IIE Transactions (2003) **35**, 527–534
Copyright © 2003 "IIE"
0740-817X/03 $12.00+.00
DOI: 10.1080/07408170390193053

Multi-state consecutive-*k*-out-of-*n* systems

JINSHENG HUANG[1], MING J. ZUO[1,*] and ZHIDE FANG[2]

[1]*Department of Mechanical Engineering, University of Alberta, Edmonton, Alberta, Canada, T6G 2G8*
E-mail: hjinshen@ualberta.ca or ming.zuo@ualberta.ca
[2]*Department of Mathematics, University of New Orleans, New Orleans, LA, 70148, USA*
E-mail: zfang@math.uno.edu

Received October 2000 and accepted July 2002

In the binary context, a consecutive-*k*-out-of-*n* system is failed if and only if at least *k* consecutive components are failed. In this paper we propose definitions of the multi-state consecutive-*k*-out-of-*n*:F and G systems. In the proposed definition, both the system and its components may be in one of $M+1$ possible states: 0, 1, . . ., and M. The dual relationship between the proposed systems is identified. The concept of dominance is used to characterize the properties of multi-state systems. The concepts of duality, equivalence, and dominance are used in evaluation of system state distribution of multi-state consecutive-*k*-out-of-*n* systems. An algorithm is provided for evaluating system state distribution of decreasing multi-state consecutive-*k*-out-of-*n*:F systems. Another algorithm is provided to bound system state distribution of multi-state consecutive-*k*-out-of-*n*:F and G systems. Several examples are included to illustrate the proposed definitions, concepts, and algorithms.

1. Introduction

In traditional reliability theory, both the system and its components are allowed to take two possible states: either working or failed. In a multi-state system, both the system and its components may experience more than two possible states, for example, completely working, partially working, and completely failed. A multi-state system reliability model provides more flexibility for modeling of equipment conditions.

In the binary context, a system with n components in sequence is called a consecutive-k-out-of-n:F (G) system if the system fails (works) whenever at least k consecutive components in the system fail (work). A consecutive-n-out-of-n:F (G) system becomes a parallel (series) system. A consecutive-one-out-of-n:F (G) system becomes a series (parallel) system. Many research results have been reported on reliability evaluation of binary consecutive-k-out-of-n system, for example, see Chiang and Niu (1981), Hwang (1982), Kuo *et al.* (1994) and Chao *et al.* (1995). The dual relationship between the consecutive-k-out-of-n:F and G systems is investigated by Kuo *et al.* (1990) and Zuo (1993).

In a multi-state system, both the system and the components are allowed to be in one of $M+1$ possible states, 0, 1, 2, . . ., and M, where M is the perfect functioning state while zero is the complete failure state. We use x_i to denote the state or performance level of component i ($i = 1, 2, \ldots, n$). Vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ represents the states of all the n components. The system state denoted by ϕ is a deterministic function of component states $x_1, x_2, \ldots$, and x_n. Thus, $\phi = \phi(\mathbf{x})$, where $\mathbf{x}$ takes values in $\mathbf{S}^n$, ϕ takes values in $\mathbf{S}$, and $\mathbf{S} = \{0, 1, 2, \ldots, M\}$.

Recently, researchers have partially extended the definitions of the binary consecutive-k-out-of-n system to the multi-state case by allowing the system to remain binary and its components to have more than two possible states, for example, see Zuo and Liang (1994) and Malinowski and Preuss (1995, 1996). Koutras (1997) extends the binary consecutive-k-out-of-n:F system to the dual failure mode environment whereas the system and each component may experience one working state and two different failure states. Aki (1992) analyzes the waiting time for a run of a specific outcome of a specific length from a sequence of non-negative integer valued random variables.

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

---

*Corresponding author

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

A definition of the generalized multi-state k-out-of-n:G system, in which k could take different values for different system state levels, has been proposed by Huang *et al.* (2000). In this paper, we propose a definition of a multi-state consecutive-k-out-of-n system. Under the proposed definition, a possibly different number of consecutive components need to be below level j for the system to be below level j for different j values. The required number of consecutive component "failures" is dependent on the system state level under consideration. Examples are given to illustrate the proposed definition. System performance evaluation algorithms are presented for the proposed multi-state consecutive-k-out-of-n:F system. In our discussion of multi-state systems, we sometimes still use the term "failure". This term now has a dynamic meaning. When we are concerning about state level j, we can say the system is "failed" or a component is "failed" if it is below state level j. Of course when the value of j changes, the meaning of "failed" also changes.

2. Notation and assumptions

2.1. *Notation*

n = number of components;
$M+1$ = number of states of the system or its components;
x_i = state of component i, $x_i \in \{0, 1, \dots M\}$, $i = 1, 2, \dots, n$;
$\mathbf{x}$ = vector of component states, $\mathbf{x} = (x_1, x_2, \dots, x_n)$;
$\phi(\mathbf{x})$ = system structure function representing the state of the system, $\phi(\mathbf{x}) \in \{0, 1, \dots, M\}$;
$\phi^{\mathrm{D}}(\mathbf{x})$ = the dual of ϕ;
k_j = minimum number of consecutive components to be in states below j;
P_{ij} = $\Pr(x_i \geq j)$;
P_j = $P_j = P_{ij}$ when the components are iid;
p_{ij} = $\Pr(x_i = j)$;
p_j = $p_j = p_{ij}$ when the components are iid;
Q_{ij} = $\Pr(x_i < j)$;
Q_j = $Q_j = Q_{ij}$ when the components are iid;
$F_j(n; k_j)$ = probability that at least k_j consecutive components are in states below j for an n component system;
$R_j(n; k_j)$ = $1 - F_j(n; k_j)$;
R_{sj} = $\Pr(\phi \geq j)$;
F_{sj} = $1 - R_{sj}$;
r_{sj} = $\Pr(\phi = j)$.

2.2. *Nomenclature*

Multi-state minimal path vector: Vector $\mathbf{y} \in \mathbf{S}^n$ is a minimal path vector to system state level j if and only if $\phi(\mathbf{y}) \geq j$ and $\phi(\mathbf{x}) < j$ for all $\mathbf{x} < \mathbf{y}$ (Boedigheimer and Kapur, 1994).

Multi-state minimal cut vector: Vector $\mathbf{y} \in \mathbf{S}^n$ is a minimal cut vector to system state level j if and only if $\phi(\mathbf{y}) < j$ and $\phi(\mathbf{x}) \geq j$ for all $x > \mathbf{y}$ (Boedigheimer and Kapur, 1994).

2.3. *Assumptions*

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

3. The multi-state consecutive-*k*-out-of-*n*:F system

We propose the following definition of the multi-state consecutive-k-out-of-n systems:

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

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

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

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

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

Example 2. (A decreasing multi-state consecutive-k-out-of-n system.) Consider a three-component system wherein both the system and the components may be in one of three possible states, state 0, state 1, and state 2. The system state and the component states have the relationships as shown in Table 1. In this example, we have $k_1 = 2$ and $k_2 = 1$.

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

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

Example 3. (An increasing multi-state consecutive-k-out-of-n system.) Consider a three-component system with $k_1 = 1$, $k_2 = 2$, and $k_3 = 3$. Both the system and the components may be in one of four possible states, namely, state 0, state 1, state 2, and state 3. Table 2 illustrates the relationship between system state and component states.

Table 1. System structure function for Example 2

$\phi(x)$	*0*	*1*	*2*
x	(0, 0, 0)	(0, 1, 0)	(2, 2, 2)
	(1, 0, 0)	$(1, 1, 0)^+$	
	(0, 0, 1)	(1, 1, 1)	
	(2, 0, 0)	(0, 2, 0)	
	(0, 0, 2)	$(2, 1, 1)^+$	
		$(2, 2, 0)^+$	
		$(2, 2, 1)^+$	

Table 2. System structure function for Example 3

$\phi(x)$	*0*	*1*	*2*	*3*
x	(0, 0, 0)	(1, 1, 1)	(0, 2, 0)	$(3, 0, 0)^+$
	$(1, 0, 0)^+$	(2, 1, 1)	(1, 2, 0)	$(3, 1, 0)^+$
	$(1, 1, 0)^+$	(1, 1, 2)	(0, 2, 1)	$(3, 1, 1)^+$
	(2, 0, 0)		$(2, 2, 0)^+$	$(3, 2, 0)^+$
	(0, 0, 2)		$(2, 2, 1)^+$	$(3, 2, 1)^+$
	(2, 1, 0)		(2, 2, 2)	$(3, 2, 2)^+$
	(2, 0, 1)			$(3, 3, 0)^+$
	(1, 0, 2)			$(3, 3, 1)^+$
	(0, 1, 2)			$(3, 3, 2)^+$
				(3, 3, 3)

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

In a binary system, we are interested in the reliability of a system given its components' reliabilities. In the multi-state context, we assume that the probabilities for each component to be in different states are known. We are interested in finding the probabilities for the system to be in different states. We will use the term *component state distribution* to indicate the probabilities that the component is in different states and the term *system state distribution* to indicate the probabilities that the system is in different states.

A commonly used approach for evaluation of a multi-state system state distribution is to extend the results for binary system reliability evaluation. Usually, a multi-state system can be divided into "functioning" with respect to state level j if $\phi(\mathbf{x}) \geq j$ and "failed" otherwise. Similarly, component i is said "functioning" with respect to state level j when $x_i \geq j$ and "failed" otherwise. However, this dichotomous method can only be applied to some multi-state systems. In the following, we discuss methods for evaluation of system state distribution of multi-state consecutive-k-out-of-n:F systems.

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

In this case, the definition of a multi-state consecutive-k-out-of-n system is equivalent to that $\phi(\mathbf{x}) < j$ if and only if at least k_j consecutive components have $x_i < j$ for $j = 1, 2, \ldots, M$. Those components with $x_i < j$ are considered "failed" with respect to state level j. The system is considered "failed" with respect to state level j if $\phi < j$. The algorithms for binary consecutive-k-out-of-n system reliability evaluation provided by Hwang (1982) can be used to find $\Pr(\phi < j)$ for $j = 1, 2, \ldots, M$ of a multi-state consecutive-k-out-of-n system. Once $\Pr(\phi < j)$ is found for $j = 1, 2, \ldots, M$, we can easily find $\Pr(\phi = j)$ for $j = 0, 1, \ldots, M$. The following equation is based on Hwang (1982):

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

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

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

$$P_{0j} = 1, \quad j = 1, 2, \ldots, M. \tag{3}$$

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

$$\Pr(\phi < j) = F_j(n;k_j), \quad \text{for } j = 1, 2, \ldots, M, \tag{4}$$

$$\Pr(\phi = 0) = \Pr(\phi < 1), \tag{5}$$

$$\Pr(\phi = M) = 1 - \Pr(\phi < M), \tag{6}$$

$$\Pr(\phi = j) = \Pr(\phi < j+1) - \Pr(\phi < j), \quad \text{for } j = 1, \ldots, M-1. \tag{7}$$

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

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

We assume that at least one of the inequalities in this case is a strict inequality. Based on the definitions of a minimal path vector provided earlier in this paper, a minimal path vector to level j could also be a minimal path vector to level $j+1$ or even higher levels. Consider the system structure given in Example 3. One of the minimal path vectors to system level 1 is (0, 2, 0) because $\phi(0, 2, 0) \geq 1$ and $\phi(\mathbf{x}) < 1$ for all $\mathbf{x} < (0, 2, 0)$. At the same time, component state vector (0, 2, 0) is a minimal path vector for system state level 2. As a result, we are unable to use binary consecutive-k-out-of-n:F system reliability evaluation formulas for evaluation of system state distribution under case II.

Before more efficient system performance evaluation algorithms can be found, one may use the minimal path or minimal cut vectors and the Sum of Disjoint Product (SDP) method for evaluation of system state distribution (for example, see Locks (1987)). In the next section, we report a method for evaluating system state distribution of a multi-state consecutive-k-out-of-n:G system.

4. The multi-state consecutive-k-out-of-n:G system

In the binary context, a consecutive-k-out-of-n:G system works if and only if at least k consecutive components work (Kuo *et al.*, 1990). In the following, we propose a definition of the multi-state consecutive-k-out-of-n:G system.

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

The condition in this definition can also be phrased as follows: $\phi(\mathbf{x}) \geq j$ $(j = 1, 2, \cdots, M)$ if at least k_j consecutive components are in state j or above; at least k_{j-1} consecutive components are in state $j-1$ or above; $\cdots$; and at least k_1 consecutive components are in state 1 or above.

In the binary context, the dual of a system ϕ is defined as: $\phi^{\mathrm{D}}(\mathbf{x}) = 1 - \phi(\mathbf{1} - \mathbf{x})$ where $\mathbf{1} = (1, 1, \ldots, 1)$. The consecutive-$k$-out-of-$n$:F and G systems are duals of each other (Kuo *et al.*, 1990). For multi-state systems, a similar definition of duality is provided by El-Neweihi *et al.* (1978):

$$\phi^{\mathrm{D}}(\mathbf{x}) = M - \phi(\mathbf{M} - \mathbf{x}), \tag{8}$$

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

$$\Pr(\phi^{\mathrm{D}}(\mathbf{x}) = j) = \Pr(\phi(\mathbf{M} - \mathbf{x}) = M - j). \tag{9}$$

Based on Equation (9), if we let $p_{ij} = p^{\mathrm{D}}_{i(M-j)}$ for $i = 1, 2, \ldots, n$ and $j = 0, 1, \ldots, M$, then the probability for the primal system to be in state j is equal to the probability for the dual system to be in state $M - j$ for $j = 0, 1, \ldots, M$.

Based on Equation (9), the multi-state consecutive-k-out-of-n:F and consecutive-k-out-of-n:G systems given in Definitions 1 and 2 are duals of each other. More specifically, the dual of an increasing multi-state consecutive-k-out-of-n:F system is a decreasing multi-state consecutive-k-out-of-n:G system. Because there are no efficient algorithms for system performance evaluation of an increasing multi-state consecutive-k-out-of-n:F system, we may use this dual relationship for efficient system performance evaluation of the increasing multi-state consecutive-k-out-of-n:F systems.

Hudson and Kapur (1983) propose the concept of *equivalence*. Two component state vectors $\mathbf{x}$ and $\mathbf{y}$ are said to be equivalent if $\phi(\mathbf{x}) = \phi(\mathbf{y})$ and this equivalence relationship is indicated by $\mathbf{x} \leftrightarrow \mathbf{y}$. All component state vectors that are equivalent to one another belong to the same equivalence class. Huang and Zuo (2002) propose the following definition of *dominance*.

Definition 3. A multi-state monotone system is called a dominant system if its structure function ϕ satisfies: $\phi(\mathbf{y}) > \phi(\mathbf{x})$ implies either: (i) $\mathbf{y} > \mathbf{x}$, or (ii) $\mathbf{y} > \mathbf{z}$, $\mathbf{x} \leftrightarrow \mathbf{z}$, and $\mathbf{x} \neq \mathbf{z}$.

We will use the following example to illustrate the concepts of duality, equivalence, and dominance.

Example 4. (Relationship between a primal system and its dual.) Consider the system structure given in Example 3 as the primal system, which is an increasing consecutive-k-out-of-n:F system. By applying Equation (8) to each component state vector in Example 3, we obtain the structure function of its dual system as shown Table 3.

The "$(\mathbf{x})^+$" sign in Table 3 represents all permutations of the elements of the component state vector $\mathbf{x}$. From examination of the structure function shown in Table 3, we find that this dual system becomes a decreasing multi-state consecutive-k-out-of-n:G system. The system is at state 1 or above if at least three consecutive components are at state 1 or above. The system is at state 2 or above if at least two consecutive components are at state 2 or above and at least three components are at state 1 or above. The system is at state 3 (or above) if at least one component is at state 3 (or above) and at least two consecutive components are at state 2 or above and at least three components are at state 1 or above. We notice that $k_1 = 3$, $k_2 = 2$, and $k_3 = 1$ and we have to use the "and" relationship to specify the conditions that should be satisfied for the system to be at a certain state or above.

Using the definition of dominance, we can also conclude that the primal system shown in Example 3 is a non-dominant system. For example, $\phi(0, 2, 0) = 2$, but at the same time, component state vector (0, 2, 0) is not greater than any component state vector that results in system state 1. On the other hand, the dual system shown in Example 4 is a dominant system because each component state vector in a column is greater than at least one component state vector in the neighboring column on the left. For example, $\phi^{\mathrm{D}}(2, 2, 1) = 2$, $\phi^{\mathrm{D}}(1, 1, 1) = 1$, and $(2, 2, 1) > (1, 1, 1)$.

Table 3. Structure function of the dual of the system given in Example 3

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

As we have seen from Examples 3 and 4, the dual of an increasing multi-state consecutive-k-out-of-n:F system becomes a decreasing multi-state consecutive-k-out-of-n:G system. Based on its definition, we know that a decreasing multi-state consecutive-k-out-of-n:G system has the following properties:

1. $n \equiv k_0 \geq k_1 \geq k_2 \geq \ldots \geq k_M$,
2. It is a dominant system,
3. The minimal path vectors to system level j will cause the system to be exactly in level j,
4. One of the minimal path vectors to level j is in the following form:

$$(\underbrace{\underbrace{\underbrace{\underbrace{j, \ldots, j}_{k_j}, j-1, \ldots, j-1}_{k_{j-1}}, j-2, \ldots, 1, \ldots, 1}_{k_1}, 0, \ldots, 0}_{n}), \tag{10}$$

   where the number of elements taking the value of i is equal to $k_i - k_{i+1} \geq 0$ for $i = 1, \ldots, j-1$.
5. Every minimal path vector to level j can be obtained by permutating the elements of this minimal path shown in (10).

Because the set of minimal path vectors to level j contains some permutations of the vector shown in (10), it is easy to find all these minimal path sets to level j. For example, suppose that a system has five components and a minimal path set to level 3 is (3, 3, 2, 1, 0). From this minimal path set, we know that for the system to be at level 3 or above, at least two consecutive components must be in level 3 or above, at least three consecutive components have to be in state 2 or above, at least four consecutive components have to be in state 1 or above, and all five components have to be in state 0 or above. We also note that the last condition that all five components have to be in state 0 or above may be omitted because it is automatically satisfied. This way, we can easily find all minimal path vectors to system level 3, as listed below:

(3, 3, 2, 1, 0), (2, 3, 3, 1, 0), (1, 3, 3, 2, 0), (0, 3, 3, 2, 1),
(0, 1, 2, 3, 3), (0, 1, 3, 3, 2), (0, 2, 3, 3, 1), (1, 2, 3, 3, 0).

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

j or above. In the following, we present a method to bound $\Pr(\phi \geq j)$ when binary algorithms cannot be used.

Suppose we have a minimal path vector for system state level j, denoted by $\mathbf{y}$, which is in the form shown in Equation (10). We will use $\mathbf{y}_j^*$ to represent all minimal path vectors to system state level j. Then, we have

$$\Pr(\phi \geq j) = \Pr(\mathbf{x} \geq \mathbf{y}_j^*), \tag{11}$$

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

$$s = \min\{i | i \in \mathbf{y}, i < j\}, \tag{12}$$

$$t = \max\{i | i \in \mathbf{y}, i < j\}. \tag{13}$$

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

$$\mathbf{L} = (\underbrace{\underbrace{j, \cdots, j}_{k_j}, s, \cdots, s}_{n}), \tag{14}$$

$$\mathbf{U} = (\underbrace{\underbrace{j, \cdots, j}_{k_j}, t, \cdots, t}_{n}). \tag{15}$$

Obviously, we have

$$\mathbf{L} \leq \mathbf{y} \leq \mathbf{U}. \tag{16}$$

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

$$\Pr(\mathbf{x} \geq \mathbf{U}_j^*) \leq \Pr(\phi \geq j) = \Pr(\mathbf{x} \geq \mathbf{y}_j^*) \leq \Pr(\mathbf{x} \geq \mathbf{L}_j^*). \tag{17}$$

Inequality (17) can be used to find upper and lower bounds on the probability that the system is in state j or above. When $s = t$, the two bounds are the same and equal to the exact value of $\Pr(\phi \geq j)$. The question remaining to be answered is how to evaluate the two bounds in the inequality (17).

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

Case I. The components are iid.

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

$$\text{Lower bound} = \Pr(\mathbf{x} \geq \mathbf{U}_j^*) = \sum_{k=k_j}^{n} (n-k+1) P_j^k (1 - P_j - Q_t)^{n-k}, \tag{18}$$

$$\text{Upper bound} = \Pr(\mathbf{x} \geq \mathbf{L}_j^*) = \sum_{k=k_j}^{n} (n-k+1) P_j^k (1 - P_j - Q_s)^{n-k}. \tag{19}$$

Case II. The components are independent.

When the components in a system are independent, Hwang (1982) provides an algorithm for reliability evaluation of a binary consecutive-k-out-of-n:F system. Kuo *et al.* (1990) provide a corresponding system reliability evaluation formula for a binary consecutive-k-out-of-n:G system. To avoid notation confusion with what has been defined in this paper, we will use u_i and v_i to represent the working and failure probabilities of component i in the binary system for $i = 1, 2, \ldots, n$. The formulas given by Hwang (1982) and Kuo *et al.* (1990) are for binary systems and require the following condition to be satisfied for every component i:

$$u_i + v_i = 1, \quad i = 1, 2, \ldots, n. \tag{20}$$

To enable our analysis of multi-state consecutive-k-out-of-n:G system, we now propose to relax the requirement shown in Equation (20) into the following:

$$0 \leq u_i + v_i \leq 1, \quad i = 1, 2, \ldots, n. \tag{21}$$

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

$$R(n;k) = v_n R(n-1;k) + u_n \left[\left(\prod_{j=n-k+1}^{n-1} u_j \right) R^*(n-k) + \sum_{i=n-k+1}^{n-1} v_i \left(\prod_{j=i+1}^{n-1} u_j \right) R(i-1;k) \right], \tag{22}$$

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

The derivation of Equation (22) is based on the Bayes theorem. The first term $v_n R(n-1;k)$ is the probability that component n is "failed" and at the same time at least k consecutive components out of the remaining $(n-1)$ components are "working." The second term $u_n \times (\bullet)$ represents the probability that component n is "working" and the system is "working". In the brackets, the first term represents the case that components $(n-1)$, $(n-2)$, $\cdots$, and $(n-k+1)$ are all "working" and the remaining $n-k$ components are either "working" or "failed" (but each component with a probability of $u_i + v_i < 1$). The second term in the brackets represents the probability that component i $(n-k+1 \le i \le n-1)$ is the first "failed" component counting from component $n-1$ downward and the $i-1$ component subsystem including components 1 through $i-1$ is "working".

Equation (22) is a recursive formula. The following boundary conditions are needed.

$$R(k;k) = u_1 u_2 \cdots u_k, \tag{23}$$

$$R(a;b) = 0, \quad \text{for } b > a > 0. \tag{24}$$

When the minimal path vectors have at most two different elements, Equation (22) provides the exact measure of the probability for the system to be in state j or above. When the number of different elements in the minimal path vectors is greater than two, Equation (22) can be used to find the upper bounds and lower bounds for $\Pr(\phi \ge j)$ for $j = 1, 2, \ldots, M$ in a multi-state consecutive-k-out-of-n:G system. The procedure for using Equation (22) is outlined below.

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

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

For upper bound calculation, define

$$v_i = 1 - u_i - Q_{is}, \quad i = 1, 2, \ldots, n, \tag{25}$$

where s is given in Equation (12). Equation (22) can then be used to derive the lower bound for $\Pr(\phi \ge j)$. For lower bound calculation, define

$$v_i = 1 - u_i - Q_{it}, \quad i = 1, 2, \ldots, n, \tag{26}$$

where t is given in Equation (13). Equation (22) can then be used to calculate the upper bound for $\Pr(\phi \ge j)$. The following example illustrates this procedure.

Example 5. Consider a multi-state consecutive-k-out-of-n:G system with the following given data:

$$\begin{gathered} n = 3,\ k_1 = 3,\ k_2 = 2,\ k_3 = 1, \\ p_{10} = 0.1,\ p_{11} = 0.2,\ p_{12} = 0.3,\ p_{13} = 0.4, \\ p_{20} = 0.2,\ p_{21} = 0.2,\ p_{22} = 0.3,\ p_{23} = 0.3, \\ p_{30} = 0.1,\ p_{31} = 0.1,\ p_{32} = 0.2,\ p_{33} = 0.6, \\ P_{10} = 1,\ P_{11} = 0.9,\ P_{12} = 0.7,\ P_{13} = 0.4, \\ P_{20} = 1,\ P_{21} = 0.8,\ P_{22} = 0.6,\ P_{23} = 0.3, \\ P_{30} = 1,\ P_{31} = 0.9,\ P_{32} = 0.8,\ P_{33} = 0.6. \end{gathered}$$

For system state level 1, the only minimal path vector is (1, 1, 1). Let $u_i = P_{i1}$ and $v_i = p_{i0}$ for $i = 1, 2, 3$. Using Equation (22) or the reliability formula for a series system, we have

$$R_{s1} = P_{11} \times P_{21} \times P_{31} = 0.9 \times 0.8 \times 0.9 = 0.648.$$

This is the exact probability for the system to be in state 1 or above.

For system state level 2, $k = 2$ and the only minimal path vectors are (2, 2, 1) and (1, 2, 2). Since these minimal path vectors have only two different element values, Equation (22) can provide the exact value of R_{s2}. Let $u_i = P_{i2}$ and $v_i = p_{i1}$ for $i = 1, 2, 3$. Using Equation (22), we have

$$\begin{aligned} R_{s2} &= R(3;2) = v_3 R(2;2) + u_3[u_2 R^*(1) + 0], \\ &= p_{31} P_{22} P_{12} + P_{32} P_{22}(p_{11} + P_{12}), \\ &= 0.1 \times 0.7 \times 0.6 + 0.8 \times 0.6(0.2 + 0.7) = 0.474. \end{aligned}$$

For system state level 3, $k = 1$ and the only minimal path vectors are (3, 2, 1), (2, 3, 1), (1, 2, 3), and (1, 3, 2). Select $\mathbf{L} = (3, 1, 1)$ and $\mathbf{U} = (3, 2, 2)$. For $\mathbf{L} = (3, 1, 1)$, $s = 1$ and let $u_i = P_{i3}$ and $v_i = p_{i1} + p_{i2}$. Using Equation (22), we have:

$$\begin{aligned} \text{Upper bound} &= R(3;1) = v_3 R(2;1) + u_3[R^*(2) + 0], \\ &= v_3[v_2 R(1;1) + u_2 R^*(1)] + u_3 R^*(2), \\ &= (p_{31} + p_{32})[(p_{21} + p_{22}) R(1;1) + P_{23} R^*(1)] \\ &\quad + P_{33} R^*(2), \\ &= 0.3 \times [0.5 \times 0.4 + 0.3 \times 0.9] + 0.6 \\ &\quad \times 0.9 \times 0.8 = 0.573. \end{aligned}$$

For $\mathbf{U} = (3, 2, 2)$, $t = 2$ and let $u_i = P_{i3}$ and $v_i = p_{i2}$.

$$\begin{aligned} \text{Lower bound} &= R(3;1) = v_3 R(2;1) + u_3[R^*(2) + 0], \\ &= v_3[v_2 R(1;1) + u_2 R^*(1)] + u_3 R^*(2), \\ &= p_{32}[p_{22} R(1;1) + P_{23} R^*(1)] + P_{33} R^*(2), \\ &= 0.2 \times [0.3 \times 0.4 + 0.3 \times 0.7] \\ &\quad + 0.6 \times 0.7 \times 0.6 = 0.318. \end{aligned}$$

Now we have:

$$0.318 \le R_{s3} \le 0.573.$$

If we use the average of the upper and lower bounds to approximate R_{s3}, then we have

$$R_{s3} \approx 0.4455.$$

The exact probability for the system to be in state 3 (or above) has been calculated with the enumeration method:

$$R_{s3} = 0.435.$$

The relative error of the approximated value for R_{s3} is about 2.4%.

The sharpness of the bounds provided by Equation (22) depends on many parameters such as k_j $(j = 1, 2, \ldots, M)$ and p_{ij} $(i = 1, 2, \ldots, n, j = 1, 2, \ldots, M)$. Once additional bounds formulas are developed, the relative sharpness of different bounds may be compared. Additional research is needed to develop other bounds formulas.

Acknowledgments

This research was partially supported by Natural Sciences and Engineering Research Council of Canada. Comments and suggestions from referees and the editor are very much appreciated.

References

Aki, S. (1992) Waiting time problems for a sequence of discrete random variables. *Annals of the Institute of Statistical Mathematics*, **44**(2), 263–378.

Boedigheimer, R.A. and Kapur, K.C. (1994) Customer-driven reliability models for multi-state coherent systems. *IEEE Transactions on Reliability*, **43**(1), 46–50.

Chao, M.T., Fu, J.C. and Koutras, M.V. (1995) Survey of reliability studies of consecutive-k-out-of-n:F & related Systems. *IEEE Transactions on Reliability*, **44**(1), 120–125.

Chiang, D.T. and Niu, S. (1981) Reliability of consecutive-k-out-of-n:F system. *IEEE Transactions on Reliability*, **30**(1), 87–89.

El-Neweihi, E., Proschan, F. and Sethuraman, J. (1978) Multi-state coherent system. *Journal of Applied Probability*, **15**, 675–688.

Griffith, W.S. (1980) Multistate reliability models. *Journal Applied Probability*, **17**, 735–744.

Haim, M. and Porat, Z. (1991) Bayes reliability modeling of a multi-state consecutive-k-out-of-n:F system, in *Proceedings of the Annual Reliability and Maintainability Symposium*, pp. 582–586.

Huang, J. and Zuo, M. (2002) Dominant multi-state systems. *IEEE Transactions on Reliability*, (to appear).

Huang, J., Zuo, M. and Wu, Y. (2000) Generalized multi-state k-out-of-n:G system. *IEEE Transactions on Reliability*, **49**(1), 105–111.

Hudson, J.C. and Kapur, K.C. (1983) Reliability analysis for multistate systems with multistate components. *IIE Transactions*, **15**(2), 127–135.

Hwang, F.K. (1982) Fast solutions for consecutive-k-out-of-n:F system. *IEEE Transactions on Reliability*, **31**(5), 447–448.

Koutras, M.V. (1997) Consecutive-k,r-out-of-n:DFM systems. *Microelectronics and Reliability*, **37**(4), 597–603.

Kuo, W., Fu, J.C. and Lou, W.Y. (1994) Opinions on consecutive-k-out-of-n:F system. *IEEE Transactions on Reliability*, **43**(4), 659–662.

Kuo, W., Zhang, W. and Zuo, M. (1990) A consecutive-k-out-of-n:G system: the mirror image of a consecutive-k-out-of-n:F system. *IEEE Transactions on Reliability*, **39**(2), 244–253.

Locks, M.O. (1987) A minimizing algorithm for sum of disjoint products. *IEEE Transactions on Reliability*, **36**, 445–453.

Malinowski, J. and Preuss, W. (1995) Reliability of circular consecutively-connected systems with multi-state components. *IEEE Transaction on Reliability*, **44**(3), 532–534.

Malinowski, J. and Preuss, W. (1996) Reliability of reverse-tree-structured systems with multi-state components. *Microelectronics Reliability*, **36**(1), 1–7.

Zuo, M. (1993) Reliability and component importance of a consecutive-k-out-of-n system. *Microelectronic Reliability*, **33**(2), 243–258.

Zuo, M.J. and Liang, M. (1994) Reliability of multistate consecutively-connected systems. *Reliability Engineering and System Safety*, **44**, 173–176.

Biographies

Jinsheng Huang received his B.S. (1982) and M.S. (1988) in Mechanical Engineering from the University of Science and Technology Beijing, China. He received his Ph.D. (2001) in Mechanical Engineering from the University of Alberta. His research interests include multi-state system reliability, maintenance decision analysis, and mechanical design. He is currently a mechanical engineer at Stantec Consulting Ltd., Edmonton, Alberta, Canada.

Ming J. Zuo received a B.S. in Agricultural Engineering from the Shandong Institute of Technology, China, in 1982, an M.S. degree in 1986 and a Ph.D. in 1989 both in Industrial Engineering from Iowa State University. He worked at the University of Windsor in Canada for a year in 1989. Since 1990 he has been with the Department of Mechanical Engineering at the University of Alberta in Canada where he is currently a Professor. His research interests include system reliability modeling, manufacturing systems, and applied operations research. He is on the Editorial Board of the journal *IIE Transactions on Quality and Reliability Engineering*. He is also a senior member of IEEE and IIE.

Zhide Fang is an Assistant Professor in the Department of Mathematics at the University of New Orleans. He received an M.S. in Statistics from the Central China Normal University in China and a Ph.D. in Statistics from the University of Alberta, Edmonton, Canada. His research interests include experimental design, statistical computation, reliability analysis, and wavelets application in statistics.

Contributed by the Reliability Modeling and Optimization Department