Reliability Bounds for Multi-State k-out-of-n Systems

Zhigang Tian, Richard C. M. Yam, Ming J. Zuo, *Senior Member, IEEE*, and Hong-Zhong Huang

***Abstract*—Algorithms have been available for exact performance evaluation of multi-state k-out-of-n systems. However, especially for complex systems with a large number of components, and a large number of possible states, obtaining "reliability bounds" would be an interesting, significant issue. Reliability bounds will give us a range of the system reliability in a much shorter computation time, which allow us to make decisions more efficiently. The systems under consideration are multi-state k-out-of-n systems with i.i.d. components. We will focus on the probability of the system in states below a certain state d, denoted by Q_{sd}. Based on the recursive algorithm proposed by Zuo & Tian [14] for performance evaluation of multi-state k-out-of-n systems with i.i.d. components, a reliability bounding approach is developed in this paper. The upper, and lower bounds of Q_{sd} are calculated by reducing the length of the $\boldsymbol{k}$ vector when using the recursive algorithm. Using the bounding approach, we can obtain a good estimate of the exact Q_{sd} value while significantly reducing the computation time. This approach is attractive, especially to complex systems with a large number of components, and a large number of possible states. A numerical example is used to illustrate the significance of the proposed bounding approach.**

***Index Terms*—Bound, k-out-of-n systems, multi-state, recursive algorithm.**

NOTATION

x_i	state of component i, $x_i = j$ if component i is in state j, $0 \leq j \leq M$, $1 \leq i \leq n$
$\mathbf{x}$	an n-dimensional vector representing the states of all components, $\mathbf{x} = (x_1, x_2, \ldots, x_n)$
$\phi(\mathbf{x})$	state of the system, which is also called the structure function of the system, $0 \leq \phi(\mathbf{x}) \leq M$
p_j	probability that a component is in state j when all components are i.i.d.
$Q_{s,d}$	$\Pr(\phi(\mathbf{x}) < d)$
$R_{s,d}$	$1 - Q_{s,d}$
N	number of components of a nominal increasing multi-state k-out-of-n:F system
m	number of possible states of a nominal increasing multi-state k-out-of-n:F system minus 1
$\boldsymbol{k}$	the $\boldsymbol{k}$ vector of the nominal increasing multi-state k-out-of-n:F system where $\boldsymbol{k} = (k_1, k_2, \ldots, k_m)$
$\boldsymbol{p}$	the probability vector of the nominal increasing multi-state k-out-of-n:F system where $\boldsymbol{p} = (p_0, p_1, \ldots, p_m)$
$Q(\bullet)$	the recursive function used by the proposed recursive algorithm, $Q = Q(m, N, \boldsymbol{k}, \boldsymbol{p})$
$Q_{\boldsymbol{k}}(\bullet)$	the $Q_{s,d}$ value when vector $\boldsymbol{k}$ has a specific value
$\mathtt{Pr}_{\mathtt{ub}}$	the obtained upper bound
$\mathtt{Pr}_{\mathtt{lb}}$	the obtained lower bound
$t_{\mathtt{ub}}$	the computation time for calculating the upper bounds
$t_{\mathtt{lb}}$	the computation time for calculating the lower bounds

I. INTRODUCTION

PRACTICAL systems like power station coal transmission systems, and wireless communication networks [9] can be modeled as multi-state systems [1], [3], [8], [11]. These systems can perform their intended functions at full capacity, different levels of reduced capacity, and of course they can also be totally failed. The multi-state system models can represent equipment conditions with more accuracy, and flexibilities than the binary system models, where the components, and the system can only be in two possible states: working, or failed.

The k-out-of-n system model is a general system reliability model, and has been studied extensively in the binary context, where the system, and the components can only be in two states. A binary k-out-of-n:G system works only if at least k components work. Apparently, there is only one k value with respect to a binary k-out-of-n:G system, or k-out-of-n:F system. However, the only reason for this, we believe, is that there are only two states in binary k-out-of-n systems. In the case of multi-state systems, we should have more than one k values in the same system model. By allowing for different k values with respect to different states, the model of a generalized multi-state k-out-of-n:G system [6], and the model of a generalized multi-state k-out-of-n:F system [12], [14] have been developed. These are natural extensions of binary k-out-of-n systems. Exact performance evaluation algorithms have been available for multi-state k-out-of-n systems with i.i.d. components [12], [14], and multi-state k-out-of-n systems with independent components [13]. These algorithms are much more efficient than enumeration methods. However, for complex systems with a large number of components, and a large number of possible states, the calculation of system state distribution will still require a significant amount of time. In practical

Manuscript received August 7, 2005; revised August 30, 2006; accepted September 26, 2006. This work was supported by the City University of Hong Kong (SRG Project #7001433), the Natural Sciences and Engineering Research Council of Canada, the Alberta Ingenuity Fund, and the Killam Trusts. Associate Editor: L. Cui.

Z. Tian and M. J. Zuo are with the Department of Mechanical Engineering, University of Alberta, Edmonton, AB T6G 2G8, Canada.

R. C. M. Yam is with the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong (e-mail: mery@cityu.edu.hk).

H.-Z. Huang is with the School of Mechatronics Engineering, University of Electronic Science and Technology of China, China.

Digital Object Identifier 10.1109/TR.2007.909766

situations, sometimes we do not have to obtain the exact system state distribution. We would rather get a good enough range of the system reliability in a much shorter computation time, which will allow us to make decisions more efficiently. This is why "reliability bounds" is an interesting, significant issue.

The issue of reliability bounding has been extensively studied, both in the binary context [8], and multi-state context [4], [5], [7], [8], [10]. Several binary reliability bounding approaches are generalized to multi-state systems by Block & Savits [2], and analysed by Meng [10]. These approaches are some simple formulas generalized from the binary case. Hudson & Kapur [7] developed bounding approaches for multi-state systems using the Inclusion-Exclusion (IE) method, and Sum of Disjoint Product (SDP) method, assuming that the minimal cut vectors or minimal path vectors are given. Huang *et al.* developed bounding approaches for generalized multi-state k-out-of-n:G systems [4], and consecutive multi-state k-out-of-n systems [5], by simplifying the minimal path or cut vectors to include no more than two different states. The limitation of their approaches are apparent; that is, we can not include more than two different states to seek better bounds.

In general, a systematic, flexible approach is still not available to obtain reliability bounds for multi-state k-out-of-n systems with i.i.d. components. As mentioned, an efficient recursive algorithm has been available for the exact performance evaluation of multi-state k-out-of-n systems [12], [14]. In this paper, we will propose a systematic, flexible reliability bounding approach based on the recursive algorithm. Using the bounding approach, we can obtain a good estimate of the exact system reliability value while significantly reducing the computation time. This approach is attractive, especially to complex systems with a large number of components, and a large number of possible states. A numerical example will be used to illustrate the significance of the proposed bounding approach.

Assumptions:

- The state space of each component, and the system is $\{0, 1, 2, \ldots, M\}$.
- The states of all components are i.i.d. random variables.
- The state of the system is completely determined by the states of the components.
- A lower state level represents a worse or equal performance of the component, or the system.

II. Recursive Algorithm for Performance Evaluation of Multi-State k-out-of-n Systems

Huang *et al.* [6] proposed the generalized multi-state k-out-of-n:G system model, where there can be different k values with respect to different states. That is, the system is in state j or above if there exists an integer value l $(j \leq l \leq M)$ such that at least k_l components are in state l or above. The definition of the generalized multi-state k-out-of-n:G system can be stated in terms of the states of both the system, and the components being below a certain level, which leads to the definitions of a generalized multi-state k-out-of-n:F system, and its two special cases proposed in [12] & [14] as follows.

Definition 1: [12], [14] An n-component system is called a generalized multi-state k-out-of-n:F system if $\phi(\mathbf{x}) < j$ $(1 \leq j \leq M)$ whenever the states of at least k_l components are below l for all l such that $j \leq l \leq M$.

Definition 2: [12], [14] A generalized multi-state k-out-of-n:F system is called an increasing multi-state k-out-of-n:F system if $k_1 < k_2 < \cdots < k_M$.

Definition 3: [12], [14] A generalized multi-state k-out-of-n:F system is called a decreasing multi-state k-out-of-n:F system if $k_1 \geq k_2 \geq \cdots \geq k_M$.

Another important definition is the definition of nominal increasing multi-state k-out-of-n:F system proposed in [12], [14].

Definition 4: [12], [14] A nominal increasing multi-state k-out-of-n:F system is the same as an increasing multi-state k-out-of-n:F system except that the probability of a component in all possible states may be less than 1.

A recursive algorithm is proposed in [12], [14] for performance evaluation of multi-state k-out-of-n systems with i.i.d. components. Specifically, we need to calculate $Q_{s,j}$, the probability that the system in states below j transitions to any system state j. This recursive algorithm will be presented briefly in the following part of this section.

As summarized in the notation, we will use $Q(m, N, \boldsymbol{k}, \boldsymbol{p})$ to represent the recursive function used in the recursive algorithm, where m is the number of possible states of the nominal increasing multi-state k-out-of-n:F system minus 1, N is the number of components of the nominal increasing multi-state k-out-of-n:F system, $\boldsymbol{k}$ is the $\boldsymbol{k}$ vector of the nominal increasing multi-state k-out-of-n:F system where $\boldsymbol{k} = (k_1, k_2, \ldots, k_m)$, and $\boldsymbol{p}$ is the probability vector of the nominal increasing multi-state k-out-of-n:F system where $\boldsymbol{p} = (p_0, p_1, \ldots, p_m)$. The recursive function, $Q(m, N, \boldsymbol{k}, \boldsymbol{p})$, is designed to represent the probability that the nominal increasing multi-state k-out-of-n:F System is in the nominal state "0". Thus, to calculate the $Q_{s,j}$ value of any multi-state k-out-of-n:F system, we can first transform it into a nominal increasing multi-state k-out-of-n:F system to state j, and do the calculation using the recursive algorithm.

The procedure of the recursive algorithms is as follows:

$$Q(m, N, \boldsymbol{k}, \boldsymbol{p}) = \sum_{i=k_1}^{k_2-1} \binom{N}{i} p_0^i \cdot Q(m-1, N-i, \dot{\boldsymbol{k}}, \dot{\boldsymbol{p}}) + Q(m-1, N, \ddot{\boldsymbol{k}}, \ddot{\boldsymbol{p}}) \tag{1}$$

where $\dot{\boldsymbol{k}} = (k_2 - i, k_3 - i, \ldots, k_m - i)$, $\dot{\boldsymbol{p}} = (p_1, p_2, \ldots, p_m)$, $\ddot{\boldsymbol{k}} = (k_2, k_3, \ldots, k_m)$, and $\ddot{\boldsymbol{p}} = (p_0, p_1 + p_2, p_3, \ldots, p_m)$.The boundary condition for the recursive algorithm is

$$Q(1, N, \boldsymbol{k}, \boldsymbol{p}) = \sum_{i=k_1}^{N} \binom{N}{i} p_0^i \cdot p_1^{N-i} \tag{2}$$

From (1) and (2), we can see that this algorithm is actually recursive on the parameter m, not on the number of components N. Therefore, we can apply the recursive algorithm to large systems including a large number of components, without leading to exponential growth of computation time.

III. The Proposed Reliability Bounding Approach

The systems under consideration are multi-state k-out-of-n systems with i.i.d. components. We will focus on the probability that the system is in states below a certain state d; that is

$$Q_{s,d} = \Pr\left(\phi(\mathbf{x}) < d\right) \tag{3}$$

The probability that the system is in state d or above, represented by R_{sd}, is equal to $1-Q_{s,d}$. Based on the recursive algorithm for performance evaluation of multi-state k-out-of-n systems with i.i.d. components [12], [14], a reliability bounding approach is proposed in this section.

As mentioned in Section II, the calculation of $Q_{s,d}$ can be transformed into the probability for the system to be in state "0" of a generated nominal increasing multi-state k-out-of-n:F system with $m+1$ possible states, totally N components, vector $\boldsymbol{k}$, and probability vector $\boldsymbol{p}$. Furthermore, the algorithm in [12], [14] is actually recursive on the parameter m, not on the number of components N. The algorithm can be applied to large systems including a large number of components without leading to an exponential growth of computation time. However, when the number of possible nominal states m increases, the computation time will increase much faster than the case when N increases. Thus, it is desirable to develop a bounding approach that is computationally more efficient.

$Q_{s,d}$ is used to represent the exact probability value that the system is in states below nominal state d. We use $Q_{\boldsymbol{k}}\ (k_1,\ k_2,\ \ldots,\ k_m)$ to represent the $Q_{s,d}$ value when $\boldsymbol{k} = (k_1,\ k_2,\ \ldots,\ k_m)$. We have the following property:

Property: For any nominal state $j\ (1 \le j \le m)$ of a nominal increasing multi-state k-out-of-n:F system, we have

$$Q_{\boldsymbol{k}}(k_1,\ \ldots,\ k_j,\ \ldots,\ k_m) \le Q_{\boldsymbol{k}}(k_1,\ \ldots,\ k_j-1,\ \ldots,\ k_m) \tag{4}$$

The reason is whenever there are k_j components in states below j, there will be always k_j-1 components in states below j. This property provides us a basis to generate bounds for $Q_{s,d}$.

Consider a specific example first. Suppose the generated $\boldsymbol{k}$ vector is (1, 2, 3, 4), which includes 4 elements in strictly increasing order. Based on (4), if $\boldsymbol{k}$ is equal to (1, 1, 1, 1), we will have a bigger $Q_{s,d}$ value because the requirement on nominal states 2, 3, and 4 become less strict. Thus, we have an upper bound for $Q_{s,d}$. Based on a similar analysis, we can find that $Q_{\boldsymbol{k}}\ (4,\ 4,\ 4,\ 4)$ is smaller than $Q_{s,d}$. Thus, we have a pair of upper, and lower bounds for $Q_{s,d}$

$$Q_{\boldsymbol{k}}(4,\ 4,\ 4,\ 4) \le Q_{s,d} = Q_{\boldsymbol{k}}(1,\ 2,\ 3,\ 4) \le Q_{\boldsymbol{k}}(1,\ 1,\ 1,\ 1) \tag{5}$$

The $Q_{s,d}$ value of a general multi-state k-out-of-n:F system, where the $\boldsymbol{k}$ vector is not necessarily in a strictly increasing order, can be calculated by transforming the system into a nominal increasing multi-state k-out-of-n:F system [12], [14]. Therefore, $Q_{\boldsymbol{k}}\ (1,\ 1,\ 1,\ 1)$ can be simplified to include only one element in the $\boldsymbol{k}$ vector; that is, $Q_{\boldsymbol{k}}\ (1,\ 1,\ 1,\ 1) = Q_{\boldsymbol{k}}(1)$. Note that, in the case of $Q_{\boldsymbol{k}}(1)$, there are only two nominal states: state 0, and state 1. Specifically, state 1, 2, 3, 4 of the original system are combined into one nominal state 1.Thus, the upper, and lower bounds in (5) can be written as

$$Q_{\boldsymbol{k}}(4) \le Q_{s,d} = Q_{\boldsymbol{k}}(1,\ 2,\ 3,\ 4) \le Q_{\boldsymbol{k}}(1) \tag{6}$$

In the bounds in (6), we include only one element in the $\boldsymbol{k}$ vectors. We will have tighter bounds if we include more elements in them. Specifically, in the case of lower bounds, we have

$$\begin{aligned} Q_{\boldsymbol{k}}(4,\ 4,\ 4,\ 4) &\le Q_{\boldsymbol{k}}(1,\ 4,\ 4,\ 4)\\ &\le Q_{\boldsymbol{k}}(1,\ 2,\ 4,\ 4)\\ &\le Q_{\boldsymbol{k}}(1,\ 2,\ 3,\ 4)\\ &= Q_{s,d}. \end{aligned} \tag{7}$$

And in the case of upper bounds, we have

$$\begin{aligned} Q_{s,d} &= Q_{\boldsymbol{k}}(1,\ 2,\ 3,\ 4)\\ &\le Q_{\boldsymbol{k}}(1,\ 2,\ 3,\ 3)\\ &\le Q_{\boldsymbol{k}}(1,\ 2,\ 2,\ 2)\\ &\le Q_{\boldsymbol{k}}(1,\ 1,\ 1,\ 1). \end{aligned} \tag{8}$$

If we write them in a simplified way as in (6), we have

$$Q_{\boldsymbol{k}}(4) \le Q_{\boldsymbol{k}}(1,\ 4) \le Q_{\boldsymbol{k}}(1,\ 2,\ 4) \le Q_{\boldsymbol{k}}(1,\ 2,\ 3,\ 4) = Q_{s,d}, \tag{9}$$

and

$$Q_{s,d} = Q_{\boldsymbol{k}}(1,\ 2,\ 3,\ 4) \le Q_{\boldsymbol{k}}(1,\ 2,\ 3) \le Q_{\boldsymbol{k}}(1,\ 2) \le Q_{\boldsymbol{k}}(1). \tag{10}$$

All the $Q_{\boldsymbol{k}}(\bullet)$ values in (9) and (10) can be calculated using the efficient recursive algorithm presented in [12], [14].

Now we consider the general case $Q_{\boldsymbol{k}}(k_1,\ k_2,\ \ldots,\ k_m)$, where there are m strictly increasing elements in the $\boldsymbol{k}$ vector. Based on the property in (4), and the analysis on the specific example above, we have the series of lower bounds for $Q_{s,d}$ as follows (showing all the m elements).

$$\begin{aligned} &Q_{\boldsymbol{k}}(k_m,\ k_m,\ \ldots,\ k_m)\\ &\quad \le Q_{\boldsymbol{k}}(k_1,\ k_m,\ \ldots,\ k_m)\\ &\quad \le Q_{\boldsymbol{k}}(k_1,\ k_2,\ k_m,\ldots,\ k_m)\\ &\quad \le \cdots \le Q_{\boldsymbol{k}}(k_1,\ k_2,\ \ldots,\ k_{m-2},\ k_m,\ k_m)\\ &\quad \le Q_{\boldsymbol{k}}(k_1,\ k_2,\ \ldots,\ k_{m-1},\ k_m) = Q_{s,d}, \end{aligned} \tag{11}$$

or in the simplified form,

$$\begin{aligned} Q_{\boldsymbol{k}}(k_m) \le Q_{\boldsymbol{k}}(k_1,\ k_m) \le \ldots \le Q_{\boldsymbol{k}}(k_1,\ k_2,\ \ldots,\ k_{m-2},\ k_m)\\ \le Q_{\boldsymbol{k}}(k_1,\ k_2,\ \ldots,\ k_{m-1},\ k_m) = Q_{s,d} \end{aligned} \tag{12}$$

And the series of upper bounds for $Q_{s,d}$ are (showing all the m elements)

$$\begin{aligned} Q_{sd} &= Q_{\boldsymbol{k}}(k_1,\ k_2,\ \ldots,\ k_{m-1},\ k_m)\\ &\le Q_{\boldsymbol{k}}(k_1,\ k_2,\ \ldots,\ k_{m-1},\ k_{m-1})\\ &\le \ldots \le Q_{\boldsymbol{k}}(k_1,\ k_2,\ k_2,\ \ldots,\ k_2)\\ &\le Q_{\boldsymbol{k}}(k_1,\ k_1,\ \ldots,\ k_1), \end{aligned} \tag{13}$$

or in the simplified form

$$\begin{aligned} Q_{s,d} &= Q_{\boldsymbol{k}}(k_1, \ldots, k_m) \\ &\le Q_{\boldsymbol{k}}(k_1, \ldots, k_{m-1}) \\ &\le \ldots \le Q_{\boldsymbol{k}}(k_1, k_2) \\ &\le Q_{\boldsymbol{k}}(k_1) \end{aligned} \tag{14}$$

In the general case, all the $Q_{\boldsymbol{k}}(\bullet)$ values in (12), and (14) can calculated using the efficient recursive algorithm presented in [12], [14]. When calculating the bounds, the more elements we include in the $\boldsymbol{k}$ vector, the tighter the bounds will be, and the longer the computation time will be.

Aside from (11) & (12), there is actually another way to calculate the lower bounds:

$$\begin{aligned} Q_{\boldsymbol{k}}(k_m, k_m, \ldots, k_m) &\le Q_{\boldsymbol{k}}(k_{m-1}, \ldots, k_{m-1}, k_m) \\ &\le Q_{\boldsymbol{k}}(k_{m-2}, \ldots, k_{m-2}, k_{m-1}, k_m) \\ &\le \cdots \le Q_{\boldsymbol{k}}(k_2, k_2, \ldots, k_{m-1}, k_m) \\ &\le Q_{\boldsymbol{k}}(k_1, k_2, \ldots, k_{m-1}, k_m) \\ &= Q_{s,d}. \end{aligned} \tag{15}$$

However, the lower bounds obtained using (15) are not as good as those obtained using (11) & (12). This will be illustrated using an example later in Section IV.

Any multi-state k-out-of-n:G system has an equivalent multi-state k-out-of-n:F system [12]; therefore, we will focus only on multi-state k-out-of-n:F systems. Given a general multi-state k-out-of-n:F system, the procedure to calculate $Q_{s,d}$ with respect to a certain state d is as follows:

1) Generate the nominal increasing multi-state k-out-of-n:F system with the $\boldsymbol{k}$ vector including m elements in a strictly increasing order.
2) Based on how complex the system is, that is, how many components there are & how large m is, we decide how many elements we want to include in the $\boldsymbol{k}$ vector when calculating the reliability bounds. A simple, effective way is to start by including only one element, investigating the obtained bounds, and iteratively increasing the number of elements included as necessary. We can certainly include a different number of elements when calculating the upper versus the lower bounds. From our numerical experiments on the bounding approach, the upper bound is usually better than the lower bound when including the same number of elements in the $\boldsymbol{k}$ vector. Therefore, it is recommended to include more elements in the $\boldsymbol{k}$ vector when calculating the lower reliability bound.
3) Based on how good the obtained bounds are, and how efficient the calculation is, we have the flexibility to include a different number of elements in the $\boldsymbol{k}$ vector, and investigate more options.

IV. Examples

We will use a numerical example to investigate the accuracy, and efficiency of the proposed reliability bounding approach. As mentioned in Section III, $Q_{s,d}$, the probability of a multi-state k-out-of-n system in states below d, can be calculated through the probability of a generated nominal increasing multi-state k-out-of-n:F system in nominal state 0. The time required for generating the nominal increasing multi-state k-out-of-n:F system is negligible. In this example, we will use the generated nominal increasing multi-state k-out-of-n:F system directly to investigate the proposed bounding approach. Thus, d is specified to be 1, and $Q_{s,d}$ represents the probability of the system being in state 0.

TABLE I
Reliability Bounding Results

m	$\mathtt{Pr_{lb}}$	$\mathtt{Pr_{ub}}$	$t_{\mathtt{lb}}$	$t_{\mathtt{ub}}$
1	4.32E-12	0.81630461	0.03	0.03
2	6.86E-04	0.81595876	0.27	0.13
3	0.31592575	0.81595735	0.91	0.40
4	0.80768654	0.81595735	3.98	1.51
5	0.81595701	0.81595735	17.24	7.65
6	0.81595735	0.81595735	71.32	41.60
7	0.81595735	0.81595735	233.63	233.63

The nominal increasing multi-state k-out-of-n:F system used in this example has 100 i.i.d. components, and 8 possible states from state 0 to state 7. Thus, we have $n = 100$, and $M = 7$. The $\boldsymbol{k}$ vector is specified to be

$$\boldsymbol{k} = (10, 15, 20, 25, 30, 35, 40). \tag{16}$$

For convenience, we set the probabilities of a component in different states to be identical; that is, the state distribution vector is $\boldsymbol{p}$ is

$$\boldsymbol{p} = (0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125, 0.125). \tag{17}$$

Actually, the value of state distribution vector $\boldsymbol{p}$ will not influence the computation time of the bounding approach. The factors that influence the computation time of the bounding approach are the values of n, M, and vector $\boldsymbol{k}$.

To calculate the exact $Q_{s,d}$ value, we need to include the total 7 elements in (16) into the $\boldsymbol{k}$ vector. Using the proposed bounding approach in Section III, we can get a series of upper, and lower bounds by including a different number of elements into the $\boldsymbol{k}$ vector. The upper, and lower bounds are calculated using (14), and (12) respectively. The programs for this example are developed with MATLAB 6.5, and implemented on a computer with a Pentium M 1.7GHz CPU, and 512 RAM. The results by the bounding approach are shown in Table I, where m represents the number of elements included in the $\boldsymbol{k}$ vector when calculating the bounds; $\mathtt{Pr_{lb}}$, and $\mathtt{Pr_{ub}}$ represent the obtained lower bound, and upper bound; and $t_{\mathtt{lb}}$, and $t_{\mathtt{ub}}$ are the computation time for calculating the bounds, respectively. When $m = M = 7$, both $\mathtt{Pr_{ub}}$, and $\mathtt{Pr_{lb}}$ are equal to the exact $Q_{s,d}$ value, 0.81595735.

From Table I, we will have more accurate upper, and lower bounds with the increase of m. It can be found that the obtained upper bounds are close to the exact $Q_{s,d}$ value, even when $m = 1$. From $m = 4$, the upper bounds are the same in Table I as

TABLE II
THE $\boldsymbol{k}$ VECTORS FOR CALCULATING THE LOWER BOUNDS WITH TWO DIFFERENT METHODS

m	Method 1	Method 2
1	(40, 40, 40, 40, 40, 40, 40)	(40, 40, 40, 40, 40, 40, 40)
2	(10, 40, 40, 40, 40, 40, 40)	(35, 35, 35, 35, 35, 35, 40)
3	(10, 15, 40, 40, 40, 40, 40)	(30, 30, 30, 30, 30, 35, 40)
4	(10, 15, 20, 40, 40, 40, 40)	(25, 25, 25, 25, 30, 35, 40)
5	(10, 15, 20, 25, 40, 40, 40)	(20, 20, 20, 25, 30, 35, 40)
6	(10, 15, 20, 25, 30, 40, 40)	(15, 15, 20, 25, 30, 35, 40)
7	(10, 15, 20, 25, 30, 35, 40)	(10, 15, 20, 25, 30, 35, 40)

TABLE III
THE LOWER BOUNDS RESULTS WITH TWO DIFFERENT METHODS

m	$\mathtt{Pr_{lb}}$ (Method 1)	$\mathtt{Pr_{lb}}$ (Method 2)	$t_{\mathtt{lb}}$ (Method 1)	$t_{\mathtt{lb}}$ (Method 2)
1	4.32E-12	4.32E-12	0.03	0.03
2	6.86E-04	6.16E-09	0.27	0.09
3	0.31592575	3.03E-06	0.91	0.35
4	0.80768654	4.78E-04	3.98	1.23
5	0.81595701	0.02203005	17.24	6.61
6	0.81595735	0.26478972	71.32	38.56
7	0.81595735	0.81595735	233.63	233.63

the exact $Q_{s,d}$ value, up to 8 decimal places. On the other hand, the lower bounds are not so good when m is small, but they are close to the exact $Q_{s,d}$ value as well when m is 4 or larger. We also investigated other increasing multi-state k-out-of-n:F systems by varying the value of M, n, vector $\boldsymbol{k}$, and vector $\boldsymbol{p}$. It turns out that we can always get good upper bounds which are close to the exact $Q_{s,d}$ value, even when m is relatively small. The obtained lower bounds show similar accuracy performance as those of the system in Table I.

The computation times $t_{\mathtt{lb}}$, and $t_{\mathtt{ub}}$ in Table I increase greatly with the increase of m. Therefore, considering the accuracy of the bounds we mentioned in the previous paragraph, it would be a good idea to use appropriate upper, and lower bounds if we do not have to find the exact value. For example, we can use the upper bound when $m = 2$, which is 0.81595876; and the lower bound when $m = 4$, which is 0.80768654. This whole range will be only about 1 percent of the exact $Q_{s,d}$ value, 0.81595735; and the total computation time is only 4.11 seconds, about 1.8 percent of that for calculating the exact value. These bounds will give us a good idea of the actual $Q_{s,d}$ value in a much shorter time.

We also investigated the efficiency of the bounding approach when the number of the components n increases. For instance, when calculating the upper bound with $m = 4$, if we increase n from 100 to 200 while keeping other settings the same, the computation time will increase from 1.51 to 3.24. This confirms that the recursive algorithm in [12] & [14] used in the bounding approach is efficient versus the number of components n. Thus, the proposed bounding approach can be used as an efficient performance evaluation approach to multi-state k-out-of-n systems with a large number of components, and possible states.

As mentioned in Section III, there is another method for calculating the lower bounds, using (15). The performances of the two methods are compared. The $\boldsymbol{k}$ vectors (before being simplified) used in the two methods are listed in Table II, where "Method 1" refers to the method using (11) & (12), and "Method 2" refers to the method using (15). The results are shown in Table III. When the number of elements m included in the $\boldsymbol{k}$ vector is 1, the two methods give the same lower bound value; this value is very close to 0, and thus is not useful for system reliability approximation at all. As the m value increases, the lower bounds provided by method 1 grows much faster than those provided by method 2. This shows that method 1 provides much tighter lower bounds for system reliability evaluation. Of course, when $m = M = 7$, all the elements in the $\boldsymbol{k}$ vector are used, and both methods give the exact system reliability value.

V. CONCLUSIONS

A reliability bounding approach is proposed in this paper based on the recursive algorithm for performance evaluation of multi-state k-out-of-n systems with i.i.d. components. The upper, and lower bounds of $Q_{s,d}$ are calculated by reducing the length of the $\boldsymbol{k}$ vector when using the recursive algorithm presented in [12] & [14]. Usually, we can get better upper bounds than lower bounds when including the same number of elements in the $\boldsymbol{k}$ vectors. Using the bounding approach, we can obtain a good estimate of the exact $Q_{s,d}$ value while significantly reducing the computation time. Generally speaking, the proposed bounding approach can be used as an efficient performance evaluation approach to multi-state k-out-of-n systems with a large number of components, and possible states.

The contributions of the proposed reliability bounding approach are: 1) An approach to obtain reliability bounds for multi-state k-out-of-n systems is proposed. Specifically, the upper, and lower bounds of $Q_{s,d}$ are calculated by reducing the length of the $\boldsymbol{k}$ vector when using the recursive algorithm presented in [12], [14]. 2) The bounding approach provides a fast reliability evaluation way, attractive especially to complex systems with a large number of components, and a large number of possible states. 3) By controlling the length of the $\boldsymbol{k}$ vector used in the proposed bounding approach, we can obtain reliability bounds with different levels of accuracy.

ACKNOWLEDGMENT

The constructive comments from reviewers, and the editor are very much appreciated.

REFERENCES

[1] R. E. Barlow and K. D. Heidtmann, "Computing k-out-of-n system reliability," *IEEE Trans. Reliability*, vol. 33, no. 4, pp. 322–323, 1984.
[2] H. W. Block and T. H Savits, "A decomposition for multistate monotone systems," *Journal of Applied Probability*, vol. 19, no. 2, pp. 391–402, 1982.
[3] E. El-Neweihi, F. Proschan, and J. Sethuraman, "Multi-state coherent system," *Journal of Applied Probability*, vol. 15, no. 4, pp. 675–688, 1978.
[4] J. Huang and M. J. Zuo, "Dominant multi-state systems," *IEEE Trans. Reliability*, vol. 53, no. 3, pp. 362–368, 2004.

[5] J. Huang, M. J. Zuo, and Z. Fang, "Multi-state consecutive-k-out-of-n systems," *IIE Transactions*, vol. 35, no. 6, pp. 527–534, 2003.
[6] J. Huang, M. J. Zuo, and Y. H. Wu, "Generalized multi-state k-out-of-n:G systems," *IEEE Trans. Reliability*, vol. 49, no. 1, pp. 105–111, 2000.
[7] J. C. Hudson and K. C. Kapur, "Reliability bounds for multistate systems with multistate components," *Operations Research*, vol. 33, no. 1, pp. 153–160, 1985.
[8] W. Kuo and M. J. Zuo*, Optimal Reliability Modeling: Principles and Applications*. New York: John Wiley & Sons, 2003.
[9] A. Lisnianski and G. Levitin*, Multi-state System Reliability: Assessment, Optimization and Applications*. Singapore: World Scientific, 2003.
[10] M. C. Meng, "Comparing two reliability upper bounds for multistate systems," *Reliability Engineering and System Safety*, vol. 87, no. 1, pp. 31–36, 2005.
[11] B. Natvig, "Two suggestions of how to define a multistate coherent system," *Advance in Applied Probability*, vol. 14, no. 2, pp. 434–455, 1982.
[12] Z. Tian, M. J. Zuo, and R. C. M. Yam, "Performance evaluation of generalized multi-state k-out-of-n systems," in *Proceedings of the 2005 Industrial Engineering Research Conference*, Atlanta, USA, May 14–18, 2005.
[13] Z. Tian and M. J. Zuo, "The multi-state k-out-of-n systems and their performance evaluation," *IIE Transactions*, 2006, submitted for publication.
[14] M. J. Zuo and Z. Tian, "Performance evaluation for generalized multistate k-out-of-n systems," *IEEE Trans. Reliability*, vol. 55, no. 2, pp. 319–327, 2006.

Zhigang Tian is a Ph.D. student at the Department of Mechanical Engineering, University of Alberta, Canada. He received the B.S. degree in 2000, and M.S. degree in 2003, both in mechanical engineering from Dalian University of Technology, Dalian, China. His research interests include reliability engineering, condition based maintenance, mechanical design optimization, and operations research.

Richard C. M. Yam is an Associate Professor in the Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Hong Kong. His research interests include manufacturing strategies, innovation and technology management, and smart asset and maintenance management.

Ming J. Zuo is a Professor at the Department of Mechanical Engineering, University of Alberta, Canada. He received the Master of Science degree in 1986, and the Ph.D. degree in 1989, both in Industrial Engineering from Iowa State University, Ames, Iowa, U.S.A. His research interests include system reliability analysis, maintenance planning and optimization, signal processing, and fault diagnosis. He is a Department Editor, and an Editorial Board member of IIE Transactions. He is an Associate Editor of IEEE TRANSACTIONS ON RELIABILITY. He is a senior member of IEEE, and IIE.

Hong-Zhong Huang is a Professor in the School of Mechatronics Engineering, University of Electronic Science and Technology of China. He received a Ph.D. degree in Reliability Engineering from Shanghai Jiaotong University, Shanghai, China. Dr. Huang has published 120 journal articles, and five books in fields of reliability engineering, optimization design, and fuzzy sets theory. He is a (senior) member of several professional societies, and has served on the boards of professional societies.