IEEE TRANSACTIONS ON RELIABILITY, VOL. 49, NO. 1, MARCH 2000

Generalized Multi-State k-out-of-n:G Systems

Jinsheng Huang, Ming J. Zuo, *Member, IEEE,* and Yanhong Wu

***Abstract*—In a binary k-out-of-n:G system, k is the minimum number of components that must work for the system to work. Let 1 represent the working state and 0 the failure state, k then indicates the minimum number of components that must be in state 1 for the system to be in state 1. This paper defines the multi-state k-out-of-n:G system: each component and the system can be in 1 of $M+1$ possible states: $0, 1, \cdots, M$. In Case I, the system is in state $\geq j$ iff at least k_j components are in state $\geq j$. The value of k_j can be different for different required minimum system-state level j. Examples illustrate applications of this definition. Algorithms for reliability evaluation of such systems are presented.**

***Index Terms*—k-out-of-n system, multi-state, reliability evaluation.**

I. INTRODUCTION

Acronym:

MS	multi-state

Notation:

n	number of components in the system
$M+1$	number of states of the system and its components, state M: perfect-functioning, state 0: complete-failure
$\boldsymbol{S}$	$\{0, 1, 2, \cdots, M\}$
x_i	state of component i, $x_i \in \boldsymbol{S}$, $i \in \{1, 2, \cdots, n\}$
$\boldsymbol{x}$	$(x_1, x_2, \cdots, x_n)$: vector of component states
$\phi(\boldsymbol{x})$	system-state structure-function: $\phi(\boldsymbol{x}) \in \boldsymbol{S}$
k_j	minimum number of components with $x_i \geq j$, $i \in \{1, 2, \cdots, n\}$
$P_{i,j}$	$\Pr\{x_i \geq j\}$
$Q_{i,j}$	$1 - P_{i,j} = \Pr(x_i < j)$
P_j	$P_{i,j}$ when components are i.i.d.
Q_j	$Q_{i,j}$ when components are i.i.d.
$p_{i,j}$	$\Pr\{x_i = j\}$
p_j	$p_{i,j}$ when components are i.i.d.
$R_{s,j}$	$\Pr\{\phi(\boldsymbol{x}) \geq j\}$
$r_{s,j}$	$\Pr\{\phi(\boldsymbol{x}) = j\}$
$R_j(a, b)$	$\Pr\{$at least b out of a components are at state j or above$\}$
$R_e^j(b, a)$	$\Pr\{$exactly b components are in state j, and the other $a-b$ components are below state $j\}$

Nomenclature: MS minimal path: $\boldsymbol{y} \in \boldsymbol{S}^n$ is a minimal path to system-state level j iff $\phi(\boldsymbol{y}) \geq j$ and $\phi(\boldsymbol{x}) < j$ for all $\boldsymbol{x} < \boldsymbol{y}$.

MS minimal cut: $\boldsymbol{y} \in \boldsymbol{S}^n$ is a minimal cut to system-state level j iff $\phi(\boldsymbol{y}) < j$ and $\phi(\boldsymbol{x}) \geq j$ for all $\boldsymbol{x} > \boldsymbol{y}$.

Binary system: The system and its components are in 1 of 2 possible states: working or failed.

MS system: Both the system and its components can have more than 2 states, *eg*, completely working, partially working, partially failed, completely failed.

The terms *series & parallel* are used in their logic-diagram sense, irrespective of the schematic-diagram or physical-layout.
◄

A MS system model is a more flexible tool (than a binary system) for representing engineering systems.

In a binary system, a k-out-of-n:G system with n components works iff at least k components work. Both series and parallel systems are special cases of the k-out-of-n:G systems:

- a series system is n-out-of-n:G;
- a parallel system is 1-out-of-n:G.
- A k-out-of-n:G system has $\binom{n}{k}$ minimal paths and $\binom{n}{k-1}$ minimal cuts.
- A k-out-of-n:F system with n components fails iff at least k components fail.
- The k-out-of-n:G and $(n-k+1)$-out-of-n:F systems are equivalent.

In a MS system, both the system and components are in 1 of $M+1$ possible states. ϕ is a deterministic function of $\boldsymbol{x}$. Thus, $\phi(\boldsymbol{x})$ takes values in $\boldsymbol{S}$.

Let there be 2 component-state vectors $\boldsymbol{x}, \boldsymbol{y}$. Then [3]:

- $\boldsymbol{y} < \boldsymbol{x}$ if $y_i \leq x_i$ for all i, and $y_i < x_i$ for at least one i.
- $\boldsymbol{y} > \boldsymbol{x}$ if $y_i \geq x_i$ for all i and $y_i > x_i$ for at least one i.
- $\boldsymbol{x}$ is a lower boundary point to system state level j iff $\phi(\boldsymbol{x}) \geq j$ and $\boldsymbol{y} < \boldsymbol{x}$ implies that $\phi(\boldsymbol{y}) < j$, $j = 1, 2, \cdots, M$.
- $\boldsymbol{x}$ is an upper boundary point to system state level j iff $\phi(\boldsymbol{x}) \leq j$ and $\boldsymbol{y} > \boldsymbol{x}$ implies that $\phi(\boldsymbol{y}) > j$, $j = 0, 1, \cdots, M-1$.

The corresponding concepts in binary systems are minimum path sets and minimum cut sets.

The definitions of binary parallel, series, and k-out-of-n:G systems have been extended to the MS cases by allowing both the system and its components to have more than 2 possible states [2], [4]:

- The state of a MS series system is the state of the worst component.
- The state of a MS parallel system is the state of the best component.

The state of a MS k-out-of-n:G system is defined to be the state of the kth best component [4]. Reference [3] defines a MS k-out-of-n:G system to be a system with both:

Manuscript received December 15, 1998; revised June 24, 1999. This work was partially supported by the Natural Sciences and Engineering Research Council of Canada.

J. Huang and M. J. Zuo are with the Department of Mechanical Engineering, University of Alberta, 4-9 Mechanical Engineering Building, Edmonton, Alberta T6G 2G8 Canada (e-mail: hjinshen@gpu.srv.ualberta.ca; ming.zuo@ualberta.ca).

Y. Wu is with the Department of Mathematics, University of Toledo, Toledo, OH 43606-3390 USA (e-mail: ywu4@math.utoledo.edu).

Responsible editor: A. von Mayrhauser.

Publisher Item Identifier S 0018-9529(00)06209-6.

- $\binom{n}{k}$ lower boundary points to system state j $(j = 1, 2, \cdots, M)$,
- $\binom{n}{k-1}$ upper boundary points to system state j $(j = 0, 1, 2, \cdots, M-1)$.

These definitions of the k-out-of-n:G system, the parallel system, and the series system in the MS context are mutually consistent. The series and the parallel systems are still special cases of the k-out-of-n:G system. Under these definitions of the k-out-of-n:G system, maintaining at least a certain system state level requires the same number of components to be at or above a certain state: at least k components must be at state j or above for the system to be at state j or above $(j = 1, 2, \cdots, M)$ where k is s-independent of the value j of the system-state level.

This paper defines the MS k-out-of-n:G system wherein maintaining at least a certain system-state level might require a different number of components to be at a certain state or above. The required number of components depends on the system-state level under consideration. Examples are given to illustrate the applications of this definition. Reliability evaluation algorithms are developed for these MS k-out-of-n:G systems.

II. ASSUMPTIONS

1) The system is MS monotone [5]:
    - $\phi(\boldsymbol{x})$ is nondecreasing in each argument.
    - $\phi(\boldsymbol{j}) = \phi(j, j, \cdots, j) = j$ for $j \in \boldsymbol{S}$.
2 The x_i are mutually s-independent.
3 The possible states of each component and of the system are ordered: State $0 \le$ State $1 \le \cdots \le$ State M.

III. DEFINITION OF THE MS k-out-of-n:G SYSTEM

Definition: $\phi(\boldsymbol{x}) \ge j$ $(j = 1, 2, \cdots, M)$ if there exists an integer value l $(j \le l \le M))$ such that at least k_l components are in states at least as good as l. An n-component system with such a property is a MS k-out-of-n:G system.

In this definition, the k_j do not have to be the same for different system states j $(1 \le j \le M)$. This means that the structure of the MS system can be different for different system-state levels. A few examples are given to illustrate this point.

Notation:

N_j number of components in state j or above

$X_{i,j}$ 1: when component i is in state j or above; 0 otherwise

$$N_j = \sum_{i=1}^{n} X_{i,j}.$$

The definition is then rephrased as: $\phi(\boldsymbol{x}) \ge j$ if at least one of the following inequalities is satisfied:

$$\begin{aligned} N_j &\ge k_j, \\ N_{j+1} &\ge k_{j+1}, \\ N_{j+2} &\ge k_{j+2}, \\ &\vdots \\ N_M &\ge k_M. \end{aligned}$$

TABLE I
RELATION BETWEEN SYSTEM AND COMPONENT STATES

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

Two special cases of this definition are considered in this paper:

Increasing MS k-out-of-n:G System: $k_1 \le k_2 \le \cdots \le k_M$. For the system to be at a higher state level j, more components must be at state j or above; *ie*, there is an increasing requirement on the number of components that must be at a certain state or above for the system to be at a higher state level.

Decreasing MS k-out-of-n:G System: $k_1 \ge k_2 \ge \cdots \ge k_M$. For the system to be at a higher state level j, fewer components must be at state j or above; *ie*, there is a decreasing requirement on the number of components that must be at certain state or above for the system to be at a higher state level.

◀

When k_j is a constant, $k_1 = k_2 = \cdots = k_M = k$, the structure of the system is the same for all system state levels. This reduces to the definitions of the MS k-out-of-n:G system in [3], [4]. We call such systems constant MS k-out-of-n:G systems. All the concepts & results of binary k-out-of-n:G systems can be easily extended to the constant MS k-out-of-n:G systems. This paper treats this constant MS k-out-of-n:G system as a special case of the increasing MS k-out-of-n:G system.

A. Example 1: An Increasing MS k-out-of-n:G System

Given: A 3-component system with $k_1 = 1$, $k_2 = 2$, $k_3 = 3$. Both the system and the components can be in 1 of 4 possible states: 0, 1, 2, 3.

Table I illustrates the relationship between system state and component states.

Because $k_1 < k_2 < k_3$ in this example, there is a simpler version of the definition of MS k-out-of-n:G system.

- The system state is #3 if all 3 components are in state 3.
- The system state is #2 or above if at least 2 components are in state 2 or above.
- The system state is #1 or above if at least 1 component is in state 1 or above.

The system in this example has a

- 3-out-of-3:G series structure at system state #3,
- 2-out-of-3:G structure at system state #2,
- 1-out-of-3:G parallel structure at system state #1.

B. Example 2: Decreasing MS k-out-of-n:G System

Given: A 3-component system wherein both the system and the components can be in 1 of 4 possible states, 0, 1, 2, 3. This

TABLE II
RELATION BETWEEN SYSTEM AND COMPONENT STATES

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

TABLE III
RELATION BETWEEN SYSTEM AND COMPONENT STATES

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

decreasing k-out-of-n:G system satisfies the relationship between system state and component states as shown in Table II $k_1 = 3$, $k_2 = 2$, $k_3 = 1$

This system is in

- state #3 if at least 1 component is in state 3 $(k_3 = 1)$;
- state #2 or above if at least 2 components are in state 2 or above $(k_2 = 2)$, or if at least 1 component is in state 3 $(k_3 = 1)$.
- state 1 or above if all 3 components are in state 1 or above $(k_1 = 3)$, or if at least 2 components are in state 2 or above $(k_2 = 2)$, or if at least 1 component is in state 3 or above $(k_3 = 1)$.

In this example, $k_1 > k_2 > k_3$, indicating a strictly decreasing MS k-out-of-n:G system. This system has a

- 1-out-of-3:G structure at system state #3,
- 2-out-of-3:G structure at system state #2,
- 3-out-of-3:G structure at system state #1.

C. Example 3: Constant MS k-out-of-n:G System

Given: A 3-component system wherein both the system and the components can be in 1 of 4 possible states: 0, 1, 2, 3. A constant MS 2-out-of-3:G system satisfies the relationships in Table III.

Because $k_1 = k_2 = k_3 = 2$, there is a simpler version of the definition for this MS k-out-of-3:G system. The system is in

- state #3 if at least 2 components are in state 3.
- state #2 or above if at least 2 components are in state 2 or above.
- state #1 or above if at least 2 components are in state 1 or above.

The system has a 2-out-of-3:G structure at system states #1, #2, #3.

D. Example 4: A Production Management Problem

Given: A plant has 5 production lines for a certain product. The plant has 4 different production levels:

- full scale for maximum customer demand (state 3),
- average scale for usual customer demand (state 2),
- low scale when the customer demand is low (state 1),
- zero scale when the plant is shut down.

Each production line also has 4 different production levels:

- full scale,
- average scale,
- low scale,
- zero scale.

Then:

- All 5 lines have to work full scale, for the system to be in state 3.
- At least 3 lines have to work at least at the average scale, for the system to be in state 2 or above.
- At least 2 lines have to work at least at the low scale, for the system to be in state 1 or above.

Such a system can be represented by an increasing MS k-out-of-n:G system model with $k_1 = 2$, $k_2 = 3$, $k_3 = 5$.

E. Example 5: A Mining Operation

Given: A shovel-truck system in an open-mine.

Such a system usually consists of a shovel and a fleet of trucks (say 20 trucks). The deteriorating rates of the trucks are often higher than that of the shovel. Each truck and the system have 5 possible states. The system is in

- state #4 if at least 14 trucks are in state 4.
- at least state #3 if at least 15 trucks are in at least state 3, or at least 14 trucks are in state 4.
- at least state #2 if at least 16 trucks are in at least state 2, or at least 15 trucks are in at least state 3, or at least 14 trucks are in state 4.
- at least state #1 if at least 18 trucks are in at least state 1, or at least 16 trucks are in at least state 2, or at least 15 trucks are in at least state 3, or at least 14 trucks are in state 4.

This system can be represented by a decreasing MS k-out-of-5:G system with $k_1 = 18$, $k_2 = 16$, $k_3 = 15$, $k_4 = 14$.

IV. SYSTEM RELIABILITY EVALUATION

A common approach for MS system reliability evaluation is to extend the results from binary system reliability evaluation. Define

- the MS system as *functioning* if $\phi(\boldsymbol{x}) \geq j$, and *failed* otherwise.
- similarly, component i as *functioning* when $x_i \geq j$, and *failed* otherwise.

Because j can have various values, *functioning* and *failure* have different meanings for different j values; *ie*, the meanings of *functioning* and *failure* are dynamic (context dependent). When $\Pr\{\phi(\boldsymbol{x}) = j\}$ or $\Pr\{\phi(\boldsymbol{x}) \geq j\}$ are calculated for all j values, they are the probability distribution of the system in various states.

A. Case I: Increasing or Constant MS k-out-of-n:G Systems

$k_1 \leq k_2 \leq \cdots \leq k_M$

The definition of a MS k-out-of-n:G is equivalent to: $\phi(\boldsymbol{x}) \geq j$ iff at least k_j components have $x_i \geq j$.

If at least k_j components are in at least state j (these components are *functioning* as far as state level j is concerned), then the system is in state j or above (the system is considered to be *functioning*). The algorithms for binary k-out-of-n:G system reliability [1], [6] can be extended in this case for MS k-out-of-n:G system reliability:

$$R_{s,\,j} = R_j(n,\,k_j), \tag{1}$$

$$R_j(n,\,k_j) = P_{n,\,j} \cdot R_j(n-1,\,k_j-1) + Q_{n,\,j} \cdot R_j(n-1,\,k_j), \tag{2}$$

The boundary conditions for (2) are:

$$R_j(a,\,b) = 0, \qquad \text{for } b > a > 0, \tag{3}$$

$$R_j(a,\,0) = 1, \qquad \text{for } a \geq 0. \tag{4}$$

The probability that the system is in state j is:

$$r_{s,\,j} = R_{s,\,j} - R_{s,\,j+1}, \qquad R_{s,\,0} = 1. \tag{5}$$

When all the components have the same state probability distribution ($p_{i,\,j} = p_j$ for all i), then

$$R_{s,\,j} = \sum_{k=k_j}^{n} \binom{n}{k} \cdot P_j^k \cdot Q_j^{n-k}. \tag{6}$$

B. Example 6

Given: In example 1, let $p_0 = 0.1$, $p_1 = 0.3$, $p_2 = 0.4$, $p_3 = 0.2$. Use (6) to calculate the system probabilities at all levels.

$$P_1 = \sum_{j=1}^{3} p_j = 0.9, \quad P_2 = \sum_{j=2}^{3} p_j = 0.6, \quad P_3 = p_3 = 0.2;$$

$Q_1 = 0.1$, $Q_2 = 0.4$, $Q_3 = 0.8$.

At level 3, $k_3 = 3$, $R_{s,\,3} = P_3^3 = 0.2^3 = 0.008$;
At level 2, $k_2 = 2$, $R_{s,\,2} = \binom{3}{2} \cdot 0.6^2 \cdot 0.4 + 0.6^3 = 0.648$;
At level 1, $k_1 = 1$, $R_{s,\,1} = \binom{3}{1} \cdot 0.9 \cdot 0.1^2 + \binom{3}{2} \cdot 0.9^2 \cdot 0.1 + 0.9^3 = 0.999$.

The system probabilities at all levels are:

$$\begin{aligned} r_{s,\,3} &= R_{s,\,3} = 0.008; \\ r_{s,\,2} &= R_{s,\,2} - R_{s,\,3} = 0.64; \\ r_{s,\,1} &= R_{s,\,1} - R_{s,\,2} = 0.351; \\ r_{s,\,0} &= R_{s,\,0} - R_{s,\,1} = 1 - 0.999 = 0.001. \end{aligned}$$

C. Case II: Decreasing MS k-out-of-n:G System

$k_1 \geq k_2 \geq \cdots \geq k_M$

At least 1 inequality is a strict inequality.

When k_i is the same for all i, use the formulas for Case I to calculate system probability distribution. Otherwise, the definition of the MS k-out-of-n:G system is equivalent to:

$\phi(\boldsymbol{x}) = j$ iff at least k_j components are at or above state j, and at most $k_l - 1$ components are at state l or above for $l = j+1,\, j+2,\, \cdots,\, M;\, j = 1,\, 2,\, \cdots,\, M$.

The case with i.i.d. components and the case with non i.i.d.\ components are treated separately. When all the components have the same state probability distribution (they are i.i.d.), (7) is used to calculate the probability that the system is in state j:

$$r_{s,\,j} = \sum_{k=k_j}^{n} \binom{n}{k} \cdot Q_j^{n-k} \cdot \left(p_j^k + \sum_{l=j+1,\,k_l>1}^{M} \beta_l(k) \right), \tag{7}$$

$\beta_l(k)$ is the probability that:

- at least 1 and at most $k_l - 1$ components are in state l,
- at most $k_u - 1$ components are in state u for $j+1 \leq u < l$,
- the total number of components that are at states between j and l inclusive is k,
- the system is in state j.

As shown in (7), for a given k, the $n - k$ components are in states below j and the remaining k components must be in state j or above. At the same time, all these component states must make sure that the system is exactly in state j. Equation (7) sums the probabilities that 'exactly k components are in state j or above, without bringing the system state above j for $k = k_j,\, k_j + 1,\, \cdots,\, n$.

The expression in the last parentheses in (7) is the probability that at least k components are in state j or above without causing the system to be in a state above j, and "p_j^k is the probability that all the k components are exactly in state j" and "$\beta_{j+1}(k)$ is the probability that at least 1 and at most $k_{j+1} - 1$ components are in state $j+1$ and the total number of components in states j and $j+1$ is k."

When $l = j+2$, $\beta_l(k) = \beta_{j+2}(k)$ is the probability that

- i_1 components are in state $j+2$ for $1 \leq i_1 \leq k_{j+2} - 1$,
- i_2 components are in state $j+1$ for $0 \leq i_2 \leq k_{j+1} - 1 - i_1$,
- the total number of components in states j, $j+1$, and $j+2$ is k.

If $k_u = 1$ for any u, then

- $j + 1 \leq u \leq M$, and $\beta_l(k) = 0$ for $l > u$ because k_l is nonincreasing.

Let

$$I_a \equiv \sum_{m=1}^{a} i_m \qquad \text{for } a = 1,\, 2,\, \cdots,\, l-j.$$

Use (8) to calculate

$$\begin{aligned} \beta_l(k) = {} & \sum_{i_1=1}^{k_l-1} \binom{k}{i_1} \cdot p_l^{i_1} \cdot \sum_{i_2=0}^{k_{l-1}-1-I_1} \binom{k-I_1}{i_2} \cdot p_{l-1}^{i_2} \\ & \cdot \sum_{i_3=0}^{k_{l-2}-1-I_2} \binom{k-I_2}{i_3} \cdot p_{l-2}^{i_3} \cdots \sum_{i_{l-j}=0}^{k_{j+1}-1-I_{l-j-1}} \\ & \cdot \binom{k-I_{l-j-1}}{i_{l-j}} p_{j+1}^{i_{l-j}} \cdot p_j^{k-I_{l-j}}. \end{aligned} \tag{8}$$

The number of terms to be summed in (8) is equal to the number of ways to assign k identical balls to $l-j$ different cells with "at least 1 and at most k_l-1" balls in cell l, at most k_t-1 balls in cells t for $j<t<l$, and the remaining balls in cell j. Thus, the computation time for $\beta_l(k)$ in (8) is much smaller than K^{l-j} where $K=\max\{k_j,\, j=1,\,2,\,\cdots,\,n\}$. In turn, the computation time for $r_{s,\,j}$ in (7) is much less than $n\cdot K^M$.

We have developed a computer program for given values of M for calculating system-state distributions. For most practical engineering problems, a limited state number M, for example $M=10$, is big enough to describe the performances of the system and its components. Thus, (8) is practical.

D. Example 7: A 4-Component MS System

Given: The system and its components can be in state 0, 1, 2, 3, 4.

$k_1=4,\, k_2=3,\, k_3=2,\, k_4=1.$

The components are i.i.d. with $p_0=0.1, p_1=0.2, p_2=0.3, p_3=0.3, p_4=0.1$.

Use (7) and (8) to calculate the system probabilities at all levels.

$Q_4=0.9,\, Q_3=0.6,\, Q_2=0.3,\, Q_1=0.1.$

At level 4: $j=4,\, k_4=1$. Using (7),

$$\begin{aligned}r_{s,\,4}&=\sum_{k=1}^{4}\binom{4}{k}\cdot Q_4^{4-k}\cdot p_4^k\\&=\sum_{k=1}^{4}\binom{4}{k}\cdot(0.9)^{4-k}\cdot 0.1^k\\&=0.3439\end{aligned}$$

At level 3: $j=3,\, k_3=2$. Using (7),

$$\begin{aligned}r_{s,\,3}&=\sum_{k=2}^{4}\binom{4}{k}\cdot Q_3^{4-k}\cdot p_3^k\\&=\sum_{k=2}^{4}\binom{4}{k}\cdot(0.6)^{4-k}\cdot 0.3^k\\&=0.2673\end{aligned}$$

At level 2: $j=2,\, k_2=3$. Because $k_3>1$, find the expression for $\beta_3(k)$. Using (8),

$$\begin{aligned}\beta_3(k)&=\sum_{i_1=1}^{1}\binom{k}{i_1}\cdot p_3^{i_1}\cdot p_2^{k-i_1}\\&=\binom{k}{1}\cdot 0.3\cdot 0.3^{k-1}\\&=k\cdot 0.3^k.\\r_{s,\,2}&=\sum_{k=3}^{4}\binom{4}{k}\cdot Q_2^{4-k}\cdot\left[p_2^k+\beta_3(k)\right]\\&=\sum_{k=3}^{4}\binom{4}{k}\cdot 0.3^{4-k}\cdot\left(0.3^k+k\cdot 0.3^k\right)\\&=0.1701.\end{aligned}$$

At level 1: $j=1,\, k_1=4$. Because $k_2=3>1$ and $k_3=2>1$, find the expression for $\beta_2(k)$ and $\beta_3(k)$. Using (8),

$$\begin{aligned}\beta_2(k)&=\sum_{i_1=1}^{2}\binom{4}{i_1}\cdot p_2^{i_1}\cdot p_1^{k-i_1},\\\beta_3(k)&=\sum_{i_1=1}^{1}\binom{4}{i_1}\cdot p_3^{i_1}\cdot\left[\sum_{i_2=0}^{1}\binom{3}{i_2}\cdot p_2^{i_2}\cdot p_1^{3-i_2}\right].\end{aligned}$$

Using (7),

$$\begin{aligned}r_{s,\,1}&=\sum_{k=4}^{4}\binom{4}{4}\cdot Q_1^{4-4}\cdot\left[p_1^k+\sum_{l=2}^{3}\beta_l(k)\right]\\&=0.2^4+\beta_2(4)+\beta_3(4)=0.0856.\end{aligned}$$

At level 0: $r_{s,\,0}=1-r_{s,\,1}-r_{s,\,2}-r_{s,\,3}-r_{s,\,4}=0.1331$

When the components are not i.i.d., (9) is used to calculate the system probability at level j.

$$r_{s,\,j}=\sum_{k=k_j}^{n}\left[R_e^j(k,\,n)+\sum_{l=j+1,\,l>1}^{M}\beta_k^j(l)\right].\tag{9}$$

$\beta_k^j(l)$ is the probability that

- at least 1 and at most k_l-1 components are at state l $(l>j)$,
- at most k_u-1 components are at state u for $j<u<l$,
- the total number of components at state j or above is k,
- $n-k$ components are at states below j,
- the system is in state j.

Part of (9), $\sum_{k=k_j}^{n} R_e^j(k,\,n)$, is the probability that at least k_j components are in state j and the other components are below state j.

$$R_e^j(k,\,n)=p_{n,\,j}\cdot R_e^j(k-1,\,n-1)+Q_{n,\,j}\cdot R_e^j(k,\,n-1),\tag{10}$$

$$R_e^j(k,\,k)=\prod_{i=1}^{k}p_{i,\,j}\tag{11}$$

$$R_e^j(1,\,k)=\sum_{i=1}^{k}\left[\prod_{g=1}^{i-1}Q_{g,\,j}\right]\cdot p_{i,\,j}\cdot\left[\prod_{g=i+1}^{k}Q_{g,\,j}\right]\tag{12}$$

$$\begin{aligned}\beta_k^j(l)=&\sum_{i_1=1}^{k_l-1}\ \sum_{i_2=0}^{k_{l-1}-1-I_1}\ \sum_{i_3=0}^{k_{l-2}-1-I_2}\cdots\sum_{i_{l-j}=0}^{k_{j+1}-1-I_{l-j-1}}\\&\cdot R\left[\left(l^{i_1},\,(l-1)^{i_2},\cdots,\,(j+1)^{i_{l-j}},\,j^{k-I_{l-j}}\right),n\right];\end{aligned}\tag{13}$$

$$I_a\equiv\sum_{m=1}^{a}i_m,\qquad\text{for } a=1,\,2,\,\cdots,\,l-j.$$

In (13),

$$R\left[\left(l^{i_1},\,(l-1)^{i_2},\,\cdots,\,(j+1)^{i_{l-j}},\,j^{k-I_{l-j}}\right),\,n\right]$$

is the probability that there are exactly

- i_1 components at level l,
- i_2 components at level $l-1$,

- $\vdots$
- i_{l-j} components at level $j+1$,
- $k - I_{l-j}$ components at level j,
- the remaining $n-k$ components at states below j.

For simplicity, denote

$$\begin{aligned}&R[(l \sim j),\, n]\\ &\quad \equiv R\left[\left(l^{i_1},\, (l-1)^{i_2},\, \cdots,\, (j+1)^{i_{l-j}},\, j^{k-I_{l-j}}\right),\, n\right].\end{aligned}$$

Calculate $R[(l \sim j),\, n]$ using the recursive relation:

$$\begin{aligned}&R[(l \sim j),\, n]\\ &\quad = p_{n,\,l} \cdot R\left[\left(l^{i_1-1},\, *\right),\, n-1\right]\\ &\qquad + p_{n,\,l-1} \cdot R\left[\left((l-1)^{i_2-1},\, *\right),\, n-1\right] + \cdots\\ &\qquad + p_{n,\,j+1} \cdot R\left[\left((j+1)^{i_{l-j}-1},\, *\right),\, n-1\right]\\ &\qquad + p_{n,\,j} \cdot R\left[\left(j^{k-I_{l-j}-1},\, *\right),\, n-1\right]\\ &\qquad + Q_{n,\,j} \cdot R[(l \sim j),\, n-1];\\ &R\left[\left(l^{i_1-1},\, *\right),\, n-1\right] \qquad (14)\\ &\quad \equiv R\left[\left(l^{i_1-1},\, (l-1)^{i_2},\cdots,\, (j+1)^{i_{l-j}},\, j^{k-I_{l-j}}\right),\, n-1\right],\end{aligned}$$

and a similar convention is followed for other notation in (14). The boundary conditions for (14) are:

$$\begin{aligned}&R[(l^0,\, (l-1)^0,\, \cdots,\, i^1,\, \cdots,\, (j+1)^0,\, j^0),\, 1]\\ &\quad = p_{1,\,i}, \qquad (15)\\ &R[(l^0,\, (l-1)^0,\, \cdots,\, (j+1)^0,\, j^0),\, 1]\\ &\quad = Q_{1,\,j}, \qquad (16)\\ &R[(l^0,\, (l-1)^0,\, \cdots,\, (j+1)^0,\, j^0),\, n]\\ &\quad = \prod_{k=1}^{n} Q_{k,\,j}, \qquad (17)\\ &R[(l^0,\, (l-1)^0,\, \cdots,\, i^n,\, \cdots,\, (j+1)^0,\, j^0),\, n]\\ &\quad = \prod_{k=1}^{n} p_{k,\,i}, \qquad (18)\\ &R[(l^0,\, (l-1)^0,\, \cdots,\, i^1,\, \cdots,\, (j+1)^0,\, j^0),\, n]\\ &\quad = \sum_{k=1}^{n} \left[\prod_{g=1}^{k-1} Q_{g,\,j}\right] \cdot p_{k,\,i} \cdot \left[\prod_{g=k+1}^{n} Q_{g,\,j}\right]. \qquad (19)\end{aligned}$$

The computation time for $r_{s,\,j}$ in the noni.i.d. case with (9) is much less than $n^2 \cdot K^M$, $K \equiv \max\{k_i,\, i = 1,\, 2,\, \cdots,\, n\}$.

E. Example 8: 3-Component MS k-out-of-n System

Given: The component probabilities are in Table IV.

Let $k_1 = 3,\ k_2 = 2,\ k_3 = 2$. Use (9), (13), and (14) to calculate the system probabilities at all levels.

From the given data, $n = 3$,

$$\begin{aligned}Q_{1,\,1} &= 0.1, & Q_{1,\,2} &= 0.3, & Q_{1,\,3} &= 0.6,\\ Q_{2,\,1} &= 0.1, & Q_{2,\,2} &= 0.2, & Q_{2,\,3} &= 0.4,\\ Q_{3,\,1} &= 0.1, & Q_{3,\,2} &= 0.3, & Q_{3,\,3} &= 0.7.\end{aligned}$$

At level 3: $j = 3,\ k_3 = 2$; $r_{s,\,3} = \sum_{k=k_3}^{n} R_e^3(k,\, n) = R_e^3(2,\, 3) + R_e^3(3,\, 3)$. For $k = 2$: $R_e^3(2,\, 3) = p_{3,\,3} \cdot R_e^3(1,\, 2) + Q_{3,\,3} \cdot R_e^3(2,\, 2) = p_{3,\,3} \cdot (p_{1,\,3} \cdot Q_{2,\,3} + Q_{1,\,3} \cdot p_{2,\,3}) + Q_{3,\,3} \cdot p_{1,\,3} \cdot p_{2,\,3} = 0.324$. For $k = 3$: $R_e^3(3,\, 3) = p_{1,\,3} \cdot p_{2,\,3} \cdot p_{3,\,3} = 0.072$. $r_{s,\,3} = 0.324 + 0.072 = 0.396$

TABLE IV
COMPONENT PROBABILITIES

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

At level 2: $j = 2,\ k_2 = 2$,

$$\begin{aligned}r_{s,\,2} &= \sum_{k=k_2}^{n} \left(R_e^2(k,\, n) + \sum_{l=3}^{3} \beta_k^2(l)\right)\\ &= \sum_{k=2}^{3} (R_e^2(k,\, 3) + \beta_k^2(3)).\end{aligned}$$

For $k = 2$:

$$\begin{aligned}R_e^2(2,\, 3) &= p_{3,\,2} \cdot R_e^2(1,\, 2) + Q_{3,\,2} \cdot R_e^2(2,\, 2)\\ &= p_{3,\,2} \cdot (p_{1,\,2} \cdot Q_{2,\,2} + Q_{1,\,2} \cdot p_{2,\,2})\\ &\quad + Q_{3,\,2} \cdot p_{1,\,2} \cdot p_{2,\,2} = 0.066,\\ \beta_2^2(3) &= \sum_{i_1=1}^{1} R\left[\left(3^{i_1},\, 2^{k-i_1}\right),\, 3\right]\\ &= R[(3^1,\, 2^1),\, 3]\\ &= p_{3,\,3} \cdot R[(3^0,\, 2^1),\, 2] + p_{3,\,2} \cdot R[(3^1,\, 2^0),\, 2]\\ &\quad + Q_{3,\,2} \cdot R[(3^1,\, 2^1),\, 2]\\ &= p_{3,\,3} \cdot (p_{1,\,2} \cdot Q_{22} + Q_{1,\,2} \cdot p_{22})\\ &\quad + p_{3,\,2} \cdot (p_{1,\,3} \cdot Q_{2,\,2} + Q_{1,\,2} \cdot p_{2,\,3})\\ &\quad + Q_{3,\,2} \cdot (p_{1,\,2} \cdot p_{2,\,3} + p_{1,\,3} \cdot p_{2,\,2}) = 0.218.\end{aligned}$$

For $k = 3$:

$$\begin{aligned}R_e^2(3,\, 3) &= p_{1,\,2} \cdot p_{2,\,2} \cdot p_{3,\,2} = 0.024.\\ \beta_3^2(3) &= \sum_{i_1=1}^{1} R\left[\left(3^{i_1},\, 2^{k-i_1}\right),\, 3\right]\\ &= R[(3^1,\, 2^2),\, 3]\\ &= p_{3,\,3} \cdot R[(3^0,\, 2^2),\, 2] + p_{3,\,2} \cdot R[(3^1,\, 2^1),\, 2] + 0\\ &= p_{3,\,3} \cdot p_{2,\,2} \cdot p_{1,\,2} + p_{3,\,2} \cdot (p_{1,\,3} \cdot p_{2,\,2} + p_{1,\,2} \cdot p_{2,\,3})\\ &= 0.122\\ r_{s,\,2} &= 0.066 + 0.218 + 0.024 + 0.122 = 0.43\end{aligned}$$

For level 1: $j = 1,\ k_1 = 3$,

$$\begin{aligned}r_{s,\,1} &= \sum_{k=k_1}^{n} \left(R_e^1(k,\, n) + \sum_{l=2}^{3} \beta_k^j(l)\right)\\ &= R_e^1(3,\, 3) + \beta_3^1(2) + \beta_3^1(3).\\ R_e^1(3,\, 3) &= p_{1,\,1} \cdot p_{2,\,1} \cdot p_{3,\,1} = 0.004.\\ \beta_3^1(2) &= \sum_{i_1=1}^{1} R\left[\left(2^{i_1},\, 1^{k-i_1}\right),\, 3\right]\\ &= R[(2^1,\, 1^2),\, 3]\end{aligned}$$

$$\begin{aligned}
&= p_{3,2} \cdot R[(2^0, 1^2), 2] + p_{3,1} \cdot R[(2^1, 1^1), 2] + 0\\
&= p_{3,2} \cdot p_{2,1} \cdot p_{1,1} + p_{3,1} \cdot (p_{1,2} \cdot p_{2,1} + p_{1,1} \cdot p_{2,2})\\
&= 0.022.\\
\beta_3^1(3) &= \sum_{i_1=1}^{1} \sum_{i_2=0}^{0} R\left[\left(3^{i_1}, 2^{i_2}, 1^{k-i_1-i_2}\right), 3\right]\\
&= R[(3^1, 2^0, 1^2), 3]\\
&= p_{3,3} \cdot R[(3^0, 2^0, 1^2), 2] + 0\\
&\quad + p_{3,1} \cdot R[(3^1, 2^0, 1^1), 2] + 0\\
&= p_{3,3} \cdot p_{2,1} \cdot p_{1,1} + p_{3,1} \cdot (p_{2,3} \cdot p_{1,1} + p_{1,3} \cdot p_{2,1})\\
&= 0.038.\\
r_{s,1} &= 0.004 + 0.022 + 0.038 = 0.064
\end{aligned}$$

For level 0: $r_{s,0} = 1 - r_{s,1} - r_{s,2} - r_{s,3} = 1 - 0.064 - 0.43 - 0.396 = 0.11$.

Equation (9) was also used to calculate the system-state probability distribution for example 7. The same results were obtained.

V. DISCUSSION

As shown in the examples in Sections III and IV, it is nontrivial to find the reliability of a decreasing MS k-out-of-n system. Basically, bounds can be established on a complicated system's reliability. References [3], [4] illustrated how to use the upper critical connection vectors (or the lower boundary points) to establish bounds in a MS case. The corresponding concept of the "upper critical connection vectors to level j" is the "minimum path sets to level j" in this paper.

The results in [3], [4] can be applied to establishing bounds on MS decreasing k-out-of-n systems.

ACKNOWLEDGMENT

The authors would like to thank their editor and referees for detailed comments and suggestions.

REFERENCES

[1] R. E. Barlow and K. D. Heidtmann, "Computing k-out-of-n system reliability," *IEEE Trans. Reliability*, vol. R-33, pp. 322–323, Oct. 1984.

[2] R. E. Barlow and A. S. Wu, "Coherent systems with multi-state components," *Mathematics of Operations Research*, vol. 3, no. 4, pp. 275–281, 1978.

[3] R. A. Boedigheimer and K. C. Kapur, "Customer-driven reliability models for multi-state coherent systems," *IEEE Trans. Reliability*, vol. 43, pp. 46–50, Mar. 1994.

[4] E. El-Neweihi, F. Proschan, and J. Sethuraman, "Multi-state coherent system," *J. Applied Probability*, vol. 15, pp. 675–688, 1978.

[5] W. S. Griffith, "Multistate reliability models," *J. Applied Probability*, vol. 17, pp. 735–744, 1980.

[6] A. M. Rushdi, "Utilization of symmetric switching functions in the computation of k-out-of-n system reliability," *Microelectronics and Reliability*, vol. 26, no. 5, pp. 973–987, 1986.

Jinsheng Huang received his B.S. (1982) and M.S. (1988) in mechanical engineering from the University of Science and Technology, Beijing. He is pursing his Ph.D. in mechanical engineering at the University of Alberta. His research interests include reliability and mechanical design.

Ming J. Zuo (M) received the B.S. (1982) in agricultural engineering from Shandong Institute of Technology, M.S. (1986) and Ph.D. (1989) both in industrial engineering from Iowa State University, Ames. He worked at the University of Windsor in Canada in 1989. In 1990 he joined the Department of Mechanical Engineering at the University of Alberta where he is a Professor. From 1996–1998, he was on leave at the City University of Hong Kong. His research interests include system reliability, manufacturing systems, and applied operations research. He is on the editorial board of the IIE Transactions on Quality and Reliability Engineering. He is a Senior Member of IIE and a Member of IEEE.

Yanhong Wu is a Visiting Professor at the Department of Mathematics, University of Toledo. He has been teaching at the University of Alberta. He received his M.S. in operations research from the Institute of Applied Mathematics, Academia Sinica, P.R. China, and a Ph.D. in statistics from the University of Toronto. His research interests include change-point problem, sequential analysis, risk theory, quality control, and reliability.