dF_s(t)}{dt} = k\binom{n}{k} f(t) F(t)^{n-k} R(t)^{k-1}. \tag{7.57}$$

Usually, as the system is used, the components in the system will fail one by one. The system is failed as soon as the $(n-k+1)$th component is failed. If we use t_i to indicate the lifetime of component i, the system lifetime is then equal to the $(n-k+1)$th smallest t_i. The expected lifetime of the system, or mean time to failure, can be evaluated using the standard equation

$$\text{MTTF}_s = \int_0^\infty t f_s(t)\, dt = \int_0^\infty R_s(t)\, dt. \tag{7.58}$$

In the following, we first illustrate that when all i.i.d. components have IFR or even constant failure rates, the system has IFR. No specific component lifetime distributions are assumed:

$$R_s(t) = \int_t^\infty f_s(x)\, dx,$$

$$\frac{1}{h_s(t)} = \frac{R_s(t)}{f_s(t)} = \frac{\displaystyle\int_t^\infty f(x) F(x)^{n-k} R(x)^{k-1}\, dx}{f(t) F(t)^{n-k} R(t)^{k-1}}$$

$$= \frac{1}{f(t)} \int_t^\infty f(x) \left(\frac{F(x)}{F(t)}\right)^{n-k} \left(\frac{R(x)}{R(t)}\right)^{k-1} dx.$$

Let $y = R(x)/R(t)$; then $dy = -[f(x)/R(t)]\, dx$:

$$\frac{1}{h_s(t)} = \frac{1}{h(t)} \int_0^1 \left(\frac{1 - yR(t)}{F(t)}\right)^{n-k} y^{k-1}\, dy. \tag{7.59}$$

Since $[1 - yR(t)]/F(t)$ is decreasing in t and $h(t)$ is assumed to be IFR, we conclude that $h_s(t)$ is increasing in t based on equation (7.59). This indicates that if all

components have IFR, the k-out-of-n:G structure preserves this IFR property of the components. If all components have constant failure rates, the k-out-of-n:G system would have IFR as long as $k \neq n$ and a constant failure rate when $k = n$.

It is generally impossible to find more specific expressions of the performance measures of the k-out-of-n:G system. However, when the components follow the exponential distribution, some explicit results can be derived. When all components follow the exponential lifetime distribution with CDF $F(t) = 1 - e^{-\lambda t}$, the expressions of system reliability and unreliability are

$$R_s(t) = \sum_{i=k}^{n} \binom{n}{i} (e^{-\lambda t})^i (1 - e^{-\lambda t})^{n-i}, \tag{7.60}$$

$$F_s(t) = \sum_{i=0}^{k-1} \binom{n}{i} (e^{-\lambda t})^i (1 - e^{-\lambda t})^{n-i}, \tag{7.61}$$

respectively. The MTTF of the system can be derived as follows. Based on equation (7.7), we have, for $k \geq 2$,

$$R_s(t; k, n) = R(t; k-1, n) - \binom{n}{k-1} e^{-\lambda t(k-1)} (1 - e^{-\lambda t})^{n-k+1}. \tag{7.62}$$

Integrating both sides of this equation results in the following recursive equation:

$$\begin{aligned} \text{MTTF}(k, n) &= \text{MTTF}(k-1, n) - \int_0^\infty \binom{n}{k-1} e^{-\lambda t(k-1)} (1 - e^{-\lambda t})^{n-k+1} \, dt \\ &= \text{MTTF}(k-1, n) - \frac{1}{\lambda} \binom{n}{k-1} \sum_{j=0}^{n-k+1} \binom{n-k+1}{j} \frac{(-1)^j}{k-1+j} \\ &= \text{MTTF}(k-1, n) - \frac{1}{\lambda(k-1)}. \end{aligned} \tag{7.63}$$

The following equation is used in the above derivations:

$$\sum_{j=0}^{N} \binom{N}{j} \frac{(-1)^j}{a+j} = \frac{N!(a-1)!}{(N+a)!} \qquad \text{for } a \geq 1. \tag{7.64}$$

$\text{MTTF}(1, n)$ represents the MTTF of a parallel system, which is $(1/\lambda) \sum_{j=1}^{n} (1/j)$. Using this boundary condition and applying equation (7.63) recursively, we find

$$\text{MTTF}(k, n) = \frac{1}{\lambda} \sum_{j=k}^{n} \frac{1}{j}. \tag{7.65}$$

Substituting $k = n$ in equation (7.65) provides the MTTF of a series system, $1/(n\lambda)$, as is expected.

Using equation (7.59), we can express the system failure rate as

$$h_s(t) = \frac{\lambda}{\displaystyle\int_0^1 y^{k-1}[(1 - ye^{-\lambda t})/(1 - e^{-\lambda t})]^{n-k}\, dy}. \tag{7.66}$$

No closed-form expression for $h_s(t)$ can be obtained even in the case when all components have exponential lifetime distributions.

Exercises

1. Verify equation (7.57).
2. Verify equation (7.63).

7.3.2 Systems with Nonidentical Components

It is generally difficult to write an expression for k-out-of-n system reliability when the components do not have identical lifetime distributions. It is possible to derive the desired expressions for simple cases. For components with exponential lifetime distributions such that component i has a constant failure rate $\lambda_i (1 \leq i \leq n)$, we have the following expressions of system reliability and MTTF for a 2-out-of-3:G system:

$$R_s(t; 2, 3) = e^{-(\lambda_1+\lambda_2)t} + e^{-(\lambda_1+\lambda_3)t} + e^{-(\lambda_2+\lambda_3)t} - 2e^{-(\lambda_1+\lambda_2+\lambda_3)t},$$

$$\text{MTTF}(2, 3) = \frac{1}{\lambda_1 + \lambda_2} + \frac{1}{\lambda_1 + \lambda_3} + \frac{1}{\lambda_2 + \lambda_3} - \frac{2}{\lambda_1 + \lambda_2 + \lambda_3}.$$

7.3.3 Systems with Load-Sharing Components Following Exponential Lifetime Distributions

Consider a k-out-of-n:G system with i.i.d. components each following the exponential lifetime distribution. When the system is put into operation at time zero, all components are working and they are equally sharing the constant load that the system is supposed to carry. In this case, the failure rate of every component is denoted by λ_0. When the system experiences the first failure, the remaining $n - 1$ working components must carry the same load on the system. As a result, the failure rate of each working component becomes λ_1, which is usually higher than λ_0. When i components are failed, the failure rate of each of the $n - i$ working components is represented by λ_i $(0 \leq i \leq n - k)$. The system is failed when more than $n - k$ components are failed. For such a system with no repair provisions, Scheuer [218] provides an analysis of the system's performance measures.

Notation

- λ_i: failure rate of each surviving component when i components have failed $(0 \leq i \leq n-k)$. Assume $\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{n-k}$ due to practical considerations.

- T_i: time to the ith failure ($T_0 \equiv 0$), $i = 1, 2, \ldots, n-k+1$
- X_i: time between the $(i-1)$th failure and the ith failure, $X_i = T_i - T_{i-1}$, $i = 1, 2, \ldots, n-k+1$
- α_i: failure rate of the system when there are i failed components, $\alpha_i = (n-i+1)\lambda_{i-1}$, $i = 1, 2, \ldots, n-k+1$

Since all components are i.i.d. following the exponential distributions, the interarrival times of failures are independent random variables and X_i follows the exponential distribution with parameter α_i for $1 \le i \le n-k+1$. The lifetime of the system is equal to the $(n-k+1)$st failure time, that is,

$$T_s = T_{n-k+1} = X_1 + X_2 + \cdots + X_{n-k+1}.$$

The MTTF of the system is then

$$\text{MTTF}_s = \sum_{i=1}^{n-k+1} \frac{1}{\alpha_i} = \sum_{i=1}^{n-k+1} \frac{1}{(n-i+1)\lambda_{i-1}}.$$

The distribution of T_s is the distribution of a sum of $n-k+1$ independent random variables, each following the exponential distribution with possibly different parameters. To find the distribution of T_s and the reliability function of the system, we need to distinguish the following three cases.

Case I: $\boldsymbol{\alpha_1 = \alpha_2 = \cdots = \alpha_{n-k+1} \equiv \alpha}$ This case arises when the load of the system is equally shared by surviving components. If the failure rate of each surviving component is directly proportional to the load it carries, we can write λ_i as

$$\lambda_i = c\left(\frac{d}{n-i}\right), \qquad i = 0, 1, 2, \ldots, n-k,$$

where d is the load on the system and c is a constant. Using this equation, we can verify the following:

$$\alpha_i = (n-i+1)\lambda_{i-1} = cd \equiv \alpha, \qquad i = 1, 2, \ldots, n-k+1.$$

Thus, under case I, X_i's for $i = 1, 2, \ldots, n-k+1$ are i.i.d. random variables following the same exponential distribution with parameter α. As a result, T_s, a sum of these $n-k+1$ i.i.d. random variables, follows the gamma distribution with scale parameter α and shape parameter $n-k+1$. The pdf of this gamma distribution is

$$f(t) = \frac{\alpha(\alpha t)^{n-k}}{(n-k)!} e^{-\alpha t}.$$

The reliability function of the system is then

$$R_s(t) = \sum_{j=0}^{n-k} \frac{(\alpha t)^j}{j!} e^{-\alpha t}. \tag{7.67}$$

Case II: $\alpha_1, \alpha_2, \ldots, \alpha_{n-k+1}$ Take Distinct Values In this case, the lifetime of the system is a sum of $n-k+1$ independent random variables each with a distinct exponential distribution parameter. The pdf of the system lifetime is a convolution of the pdf's of these $n-k+1$ exponential random variables. With the technique of Laplace transform, the pdf of the system's lifetime is found to be

$$f_s(t) = \left(\prod_{i=1}^{n-k+1} \alpha_i \right) \sum_{i=1}^{n-k+1} \frac{e^{-\alpha_i t}}{\prod_{j=1, j\neq i}^{n-k+1} (\alpha_j - \alpha_i)}.$$

From $f_s(t)$, we find the reliability function of the system:

$$R_s(t) = \sum_{i=1}^{n-k+1} A_i e^{-\alpha_i t}, \tag{7.68}$$

$$A_i = \prod_{j=1, j\neq i}^{n-k+1} \frac{\alpha_j}{\alpha_j - \alpha_i}, \qquad i = 1, 2, \ldots, n-k+1. \tag{7.69}$$

Case III: $\alpha_1, \alpha_2, \ldots, \alpha_{n-k+1}$, Are neither Identical nor Distinct Specifically, assume that these α_i's take a $(1 < a < n)$ distinct values, $\beta_1, \beta_2, \ldots, \beta_a$. With possibly some renumbering of these α_i values, assume

$$\begin{aligned}
\alpha_1 = \alpha_2 = \cdots = \alpha_{r_1} &\equiv \beta_1, \\
\alpha_{r_1+1} = \alpha_{r1+2} = \cdots = \alpha_{r_1+r_2} &\equiv \beta_2, \\
&\vdots \\
\alpha_{r_1+r_2+\cdots+r_{a-1}+1} = \cdots = \alpha_{r_1+r_2+\cdots+r_a} &\equiv \beta_a, \\
r_1 + r_2 + \cdots + r_a = n-k+1, \qquad 1 \le r_i < n, &\qquad i = 1, 2, \ldots, a.
\end{aligned}$$

Under case III, the interarrival times of failures are divided into a $(a > 1)$ groups. Group j has r_j identical interarrival times following the exponential distribution with the same parameter. The interarrival times in different groups follow exponential distributions with different parameters. If we define the lifetime of each group as the sum of the interarrival times within the group, such group lifetimes then follow the gamma distribution. The lifetime of group j, denoted by V_j, for $j = 1, 2, \ldots, a$ follows the gamma distribution with scale parameter β_j and shape parameter r_j. In addition, these group lifetimes are independent. As a result, we can write the lifetime of the system, T_s, as a sum of the lifetimes of the groups, each following a different gamma distribution:

$$T_s = V_1 + V_2 + \cdots + V_a.$$

The reliability function of the system is given by

$$R_s(t) = B \sum_{j=1}^{a} \sum_{\ell=1}^{r_j} \frac{\Phi_{j\ell}(-\beta_j)}{(\ell-1)!\beta^{r_j-\ell+1}} \sum_{i=0}^{r_j-\ell} \frac{(\beta_j t)^i e^{-\beta_j t}}{i!}, \tag{7.70}$$

where

$$B = \prod_{j=1}^{a} \beta_j^{r_j}, \tag{7.71}$$

$$\Phi_{j\ell}(t) = \frac{d^{\ell-1}}{dt^{\ell-1}} \prod_{i=1, i\neq j}^{a} (\beta_i + t)^{-r_i}. \tag{7.72}$$

These equations can be derived as follows. Assume that V_j has the gamma distribution with scale parameter β_j and shape parameter r_j (a positive integer) and its pdf can be written as

$$f_j(t) = \frac{\beta_j(\beta_j t)^{r_j-1} e^{-\beta_j t}}{(r_j-1)!}.$$

The Laplace transform of $f_j(t)$ is

$$\mathcal{L}_j(s) = \left(\frac{\beta_j}{\beta_j + s}\right)^{r_j}.$$

The pdf of T_s is a convolution of the individual pdf's of the lifetimes of these a groups. The Laplace transform of a convolution of functions is equal to the product of the Laplace transforms of these individual functions. As a result, the Laplace transform of the pdf of T_s is

$$\mathcal{L}_s(s) = \prod_{j=1}^{a} \left(\frac{\beta_j}{\beta_j + s}\right)^{r_j}. \tag{7.73}$$

The inverse Laplace transform of equation (7.73) will give the pdf of T_s [73]:

$$f_s(t) = B \sum_{j=1}^{a} \sum_{\ell=1}^{r_j} \frac{\Phi_{j\ell}(-\beta_j)}{(\ell-1)!(r_j-\ell)!} t^{r_j-\ell} e^{-\beta_j t}. \tag{7.74}$$

From this pdf of T_s, we can find the $R_s(t)$ as given in equation (7.70).

Exercises

1. Derive equation (7.68).
2. Derive the system reliability function under case II.

7.3.4 Systems with Load-Sharing Components Following Arbitrary Lifetime Distributions

Liu [145] provides an analysis of the k-out-of-n:G system with i.i.d. components whose lifetime distributions are not necessarily exponential. Repair of failed components is not allowed. Surviving components equally share the constant load of the system. The lifetime distribution of a component under a constant load can be represented by the accelerated failure time model (AFTM) or the accelerated life model. The parametric form of the AFTM for each component is assumed to be known. The AFTM specifies that the effect of load on the lifetime of a component is multiplicative in time. The reliability function of a component under the AFTM can be expressed as

$$R(t, \mathbf{z}) = R_0(t\psi(\mathbf{z})), \tag{7.75}$$

where $\mathbf{z}$ is a vector representing the loads on the component, $\psi(\mathbf{z})$ is an acceleration factor, and $R_0(\cdot)$ is the reliability function of an arbitrary statistical distribution. For more discussions on AFTM, readers are referred to Nelson [175]. When there is only one type of load, z, commonly used forms of $\psi(z)$ include

$$\psi(z) = e^{\alpha z}, \qquad \psi(z) = z^{\alpha}.$$

For example, if $R_0(\cdot)$ is of the Weibull distribution, that is, $R_0(x) = e^{-(t/\eta)^{\beta}}$, and $\psi(z) = z^{\alpha}$, we can write the load-dependent reliability function of the component as

$$R(t; z) = \exp\left[-\left(\frac{tz^{\alpha}}{\eta}\right)^{\beta}\right]. \tag{7.76}$$

When $R_0(\cdot)$ is Weibull, the AFTM is equivalent to the proportional hazard model (PHM) [59], wherein the load acts multiplicatively on the failure rate. When $R_0(\cdot)$ is not Weibull, the AFTM is not equivalent to the PHM. In the following, we illustrate the reliability analysis of a k-out-of-n:G system with i.i.d. load-sharing components whose lifetimes can be modeled with AFTM as given in equation (7.75).

Notation

- $R(t; z)$: reliability function of each component when the total load on the system is z
- z_{n-j}: total load to be shared by $n-j$ surviving components when j components are failed

When $k = n$, we have a series system. All components have to work for the system to work. Since components are independent,

$$R(t; n, n) = \prod_{i=1}^{n} R(t; z_n) = [R(t; z_n)]^n .$$

When $k = n-1$, for the system to survive beyond t, either all components survive beyond t or one component fails at time $x (0 < x < t)$ and all other components

survive the remaining time duration $t - x$:

$$R(t; n-1, n) = R(t; n, n) + n \int_0^t f(x; z_n) \left[R(t - x - \hat{x}; z_{n-1})\right]^{n-1} dx, \tag{7.77}$$

where $\hat{x} = x\psi(z_n)/\psi(z_{n-1})$.

When $k = n - 2$, we have

$$\begin{aligned} R(t; n-2, n) = R(t; n-1, n) &+ \frac{n!}{(n-2)!} \int_0^t \int_0^{x_1} f(x; z_n) f(x_1 - x + \hat{x}; z_{n-1}) \\ &\times \left[R(t - x_1 + \hat{x}_1; z_{n-2})\right]^{n-2} dx\, dx_1, \end{aligned} \tag{7.78}$$

where $\hat{x} = x\psi(z_n)/\psi(z_{n-1})$ and $\hat{x}_1 = (x_1 - x + \hat{x})\psi(z_{n-1})/\psi(z_{n-2})$.

When $k = n - 3$,

$$\begin{aligned} R(t; n-3, n) = R(t; n-2, n) &+ \frac{n!}{(n-3)!} \int_0^t \int_0^{x_2} \int_0^{x_1} f(x; z_n) f(x_1 - x + \hat{x}; z_{n-1}) \\ &\times f(x_2 - x_1 + \hat{x}_1; z_{n-2}) \\ &\times \left[R(t - x_2 + \hat{x}_2; z_{n-3})\right]^{n-3} dx\, dx_1\, dx_2, \end{aligned} \tag{7.79}$$

where $\hat{x} = x\psi(z_n)/\psi(z_{n-1})$, $\hat{x}_1 = (x_1 - x + \hat{x})\psi(z_{n-1})/\psi(z_{n-2})$, and $\hat{x}_2 = (x_2 - x_1 + \hat{x}_1)\psi(z_{n-2})/\psi(z_{n-3})$.

Generally, the following equation can be used for evaluation of $R(t; j, n)$ for $1 \le j < n$:

$$\begin{aligned} R(t; j, n) = R(t; j+1, n) &+ \frac{n!}{j!} \int_0^t \int_0^{x_{n-j-1}} \int_0^{x_{n-j-2}} \cdots \int_0^{x_2} \int_0^{x_1} f(x; z_n) \\ &\times f(x_1 - x + \hat{x}; z_{n-1}) f(x_2 - x_1 + \hat{x}_1; z_{n-2}) \times \cdots \\ &\times f(x_{n-j-2} - x_{n-j-3} + \hat{x}_{n-j-3}; z_{n-(n-j-2)}) \\ &\times f(x_{n-j-1} - x_{n-j-2} + \hat{x}_{n-j-2}; z_{n-(n-j-1)}) \times \left[R(t - x_{n-j-1} \right. \\ &\left. + \hat{x}_{n-j-1}; z_{n-(n-j)})\right]^j dx\, dx_1\, dx_2 \cdots dx_{n-j-2}\, dx_{n-j-1}, \end{aligned} \tag{7.80}$$

where $\hat{x}_i = (x_i - x_{i-1} + \hat{x}_{i-1})\psi(z_{n-i})/[\psi(z_{n-(i+1)})]$ for $i = 1, 2, \ldots, n - j - 1$ and $s_0 \equiv s$.

The procedure outlined above is enumerative in nature. More efficient methods for handling arbitrary load-dependent component lifetime distributions are needed.

7.3.5 Systems with Standby Components

As we mentioned before, the k-out-of-n system structure has built-in redundancy. Actually the system requires only k components to work for the system to work. In deriving the equations for k-out-of-n system performance evaluations so far in this chapter, we have treated the extra $n-k$ components as active redundant components. In other words, they are in hot standby mode. In this section, we will analyze the k-out-of-n system with cold and warm standby components.

Cold Standby with i.i.d. Components and Perfect Switching In Chapter 4, we discussed standby systems with n components. When the components are i.i.d. following the exponential lifetime distribution with parameter λ, the lifetime of the system follows the gamma distribution with scale parameter λ and shape parameter n. In the following, we will show that the lifetime of a k-out-of-n:G system with i.i.d. cold standby components can also be described by a gamma distribution.

For a k-out-of-n:G system with standby components, k components are put into operation initially and $n-k$ components are in standby. Whenever one of the active components is failed, one of the standby components is switched into operation. No repair provisions are allowed. Sensing and switching are assumed to be perfect. The system is failed when $n-k+1$ component failures have been experienced. The k active components can be viewed as a series subsystem since all of them are required to work for the k-out-of-n:G system to work. As was explained in Chapter 4, the failure rate of a series system is equal to the sum of the failure rates of the components when all components have constant failure rates. If all components in the k-out-of-n:G system are i.i.d. with a constant failure rate λ, then the series subsystem with k active components has a failure rate of $k\lambda$. Whenever one of the components in this series subsystem is failed, it is replaced by a standby component and a new series subsystem is formed. Because of the memoryless property of the exponential distribution, each series subsystem follows the exponential lifetime distribution with parameter $k\lambda$. The system is failed when the k-component series subsystem, including the last standby component, is failed. Thus, we have the following expression of system lifetime:

$$T_s = T_1 + T_2 + \cdots + T_{n-k+1}, \tag{7.81}$$

where T_i represents the lifetime of the ith k-component series subsystem. Even though these k-component series subsystems have components in common, their lifetimes $T_1, T_2, \ldots, T_{n-k+1}$ are i.i.d. random variables because of the memoryless property of the exponential distribution. The sum of i.i.d. random variables with the exponential distribution follows the gamma distribution. Thus, T_s follows the gamma distribution with scale parameter $k\lambda$ and shape parameter $n-k+1$:

$$f_s(t) = k\lambda e^{-k\lambda t}\frac{(k\lambda t)^{n-k}}{(n-k)!}, \qquad t \geq 0, \tag{7.82}$$

$$R_s(t) = e^{-k\lambda t} \sum_{j=0}^{n-k} \frac{(k\lambda t)^j}{j!}, \tag{7.83}$$

$$\text{MTTF}_s = \frac{n-k+1}{k\lambda}. \tag{7.84}$$

The derivations outlined above are based on the fact that the $n-k+1$ k-component series subsystems have i.i.d. exponential lifetime distributions. This is satisfied only when each component follows the exponential distribution. When the lifetime distribution of each component is not exponential, we cannot use the results shown above.

Warm Standby System with i.i.d. Components and Perfect Switching As mentioned in Chapter 4, warm standby systems are more complicated to analyze because both active and dormant components may fail. Assuming that all components are i.i.d. and the lifetime of each component follows the exponential distribution with parameter λ_a in the active state and parameter λ_d in the dormant state, She and Pecht [227] provide a closed-form expression for system reliability function.

Notation

- λ_a: constant failure rate of an active component
- λ_d: constant failure rate of a dormant or standby component
- $f_a(\cdot)$, $R_a(\cdot)$: pdf and reliability function of an active component, respectively
- $f_d(\cdot)$, $R_d(\cdot)$: pdf and reliability function of a dormant component, respectively

The event that the system survives beyond time t may be expressed as the union of the following mutually exclusive events:

1. The k active components all survive beyond time t.
2. One of the k active components fails in interval $(x, x+dx)$ for $0 < x < t$, all $n-k$ dormant components survive beyond time x, and the $(n-1)$-component subsystem with k active and $n-k-1$ dormant components survives the remaining time period $t-x$.
3. One of the $n-k$ dormant components fails in interval $(x, x+dx)$ for $0 < x < t$; all k active components survive beyond time x; and the $(n-1)$-component subsystem with k active and $n-k-1$ dormant components survives the remaining time period $t-x$.

Based on this decomposition, we can express $R(t; k, n)$ as

$$\begin{aligned} R(t; k, n) = e^{-k\lambda_a t} &+ \int_0^t \binom{k}{1} f_a(x) R_d(x)^{n-k} R_s(t-x; k, n-1)\, dx \\ &+ \int_0^t \binom{n-k}{1} f_d(x) R_a(x)^k R(t-x; k, n-1)\, dx. \end{aligned} \tag{7.85}$$

This equation can be applied recursively until we reach $R(z; k, k) = e^{-k\lambda_a z}$, which is the reliability function of a k-component series system. The closed-form expression for the system reliability is

$$R(t; k, n) = \frac{1}{(n-k)!\lambda_d^{n-k}} \sum_{i=0}^{n-k} (-1)^i \binom{n-k}{i} \times \left[\prod_{j=0, j\neq i}^{n-k} (k\lambda_a + j\lambda_d) \right] e^{-(k\lambda_a + i\lambda_d)t}. \tag{7.86}$$

When $\lambda_d = \lambda_a = \lambda$, equation (7.86) reduces to the system reliability function of a k-out-of-n:G system with active redundancy given in equation (7.60). When $\lambda_d = 0$, we get the reliability function of a k-out-of-n:G system with cold standby components, as given in equation (7.83).

Exercises

1. Derive the expression of the MTTF of the warm standby system.
2. Verify that equation (7.86) reduces to equation (7.60) when $\lambda_d = \lambda_a = \lambda$.
3. Verify that equation (7.85) reduces to equation (7.83) when $\lambda_d = 0$.

7.4 REPAIRABLE k-OUT-OF-n SYSTEMS

We have discussed the k-out-of-n:G systems with active redundant components, with standby components, or with load-sharing components. In this section, we will develop a general model for analysis of such systems when they are repairable. After such a model is developed, we will analyze various system performance measures under different assumptions.

When a k-out-of-n:G system is put into operation, all n components are in good condition. As the system is used, components will fail one after another. The system is failed when the number of working components goes down below k or the number of failed components has reached $n-k+1$. If resources are allocated to repair failed components, we should be able to keep the number of failed components below $n-k+1$ for a much longer time. This way, we expect to prolong the system life cycle. Whenever the number of failed components at any instant of time is higher than $n-k$, the system is failed and its life cycle is complete.

Many situations exist in which more than one failed component can be repaired simultaneously (in parallel). This can be achieved when there exist more than one repairman or repair facility. As the number of repair facilities is increased, we expect a better chance of extending the system operating time until its first failure.

In the following section we describe a general repairable k-out-of-n:G system model with multiple repair facilities. The components may be in active redundancy, standby, or load sharing. Such a model will allow us to evaluate such performance

measures of the system as mean time to failure, steady-state availability, and mean time between failures.

7.4.1 General Repairable System Model

Here are the model descriptions and assumptions:

1. The system is a k-out-of-n:G structure with possibly cold standby and/or load-sharing components.
2. The failure of each component is self-revealing.
3. All active components are i.i.d. following the exponential lifetime distributions. However, the parameter of the lifetime distribution of each component may change depending on the load applied on the component.
4. There are r identical repair facilities available ($1 \leq r \leq n-k+1$). Only one repair facility may be assigned to the repair of a failed component. The time needed by any repair facility to repair any failed component is i.i.d. with the exponential distribution.
5. Whenever a component fails, repair immediately commences if a repair facility is available; if not, the failed component must wait for the first available repair facility. Components are repaired on a first-come, first-served basis.
6. The system is considered failed as soon as the number of components in the failed state has reached $n-k+1$.
7. While the system is down, no further units can fail.
8. The state of the system is defined to be the number of failed components in the system that are either waiting for or are receiving repair.
9. The system state is decreased by 1 whenever a failed component becomes operational and increased by 1 whenever a working component becomes failed.
10. The probability that two or more components are restored to the working condition or become failed in a small time interval is negligible.

Notation

- i: number of failed components in the system, $i = 0, 1, \ldots, n-k+1$
- t: time
- λ_i: failure rate of the system when there are i failed components, $0 \leq i \leq n-k$
- μ_i: repair rate of the system when there are i failed components, $1 \leq i \leq n-k+1$
- $P_i(t)$: probability that there are i failed components in the system at time t, $0 \leq i \leq n-k+1$
- P_i: steady-state probability, $P_i = \lim_{t\to\infty} P_i(t)$, $0 \leq i \leq n-k+1$
- $P_i'(t)$: first derivative of $P_i(t)$, $0 \leq i \leq n-k+1$
- $\mathcal{L}_i(s)$: Laplace transform of $P_i(t)$, $0 \leq i \leq n-k+1$

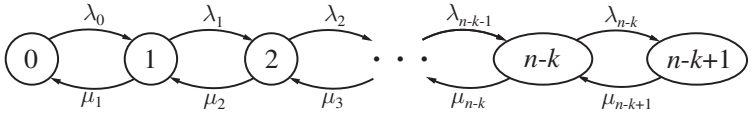

FIGURE 7.3 General transition diagram for repairable k-out-of-n:G system.

- $A_s(t)$: point availability of the system at time t
- A_s: steady-state availability of the system, $A_s = \lim_{t\to\infty} A_s(t)$

Based on the model descriptions, the system state transition diagram is given in Figure 7.3. The numbers in the circles in Figure 7.3 indicate the system states. The system state $n - k + 1$ indicates system failure.

To evaluate $P_i(t + \Delta t)$, we note that at time $t + \Delta t$ the system can be in state i only if one of the following disjoint events occurs:

1. At time t the system is in state i and during $(t, t + \Delta t)$ no change in system state occurs.
2. At time t the system is in state $i - 1$ and a transition to state i occurs during $(t, t + \Delta t)$.
3. At time t the system is in state $i + 1$ and a transition to state i occurs during $(t, t + \Delta t)$.
4. During $(t, t + \Delta t)$, the system state changes by two or more.

Since Δt is very small, the probability of the last event is $o(\Delta t)$, as assumed. As a result, we have

$$\begin{aligned} P_i(t + \Delta t) &= P_i(t)(1 - \lambda_i\,\Delta t)(1 - \mu_i\,\Delta t) + P_{i-1}(t)\lambda_{i-1}\Delta t \\ &\quad \times (1 - \mu_{i-1}\,\Delta t) + P_{i+1}(t)\mu_{i+1}\,\Delta t(1 - \lambda_{i+1}\,\Delta t) + o(\Delta t) \\ &= P_i(t) - P_i(t)(\lambda_i + \mu_i)\,\Delta t + P_{i-1}(t)\lambda_{i-1}\Delta t \\ &\quad + P_{i+1}(t)\mu_{i+1}\Delta t + o(\Delta t). \end{aligned} \tag{7.87}$$

Rearranging the terms in equation (7.87) and letting $\Delta t \to 0$, we have

$$\begin{aligned} P_i'(t) &= -(\lambda_i + \mu_i)P_i(t) + \lambda_{i-1}P_{i-1}(t) + \mu_{i+1}P_{i+1}(t) \\ &\quad \text{for } i = 0, 1, \ldots, n - k + 1, \end{aligned} \tag{7.88}$$

where $P_i(t) \equiv 0$ for $i < 0$ or $i > n - k + 1$. We shall assume the initial conditions $P_i(0) = 0$ if $i \neq 0$ and $P_0(0) = 1$, that is, all components are assumed to be initially in the working state. Considering these initial conditions and assumptions, we can rewrite equation (7.88) as

$$P_0'(t) = -\lambda_0 P_0(t) + \mu_1 P_1(t), \tag{7.89}$$

$$P_i'(t) = -(\lambda_i + \mu_i)P_i(t) + \lambda_{i-1}P_{i-1}(t) + \mu_{i+1}P_{i+1}(t) \quad \text{for } 1 \le i \le n-k, \tag{7.90}$$

$$P_{n-k+1}'(t) = -\mu_{n-k+1}P_{n-k+1}(t) + \lambda_{n-k}P_{n-k}(t). \tag{7.91}$$

One of these equations can be written as a linear combination of the other $n-k+1$ equations because the system must be in one of the $n-k+2$ states at any instant of time, that is,

$$P_0(t) + P_1(t) + \cdots + P_{n-k+1}(t) = 1 \qquad \text{for any } t \ge 0. \tag{7.92}$$

Thus, equation (7.91) should be replaced by equation (7.92). The set of differential equations to be solved are equations (7.89), (7.90), and (7.92). Solving this set of differential equations results in the probability distribution of the system in various states as a function of time. Once this distribution is found, we can evaluate system performance measures such as mean time to failure, mean time between failures, and steady-state availability of the system. The Laplace transform is an effective method for solving systems of differential equations.

Taking Laplace transforms of equations (7.89), (7.90), and (7.92) yields the following linear equations in terms of $\mathcal{L}_i(s)$ for $i = 0, 1, \ldots, n-k+1$:

$$(s + \lambda_0)\mathcal{L}_0(s) - \mu_1\mathcal{L}_1(s) = 1, \tag{7.93}$$

$$(s + \lambda_i + \mu_i)\mathcal{L}_i(s) - \lambda_{i-1}\mathcal{L}_{i-1}(s) - \mu_{i+1}\mathcal{L}_{i+1}(s) = 0 \quad \text{for } 1 \le i \le n-k, \tag{7.94}$$

$$s(\mathcal{L}_0(s) + \mathcal{L}_1(s) + \cdots + \mathcal{L}_{n-k+1}(s)) = 1. \tag{7.95}$$

Using matrix notation, we can rewrite these equations as

$$\mathbf{DX} = \mathbf{B}, \tag{7.96}$$

where

$$\mathbf{X} = \begin{bmatrix} \mathcal{L}_0(s) \\ \mathcal{L}_1(s) \\ \vdots \\ \mathcal{L}_{n-k}(s) \\ \mathcal{L}_{n-k+1}(s) \end{bmatrix}_{(n-k+2)\times 1}, \qquad \mathbf{B} = \begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}_{(n-k+2)\times 1},$$

$$\mathbf{D} = \begin{bmatrix} s+\lambda_0 & -\mu_1 & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ -\lambda_0 & s+\lambda_1+\mu_1 & -\mu_2 & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & -\lambda_1 & s+\lambda_2+\mu_2 & -\mu_3 & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & -\lambda_{n-k-1} & s+\lambda_{n-k}+\mu_{n-k} & -\mu_{n-k+1} \\ s & s & s & s & s & \cdots & s & s & s \end{bmatrix},$$

where $\mathbf{D}$ is an $(n-k+2) \times (n-k+2)$ matrix.

7.4.2 Systems with Active Redundant Components

The general repair model is used for evaluation of system performance measures.

Reliability Function and Mean Time to Failure Even when the system under consideration is a repairable system, we are still interested in finding the MTTF of the system. This is different from the nonrepairable system case that was analyzed earlier. In this case, as components fail, they also get repaired. If repairs are timely enough, the system may experience more than $n - k$ cumulative component failures without experiencing a system failure.

Since we are interested in finding the MTTF of the system, we need to assume that state $n-k+1$ is an absorbing state. As soon as the number of components in the failed state at any instant of time reaches $n - k + 1$, the system is considered failed.

As a result, we have to assume $\mu_{n-k+1} = 0$. When the system is in state $i (0 \leq i \leq n - k)$, there are i failed components and $n - i$ active working components in the system, and the failure rate of the system is $\lambda_i = (n - i)\lambda$. If the number of failed components is less than or equal to the total number of repair facilities, all failed components are being repaired, and thus the repair rate of the system is $\mu_i = i\mu (1 \leq i \leq r)$. However, if $i > r$, μ_i will be a constant equal to $r\mu$ as all repair facilities are being used and some failed components are waiting for repair. The following summarizes these conditions:

$$\lambda_i = (n - i)\lambda, \qquad 0 \leq i \leq n - k,$$

$$\mu_i = \begin{cases} i\mu & \text{for } 0 \leq i \leq r, \\ r\mu & \text{for } r < i \leq n - k, \\ 0 & \text{for } i = n - k + 1. \end{cases}$$

In the system of differential equations, $P_{n-k+1}(t)$ is the probability that the system is in the failed state at time t. Thus, the reliability function of the system is $R_s(t) = 1 - P_{n-k+1}(t)$. We can use the Laplace transform to find $P_{n-k+1}(t)$.

Because $\mu_{n-k+1} = 0$, equation (7.91) can be written as

$$P'_{n-k+1}(t) = \lambda_{n-k} P_{n-k}(t) = k\lambda P_{n-k}(t), \tag{7.97}$$

$$P_{n-k+1}(t) = k\lambda \int_0^t P_{n-k}(x)\, dx, \tag{7.98}$$

assuming that $P_{n-k+1}(0) = 0$. Note that $P_{n-k+1}(t)$ and $P'_{n-k+1}(t)$ actually represent the CDF and the pdf, respectively, of the system lifetime. Also, because $\mu_{n-k+1} = 0$, $P_{n-k+1}(t)$ disappears from equation (7.90). As a result, equations (7.89) and (7.90) include $n - k + 1$ equations and $n - k + 1$ variables, $P_0(t), P_1(t), \ldots, P_{n-k}(t)$. After the Laplace transform, the system of linear equations (7.96) includes $n - k + 1$ equations and $n - k + 1$ variables. The last entries in vectors $\mathbf{X}$ and $\mathbf{B}$ and the last row and the last column of matrix $\mathbf{D}$ are removed.

The method of determinants may be used to solve equation (7.96) for $\mathcal{L}_{n-k}(s)$,

$$\mathcal{L}_{n-k}(s) = \frac{|\mathbf{D}'|}{|\mathbf{D}|}, \tag{7.99}$$

where $\mathbf{D}'$ is the matrix obtained from $\mathbf{D}$ by replacing the $(n-k+1)$st column (the last column) of $\mathbf{D}$ by vector $\mathbf{B}$. The determinant of matrix $\mathbf{D}'$ is

$$|\mathbf{D}'| = \prod_{i=0}^{n-k-1} \lambda_i = \frac{n!\lambda^{n-k}}{k!}. \tag{7.100}$$

Since $|\mathbf{D}|$ is a polynomial in s with degree $n-k+1$ and leading coefficient 1, we can write $|\mathbf{D}| = \prod_{i=1}^{n-k+1}(s-s_i)$ where each s_i is a (distinct) root of the polynomial. Therefore,

$$\frac{1}{|\mathbf{D}|} = \left(\prod_{i=1}^{n-k+1}(s-s_i)\right)^{-1} = \sum_{j=1}^{n-k+1}\left(\prod_{i=1,i\neq j}^{n-k+1}(s_j-s_i)\right)^{-1}\frac{1}{s-s_j}, \tag{7.101}$$

$$\mathcal{L}_{n-k}(s) = \frac{|\mathbf{D}'|}{|\mathbf{D}|} = \frac{n!\lambda^{n-k}}{k!}\sum_{j=1}^{n-k+1}\left(\prod_{i=1,i\neq j}^{n-k+1}(s_j-s_i)\right)^{-1}\frac{1}{s-s_j}. \tag{7.102}$$

An inverse Laplace transform of equation (7.102) yields

$$P_{n-k}(t) = \frac{n!\lambda^{n-k}}{k!}\sum_{j=1}^{n-k+1}\left(\prod_{i=1,i\neq j}^{n-k+1}(s_j-s_i)\right)^{-1} e^{s_j t}, \tag{7.103}$$

$$\begin{aligned} R_s(t) &= 1 - \frac{n!\lambda^{n-k+1}}{(k-1)!}\sum_{j=1}^{n-k+1}\left(\prod_{i=1,i\neq j}^{n-k+1}(s_j-s_i)\right)^{-1}\int_0^t e^{s_j x}\,dx \\ &= 1 - \frac{n!\lambda^{n-k+1}}{(k-1)!}\sum_{j=1}^{n-k+1}\left[\frac{e^{s_j t}-1}{s_j}\left(\prod_{i=1,i\neq j}^{n-k+1}(s_j-s_i)\right)^{-1}\right] \\ &= 1 - \sum_{j=1}^{n-k+1} C_j(e^{s_j t}-1), \end{aligned} \tag{7.104}$$

where

$$C_j = \left(\prod_{i=1,i\neq j}^{n-k+1} s_i\right)\left(\prod_{i=1,i\neq j}^{n-k+1}(s_j-s_i)\right)^{-1}.$$

Derivation of equation (7.104) uses the fact that $[n!/(k-1)!]\lambda^{n-k+1} = \prod_{i=1}^{n-k+1} s_i$, which is the constant term in the polynomial representing the determinant of the matrix $\mathbf{D}$. It can be shown that $s_i < 0$ for all i; thus $t^2 R_s(t) \to 0$ as $t \to +\infty$.

Now that an expression of the system reliability function is known, we can derive the MTTF of the system based on its definition:

$$\text{MTTF}_s = \int_0^\infty R_s(t)\,dt = \sum_{i=0}^{n-k} \int_0^\infty P_i(t)\,dt. \tag{7.105}$$

Since $\mathcal{L}_i(0) = \int_0^\infty P_i(t)\,dt$, MTTF_s can also be written as

$$\text{MTTF}_s = \sum_{i=0}^{n-k} \mathcal{L}_i(0). \tag{7.106}$$

If we let $\mathbf{D}_i'$ denote the matrix obtained from $\mathbf{D}$ by replacing the $(i+1)$th column by the vector $\mathbf{B}$, then

$$\mathcal{L}_i(0) = \frac{|\mathbf{D}_i'|_{s=0}}{|\mathbf{D}|_{s=0}}. \tag{7.107}$$

Since

$$|\mathbf{D}|_{s=0} = \frac{n!\lambda^{n-k+1}}{(k-1)!}, \tag{7.108}$$

we have

$$\text{MTTF}_s = \frac{(k-1)!}{n!\lambda^{n-k+1}} \sum_{i=0}^{n-k} |\mathbf{D}_i'|_{s=0}. \tag{7.109}$$

Provided that the s_j's are known, another way for evaluation of MTTF_s is to use the fact that $P'_{n-k+1}(x)$ is actually the pdf of the system lifetime. With equation (7.97), an equivalent form for $R_s(t)$ is

$$R_s(t) = \int_t^\infty P'_{n-k+1}(x)\,dx = \int_t^\infty k\lambda P_{n-k}(x)\,dx. \tag{7.110}$$

This implies that

$$\text{MTTF}_s = \int_0^\infty \int_t^\infty k\lambda P_{n-k}(x)\,dx\,dt. \tag{7.111}$$

Exercises

1. Derive $R_s(t)$ and MTTF_s when $k = 2$, $n = 3$, and $r = 1$.
2. Derive $R_s(t)$ and MTTF_s when $k = 3$, $n = 5$, and $r = 2$.

3. Verify equation (7.101).
4. Verify equation (7.104).

Steady-State Availability As the k-out-of-n:G system is used, the number of failed components in the system changes. When it reaches $n-k+1$, the system is failed and all repair facilities are utilized to repair failed components. As soon as the number of failed components goes down below $n-k+1$, the system starts working again. Thus, the system state changes between up and down over time. The probability that the system is in the working state at time t is called the point availability of the system. The system point availability is given by

$$A_s(t) = 1 - P_{n-k+1}(t). \tag{7.112}$$

To evaluate the availability of the system, we have to treat state $n-k+1$ as a transient state too, that is, $\mu_{n-k+1} = r\mu$. The following summarizes the system parameters:

$$\lambda_i = (n-i)\lambda, \qquad 0 \le i \le n-k,$$

$$\mu_i = \begin{cases} i\mu & \text{if } 0 \le i \le r, \\ r\mu & \text{if } r < i \le n-k+1. \end{cases}$$

To evaluate the point availability $A_s(t)$ of the system, the general repair system model can be used. With the Laplace transform technique, it suffices to find $\mathcal{L}_{n-k+1}(s)$. Using the method of determinants, we observe that

$$\mathcal{L}_{n-k+1}(s) = \frac{|\mathbf{D}'|}{|\mathbf{D}|}, \tag{7.113}$$

where $\mathbf{D}'$ is the matrix obtained from $\mathbf{D}$ upon replacing the $(n-k+2)$nd column (the last column) by vector $\mathbf{B}$. Since

$$|\mathbf{D}'| = \frac{n!\lambda^{n-k+1}}{(k-1)!}, \tag{7.114}$$

we have

$$\mathcal{L}_{n-k+1}(s) = \frac{n!\lambda^{n-k+1}}{(k-1)!|\mathbf{D}|}. \tag{7.115}$$

The steady-state availability of the system is given by

$$\begin{aligned} A_s &= \lim_{t\to\infty} A_s(t) = 1 - \lim_{t\to\infty} P_{n-k+1}(t) \\ &= 1 - \lim_{s\to 0} s\mathcal{L}_{n-k+1}(s) = 1 - \lim_{s\to 0} \frac{sn!\lambda^{n-k+1}}{(k-1)!|\mathbf{D}|}. \end{aligned} \tag{7.116}$$

The procedure outlined above is necessary if one is interested in the availability of the system as a function of time t. However, the mathematical derivation is very

tedious. If one is only interested in the steady-state availability of the system, there is no need to derive the point availability first. The differential equations of the system given in equations (7.89), (7.90), and (7.92) can be used directly. First, we try to find the steady-state (or time-independent) distribution of the system in different states. This solution is provided by defining

$$P_i = \lim_{t\to\infty} P_i(t) \qquad \text{for } i = 0, 1, \ldots, n-k+1, \tag{7.117}$$

provided that the limits exist. Taking the limit of both sides of equations (7.89), (7.90), and (7.92) as $t \to \infty$ and noting that $\lim_{t\to\infty} P_i'(t) = 0$, we obtain

$$-\lambda_0 P_0 + \mu_1 P_1 = 0, \tag{7.118}$$

$$-(\lambda_i + \mu_i)P_i + \lambda_{i-1}P_{i-1} + \mu_{i+1}P_{i+1} = 0 \qquad \text{for } i = 1, \ldots, n-k+1, \tag{7.119}$$

$$P_0 + P_1 + \cdots + P_{n-k+1} = 1. \tag{7.120}$$

A simple induction argument shows that

$$P_i = \frac{\lambda_0 \cdots \lambda_{i-1}}{\mu_1 \cdots \mu_i} P_0 \qquad \text{for } i = 1, \ldots, n-k+1. \tag{7.121}$$

Applying equation (7.120), we have

$$P_0 = \left(1 + \sum_{i=1}^{n-k+1} \frac{\lambda_0 \cdots \lambda_{i-1}}{\mu_1 \cdots \mu_i}\right)^{-1}, \tag{7.122}$$

$$P_i = \frac{\lambda_0 \cdots \lambda_{i-1}}{\mu_1 \cdots \mu_i} \left(1 + \sum_{i=1}^{n-k+1} \frac{\lambda_0 \cdots \lambda_{i-1}}{\mu_1 \cdots \mu_i}\right)^{-1} \qquad \text{for } i = 1, \ldots, n-k+1, \tag{7.123}$$

$$A_s = 1 - P_{n-k+1} = 1 - \frac{\lambda_0 \cdots \lambda_{n-k}}{\mu_1 \cdots \mu_{n-k+1}} \left(1 + \sum_{i=1}^{n-k+1} \frac{\lambda_0 \cdots \lambda_{n-k}}{\mu_1 \cdots \mu_{n-k+1}}\right)^{-1}. \tag{7.124}$$

Mean Time between Failures The mean time between failures (MTBF) is defined to be the expected length of operating time of the system between successive failures. It does not include the time that the system spends in the failed state. The mean time to repair (MTTR) indicates the average length of time that the system stays in the failed state.

It is often necessary to calculate MTBF quickly in order to make timely design decisions. Although a general formula is known, it is not easily remembered nor

derived. Angus [13] presents a simple way of obtaining an expression of MTBF. With this method, the MTBF expression is easily reproduced by remembering a few simple concepts.

In the following, we assume that there are $n-k+1$ repair facilities. This means that no failed components need to wait for repair. The system transition parameters are summarized as follows:

$$\lambda_i = (n-i)\lambda, \qquad 0 \le i \le n-k,$$
$$\mu_i = i\mu, \qquad 1 \le i \le n-k+1.$$

The MTBF of the system is the average (successful operating) time between visits to state $n-k+1$, the system down state. It is the average time for the system to go from a working state (with possibly some failed components) to the failed state. This should be distinguished from the mean time to the first failure (MTTF) of the system, which is defined as the average time for the system to go from the working state with zero failed components to the failure state. In the following, we illustrate the derivation of MTBF of a k-out-of-n:G system.

Let $N(t)$ indicate the number of failed components in the system at time t. Because of the Markov nature of the process $\{N(t);\ t \ge 0\}$, once the process arrives at the state $n-k+1$, the sequence of times between successive visits to state $n-k+1$ forms an i.i.d. sequence of random variables. The mean of each of these random variables is MTBF + MTBR. The portion of this average that represents successful operation time is MTBF. It follows from the renewal theory (in particular, the analysis of alternating renewal processes) that

$$A_s = \lim_{t\to\infty} \Pr(\text{system is working at time } t) = \lim_{t\to\infty} \Pr(N(t) \le n-k)$$
$$= \frac{\text{MTBF}_s}{\text{MTBF}_s + \text{MTBR}_s}. \tag{7.125}$$

Solving for MTBF_s gives

$$\text{MTBF}_s = \frac{A_s \times \text{MTBR}_s}{1 - A_s}. \tag{7.126}$$

Each component has a MTBF of $1/\lambda$ and a MTBR of $1/\mu$. The steady-state availability of each component A is

$$A = \frac{\mu}{\lambda + \mu}.$$

When the system is down, there are $n-k+1$ units undergoing repair, and because of the Markov assumptions,

$$\text{MTBR}_s = \frac{1}{\mu(n-k+1)}. \tag{7.127}$$

Because there is always a repair facility available for a failed component and the components are s-independent, the limiting probability ($t \to \infty$) of finding exactly $j (k-1 \le j \le n)$ components working at time t is given by the truncated binomial distribution (truncated because the Markov process is not allowed to visit states $n-k+2, n-k+3, \ldots, n$):

$$P_j = \frac{\Pr(\text{exactly } j \text{ components are available})}{\Pr(\text{At least } k-1 \text{ components are available})} \tag{7.128}$$

$$= \frac{\binom{n}{j} A^j (1-A)^{n-j}}{\sum_{i=k-1}^{n} \binom{n}{i} A^i (1-A)^{n-i}} \quad \text{for } j = k-1, k, k+1, \ldots, n, \tag{7.129}$$

$$A_s = P_k + P_{k+1} + \cdots + P_n = \frac{\sum_{i=k}^{n} \binom{n}{i} A^i (1-A)^{n-i}}{\sum_{i=k-1}^{n} \binom{n}{i} A^i (1-A)^{n-i}}, \tag{7.130}$$

$$\text{MTBF}_s = \frac{A_s \text{MTBR}_s}{1-A_s} = \frac{\sum_{i=k}^{n} \binom{n}{i} A^i (1-A)^{n-i}}{\mu(n-k+1)\binom{n}{k-1} A^{k-1} (1-A)^{n-k+1}}$$

$$= \frac{\sum_{j=0}^{n-k} \binom{n}{j} (\lambda/\mu)^j}{k\lambda \binom{n}{k} (\lambda/\mu)^{n-k}}. \tag{7.131}$$

This formula is easily recalled by remembering the following basic concepts:

1. $A_s = \text{MTBF}_s/(\text{MTBF}_s + \text{MTBR}_s)$.
2. $\text{MTBR} = 1/[\mu(n-k+1)]$ since $n-k+1$ components are under simultaneous repair when the system is down.
3. The number of components working as $t \to \infty$ follows the truncated binomial distribution.
4. $\text{MTBF}_s = A_s \times \text{MTBR}_s/(1-A_s)$.

Exercise

1. Verify equation (7.131).

7.4.3 Systems with Load-Sharing Components

When the working components equally share the load of the system, the failure rate of each component depends on the load that it has to carry. The load that is allocated to each component depends on the number of failed components that exist in the system. Shao and Lamberson [226] provide an analysis of a repairable k-out-of-n:G system with load-sharing components considering imperfect switching. The sensing and switching mechanism is responsible for detection of component failures and the redistribution of the load of the system equally among surviving components. System performance measures such as reliability and availability are analyzed. Several

errors exist in this paper that are corrected by Akhtar [6]. Newton [176] provides an alternative argument for evaluation of the MTTF and MTBF of such systems.

In this section, we consider a repairable k-out-of-n:G system with load-sharing components. For simplicity of analysis, the sensing and switching mechanism is assumed to be perfect. Service is needed whenever the number of failed components in the system changes (when another component is failed or when a failed component is repaired) to redistribute the load of the system. When the sensing and switching mechanism is imperfect, we say that the system has imperfect fault coverage. This will be discussed in a later section in this chapter. Other assumptions are as given in the general model for a repairable k-out-of-n:G system described in Section 7.4.1. We provide expressions of system reliability, availability, MTTF, and MTBF of such systems.

Assumptions

1. The failure rates of all working components are the same. They are dependent on the number of working components in the system.
2. A repaired component is as good as new and immediately shares the load of the system.

Notation

- λ_i: failure rate of each component when there are i failed components, $i = 0, 1, \ldots, n-k$. Generally, we have $\lambda_0 \leq \lambda_1 \leq \cdots \leq \lambda_{n-k}$.

The only difference between the load-sharing system model in this section and the repairable system model with active redundant components discussed in Section 7.4.2 is that the component failure rate is not a constant any more. The failure rate of the system with i failed components, α_i, can be written as

$$\alpha_i = (n-i)\lambda_i, \qquad i = 0, 1, \ldots, n-k. \tag{7.132}$$

The same techniques as used in Section 7.4.2 can be used to derive the required system performance measures.

Exercise

1. Find expressions of $R_s(t)$, MTTF_s, MTBF_s, and A_s.

7.4.4 Systems with both Active Redundant and Cold Standby Components

Morrison and Munshi [170] provide an analysis of a k-out-of-n:G system with additional standby components. The system consists of n active components where at least k of them have to work for the system to work. In addition, there are m spare components available. All components are i.i.d. There are r repair facilities. Repairs are perfect. The lifetime of an active component is exponentially distributed. Repair

time also follows the exponential distribution. Detection of active component failure and switching of a standby component to the active state are perfect and instant. We will use i to represent the number of failed components in the system and to indicate the state of the system. Both cold and hot standby are analyzed by Morrison and Munshi. We will not discuss the hot standby case here as it is exactly the same as if all of the components are active, which has been discussed in previous sections.

Notation

- n: number of active components used
- m: number of standby components or spares
- γ: λ/μ

Since the spare components are in cold standby, we have the following failure rates and repair rates at different system states:

$$\lambda_i = \begin{cases} n\lambda & \text{if } 0 \le i \le m, \\ (n+m-i)\lambda & \text{if } m < i \le n+m, \\ 0 & \text{if } i > n+m, \end{cases} \tag{7.133}$$

$$\mu_i = \begin{cases} i\mu & \text{if } i \le r, \\ r\mu & \text{if } i > r. \end{cases} \tag{7.134}$$

With these system state transition parameters and following the same procedure as outlined in Section 7.4.1, we can develop a set of differential equations for state probabilities. Solving these differential equations, we can obtain expressions of system state probabilities. Letting time go to infinity, we find the following steady-state probabilities:

$$P_i = \begin{cases} \dfrac{n^i}{i!}\gamma^i P_0, & 0 \le i \le \min\{r, m\}, \\ \dfrac{n^i}{r^{i-r}r!}\gamma^i P_0, & r+1 \le i \le m, \\ n^m\gamma^i \dfrac{n!}{(n+m-i)!i!}P_0, & m+1 \le i \le r, \\ \dfrac{1}{r^{i-r}r!}n^m\gamma^i \dfrac{n!}{(n+m-i)!}P_0, & \max\{r, m\}+1 \le i \le n+m. \end{cases} \tag{7.135}$$

If all spares are in hot standby, the following is provided for verification purposes:

$$P_i = \begin{cases} \dfrac{(n+m)!}{(n+m-i)!i!}\gamma^i P_0, & 0 \le i \le r, \\ \dfrac{(n+m)!}{(n+m-i)!r^{i-r}r!}\gamma^i P_0, & i > r. \end{cases}$$

Exercise

1. Verify equation (7.135).

7.5 WEIGHTED k-OUT-OF-n:G SYSTEMS

Wu and Chen [247] propose a variation of the k-out-of-n:G system, called the weighted k-out-of-n:G model. In a weighted k-out-of-n:G system, component i carries a weight of w_i, $w_i > 0$ for $i = 1, 2, \ldots, n$. The total weight of all components is w, $w = \sum_{i=1}^{n} w_i$. The system works if and only if the total weight of working components is at least k, a prespecified value. Since k is a weight, it may be larger than n because they have different measuring units. Such a weighted k-out-of-n:G system is equivalent to a weighted $(w-k+1)$-out-of-n:F system wherein the system fails if and only if the total weight of failed components is at least $w-k+1$. With this definition, the k-out-of-n:G system is a special case of the weighted k-out-of-n:G system wherein each component has a weight of 1.

A recursive equation is provided by Wu and Chen [247]. In the following, $R(i, j)$ represents the probability that a system with j components can output a total weight of at least i. Then, $R(k, n)$ is the reliability of the weighted k-out-of-n:G system. The following recursive equation can be used for reliability evaluation of such systems:

$$R(i, j) = p_j R(i - w_j, j - 1) + q_j R(i, j - 1), \tag{7.136}$$

which requires the following boundary conditions:

$$R(i, j) = 1 \qquad \text{for } i \leq 0,\ j \geq 0, \tag{7.137}$$

$$R(i, 0) = 0 \qquad \text{for } i > 0. \tag{7.138}$$

It should be noted that w_i $(1 \leq i \leq n)$ may not be integer. When $w_i = 1$ for all $1 \leq i \leq n$, we have the usual k-out-of-n:G system. The computational complexity of equation (7.136) is $O(k(n - k + 1))$ when $w_i = 1$ for all i. However, when $w_i > 1$ for all $1 \leq i \leq n$, the number of terms to be computed may be much less than $k(n - k + 1)$, as illustrated in the following example.

Example 7.4 Consider a weighted 5-out-of-3:G system. It has three components with weights 2, 6, and 4. The system works if and only if the total weight of working components is at least 5.

This is a very simple example. We can easily solve the problem without using equation (7.136). The following are the minimal paths of the system:

Component 2 works with a total output of 6.
Components 1 and 3 work with a total output of 6.

Thus, we can find the system reliability as

$$R_s = \Pr(x_2 \cup x_1 x_3) = \Pr(x_2) + \Pr(\overline{x}_2\, x_1 x_3) = p_2 + q_2 p_1 p_3.$$

If we apply equation (7.136), here are the terms to be calculated:

$$R(1,1) = p_1 R(-1,0) + q_1 R(1,0) = p_1,$$

$$R(5,1) = p_1 R(3,0) + q_1 R(5,0) = 0,$$

$$R(1,2) = p_2 R(-5,1) + q_2 R(1,1) = p_2 + p_1 q_2,$$

$$R(5,2) = p_2 R(-1,1) + q_2 R(5,1) = p_2,$$

$$R(5,3) = p_3 R(1,2) + q_3 R(5,2) = p_3(p_2 + p_1 q_2) + q_3 p_2 = p_2 + q_2 p_1 p_3.$$

The total number of terms calculated is only five, much less than $(n+1)(k+1) = 20$.