Zhifeng LI
Zili WANG
Yi REN
Dezhen YANG
Xing LV

A NOVEL RELIABILITY ESTIMATION METHOD OF MULTI-STATE SYSTEM BASED ON STRUCTURE LEARNING ALGORITHM

NOWATORSKA METODA OCENY NIEZAWODNOŚCI SYSTEMÓW WIELOSTANOWYCH W OPARCIU O ALGORYTM UCZENIA STRUKTURY

Traditional reliability models, such as fault tree analysis (FTA) and reliability block diagram (RBD), are typically constructed with reference to the function principle graph that is produced by system engineers, which requires substantial time and effort. In addition, the quality and correctness of the models depend on the ability and experience of the engineers and the models are difficult to verify. With the development of data acquisition, data mining and system modeling techniques, the operational data of a complex system considering multi-state, dependent behavior can be obtained and analyzed automatically. In this paper, we present a method that is based on the K2 algorithm for establishing a Bayesian network (BN) for estimating the reliability of a multi-state system with dependent behavior. Facilitated by BN tools, the reliability modeling and the reliability estimation can be conducted automatically. An illustrative example is used to demonstrate the performance of the method.

Keywords: *reliability analysis, Bayesian network, structure learning, multi-state system (MSS), dependent failure.*

Tradycyjne modele niezawodności, takie jak analiza drzewa błędów (FTA) czy schemat blokowy niezawodności (RBD), buduje się zazwyczaj w oparciu o tworzone przez inżynierów systemowych schematy zasad działania systemu, których przygotowanie wymaga dużych nakładów czasu i pracy. Jakość i poprawność tych modeli zależy od umiejętności i doświadczenia inżynierów, a same modele są trudne do zweryfikowania. Dzięki rozwojowi technik akwizycji i eksploracji danych oraz modelowania systemów, dane operacyjne złożonego systemu uwzględniające jego zależne, wielostanowe zachowania mogą być pozyskiwane i analizowane automatycznie. W artykule przedstawiono metodę konstrukcji sieci bayesowskiej (BN) opartą na algorytmie K2, która pozwala na ocenę niezawodności systemu wielostanowego o zachowaniach zależnych. Dzięki narzędziom BN, modelowanie i szacowanie niezawodności może odbywać się automatycznie. Działanie omawianej metody zilustrowano na podstawie przykładu.

Słowa kluczowe: *analiza niezawodności, sieć bayesowska, uczenie struktury, system wielostanowy, uszkodzenie zależne.*

1. Introduction

For estimating the reliability of a complex system, constructing an accurate reliability model of the system is essential. A variety of popular reliability models are available, such as fault trees, reliability block diagrams, and Bayesian networks [28]. Unfortunately, they require professional knowledge and experience in modeling, along with a detailed understanding of the system structure and of how the system operates. Moreover, substantial effort is required for constructing these models for complex systems, even if the systems are medium-scale. Two main difficulties are encountered in building these models. (1) Modern systems consist of hardware and software with complex interactions, which are becoming increasingly difficult to model. (2) Reliability models and function principle graphs describe different aspects of the system. Although the designers know how the systems work, they lack the skills and experience in reliability modeling and analysis. There is a gap between system functional models and reliability models.

To overcome these problems, researchers proposed automatic transformation methods for converting the function principle graphs to fault trees [1, 20] or RBDs [11]. In addition, Bucci attempted to con-struct dynamic fault tree and event tree from corresponding Markov models such that time-dependent failure can be considered [2]. Moreover, the dynamic reliability models such as the dynamic fault tree (DFT) or dynamic reliability block diagram can be transformed into a dynamic Bayesian network to estimate the reliability of a dynamic system [17, 19, 21]. Even if reliability models could be automatically generated from function principle graphs, they would not reflect the changes that occur when the system is operating. In addition, when human factors are involved, it is difficult to incorporate these factors into the reliability models.

Methods such as the GO methodology and the Altarica project have been proposed for overcoming the problem of models not matching the specifications of the systems under study. The GO methodology uses a straightforward inductive logic to construct system models [18]. Enhanced methods improve the performance of the GO methodology and provide additional analysis results about the systems. A new quantification algorithm that is based on BDD is proposed in [5]. The quality analysis method, in combination with FMEA, is discussed in [12] and the dynamic behavior can be modeled via the extended GO methodology [25]. Altarica is a high-level modeling language

that can describe the hierarchies of the system and the behaviors of components of the system [7, 23]. The model can be compiled into lower-level formalisms for analysis of the reliability and dependability of the systems. The methods that are discussed above can provide models that are similar to the function principle graph; however, these methods are diffcult to use in practice due to their complexity.

Data-driven system reliability estimation methods such as the prognostics and health management (PHM) method were proposed many years ago. However, the traditional PHM methods are designed for specified operational conditions and the analysis results are limited to the systems that are in operation [30]. Hence, researchers studied how to obtain a generic model from data. Zaitseva [31] presented an approach that is based on a decision tree and learns the structure function of the system that represents the states of the system and components from the source data. The method is used not only to construct reliability models of the general system but also to estimate the reliability of the corresponding human factor system [13]. Zaitseva extended this method to multi-state systems [32]. However, the approach is only suitable for relatively simple systems that have no complicated interactions among components. Doguc [4] proposed a structure learning method that is based on Bayesian networks for constructing the reliability model. However, he only utilized a binary system and did not consider the interactive effects between components.

Dependent failure is an important behavior that can substantially affect the reliability of a system. Two types of dependent failure were identified in [27]: The first type can be described by the functional-dependent gate in DFT. The second type is found in multi-state systems. Thus, dependent failure should be considered in reliability models. Unfortunately, it is difficult to identify the dependencies between components and the interactive effects between components and subsystems.

In this paper, we propose a method that is based on the structure learning algorithm for modeling and estimating the reliability of multi-state systems while considering dependent failures. We focus on the dependency behavior between components.

The remainder of this paper is organized as follows: Section 2 briefly summarizes the multi-state system with correlative behavior. Section 3 presents a methodology for modeling the cause and effect relationships of multi-state systems that is based on the K2 algorithm [3] and evaluating conditional probability tables (CPTs). In Section 4, an example is used to demonstrate the entire modeling process. Section 5 presents the experimental analysis results on the accuracy and performance of the methodology. Finally, in Section 6, the conclusions of this work are presented and discussed.

2. Multi-state system with dependency behavior

Multi-state systems were introduced in 1968. Many researchers gradually contributed to the reliability theory of multi-state systems and developed a variety of methods for evaluating the reliability of multi-state systems. The related works are referenced in [14, 29].

For simplicity, traditional reliability modeling methods assume that the components are independent. However, this assumption is not practical for real engineering systems. In [27], the authors described several dependent failure scenarios, such as common cause failure, load-sharing, cascade failure [24], sequential failure, cross-system dependencies and interaction between components. These dependent failures would severely affect the reliability estimation of the systems.

Thus, researchers improved the traditional models to include dependent failures. A stochastic process [16] is an intuitive model for representing the correlations between components. Song [26] presented stochastic multivalued models for evaluating the reliability of an MSS with dependent multistate components. However, the space explosion problem poses substantial challenges for large-scale multi-

state systems. So researchers explored the combinatorial methods to model the multi-state system with dependent failure. Levitin [15] extended the universal generating function approach to multi-state systems with dependent elements. The dependent failure is of the second type [26]. Nagayama [22] analyzed the reliability of a multi-state system with partially dependent components based on the multi-valued decision diagrams.

In the works that are discussed above, the dependencies between components are formulated as conditional probabilities. The problem can be more easily represented by a Bayesian network, which can describe complicated relationships. Thus, BN is an alternative way of modeling a multi-state system with dependency behavior. Various BN tools such as BNT [6] support BN inference and implement many learning algorithms. Consequently, BN is a more suitable method for modeling multi-state systems with consideration of the dependency behavior.

3. Methodology

In this section, a reliability estimation method for multi-state systems that is based on the K2 algorithm [3] and the point estimation method is proposed. The dependent relationships among components, subsystems and the system are learned from data via the K2 algorithm. The parameters of the BN are estimated via the point estimation method. Then, the reliability of the multi-state system with consideration of the customer demand can be evaluated. Facilitated by BN tools, the whole process can be conducted automatically.

3.1. Structure learning algorithm K2

The K2 algorithm was proposed in [3] as a heuristic-search method. This algorithm assumes the following: 1) the variables are ordered and 2) all structures are equally likely. According to the assumptions, high-rank variables will not be the parents of low-rank variables. Hence, the search space of the parent sets of a variable can be reduced substantially.

The K2 algorithm consists of two main components:
1) A scoring function that quantifies the associations and ranks the parent sets according to their scores:

$$g(i, \pi_i) = \prod_{j=1}^{q_i} \frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!} \prod_{k=1}^{r_i} N_{ijk}! \quad (1)$$

i :   index of the components.

π_i :   set of parents of components x_i .

q_i : $|\phi_i|$.

ϕ_i :   list of all possible instantiations of the parents of x_i in database D, namely, if x_1 ,…, x_s are the parents of x_i , then ϕ_i is the Cartesian product $\{x_{10}, x_{11}, \cdots, x_{1j}, \cdots x_{1n_1}\} \times \cdots \times \{x_{s0}, x_{s1}, \cdots, x_{sj}, \cdots x_{sn_s}\}$. n_1 , n_s represent the numbers of the states of the components.

$r_i = |V_i|$.

V_i :   list of all possible state values of the components x_i .

α_{ijk} :   the number of cases in D in which component x_i is instantiated with its kth value and the parents of x_i in π_i are instantiated with the jth instantiation in ϕ_i .

$N_{ij} = \sum_{k=1}^{r_i} \alpha_{ijk}$: the number of instances in the database in which the parents of x_i in π_i are instantiated with the jth instantiation in ϕ_i .

To improve the run-time speed of K2, the logarithmic version of the equation above is implemented in this paper. The equation is described as follows:

$$\log(g(i, \pi_i)) = \sum_{j=1}^{q_i} \log(\frac{(r_i - 1)!}{(N_{ij} + r_i - 1)!}) + \sum_{k=1}^{r_i} \log(N_{ijk}!)$$

$$= \sum_{j=1}^{q_i} \left[\log((r_i - 1)!) - \log((N_{ij} + r_i - 1)!) \right] + \sum_{k=1}^{r_i} \log(N_{ijk}!) \quad (2)$$

$$= \sum_{j=1}^{q_i} \left[\log(\Gamma(r_i - 1)) - \log(\Gamma(N_{ij} + r_i - 1)) \right] + \sum_{k=1}^{r_i} \log(\Gamma(N_{ijk}))$$

where $\Gamma(\cdot)$ denotes the gamma function and the other parameters are as defined above;

2) A greedy-search method that incrementally adds nodes to the parent set to reduce the search space.

With the heuristic, the K2 algorithm does not need to consider all possible parents sets; it adds incrementally the parent whose addition most increases the probability of the resulting structure. If the addition of no single parent can increase the probability, no additional parents are added to the node: Initially, node x_i has no parents and the nodes $x_1, x_2, \cdots, x_{i-1}$ are candidates of the parent sets. The parent sets may be $\phi, \{x_1\}, \{x_2\}, \cdots, \{x_{i-1}\}, \{x_1, x_2\}, \cdots, \{x_1, x_{i-1}\}, \cdots, \{x_1, x_2, \cdots, x_{i-1}\}$. Hence, the parent set space is reduced substantially. In addition, to increase the efficiency of the algorithm, the K2 algorithm uses a parameter u to restrict the maximum number of parents. The pseudo-code of the K2 algorithm can be found in [3].

According to the description above, to use the K2 algorithm to learn the exact Bayesian network, the variables' ordering should correspond to the practical operational mode of the system. For example, a subsystem should be higher in the ordering than its child nodes. In addition, the parameter u should be set reasonably to balance the efficiency and the correctness of the Bayesian network. In [3], several suggestions are proposed for obtaining the most probable structure.

3.2. Probability distribution estimation

Via the K2 algorithm, the relationships among the components, subsystems and system can be obtained. To evaluate the reliability of MSS, the parameters of the Bayesian network should be estimated using the data.

Traditional reliability methods typically assume that the failure times of the components follow a probability distribution. However, this assumption may not hold in practice. Researchers considered integrating data from various levels of the system to reduce the uncertainty in the system reliability assessment. These methods [8, 9] can use the data to estimate the parameters of multi-state components. However, these approaches require prior knowledge about the system structure and about the cause and effect between components. These conditions are difficult to satisfy when we only have the function principle graph and the operation data of the system. Consequently, statistical estimation theory is used to determine the probability distribution.

There are two main types of estimation procedures in statistics: point and interval estimation. For convenience, point estimation is applied in this paper. Interval estimation is also applicable in our method.

First, according to the BN model that was learned from the data, the nodes without parents can be identified. The probability distributions of these nodes can be calculated easily. For node X_i, denotes the number of times that state j of component X_i occurs in the data as H_{ij} and the number of all instances in the data as H_i. The state probability distribution of X_i is presented in Table 1, where n_i denotes the number of states of component X_i.

Second, the conditional probability distributions of the nodes that have parents are estimated. For example, the structure of nodes X_i, X_j, X_k is illustrated in Fig. 1.
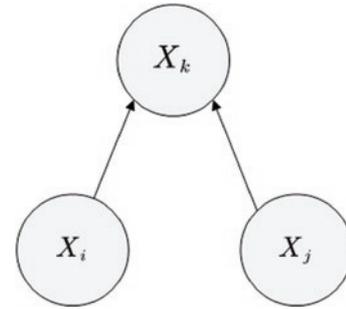

Fig. 1. Illustrative example

The CPT of X_k can be estimated via the following formula:

$$P\{X_k = c_{kr} \mid X_i = c_{iw}, X_j = c_{jv}\}$$
$$= \frac{P\{X_k = c_{kr}, X_i = c_{iw}, X_j = c_{jv}\}}{P\{X_i = c_{iw}, X_j = c_{jv}\}} \quad (3)$$
$$= \frac{H\{X_k = c_{kr}, X_i = c_{iw}, X_j = c_{jv}\}}{H\{X_i = c_{iw}, X_j = c_{jv}\}}$$

$H\{X_k = c_{kr}, X_i = c_{iw}, X_j = c_{jv}\}$ denotes the number of instances when $X_k = c_{kr}, X_i = c_{iw}, X_j = c_{jv}$. $H\{X_i = c_{iw}, X_j = c_{jv}\}$ is also available.

According to the method above, a BN that represents the dependent relationships and all parameters that represent the logical relationships can be obtained. Hence, the reliability of the MSS can be evaluated. Assume the system has M unique states. The reliability of the multi-state system can be expressed by the following formula:

$$R(t) = \sum_{s_i \geq w(t)}^{M} P\{S(t) = s_i\} \quad (4)$$

where $S(t)$ denotes the current state of the system and $w(t)$ denotes the demand for the system at time t. According to the model that is obtained above, the joint probability distribution $P(S, X_1, \cdots, X_j, \cdots, X_m)$ can be calculated. Thus, the marginal probability distribution $P\{S(t) = s_i\} = p_{s_i}$ is also obtained.

4. Illustrative example

4.1. Classification of Failure Conditions

In this paper, the task processing system example from [15] is used to demonstrate our method. The system logic diagram is shown as Fig. 2.

Table 1. Probability distribution of component

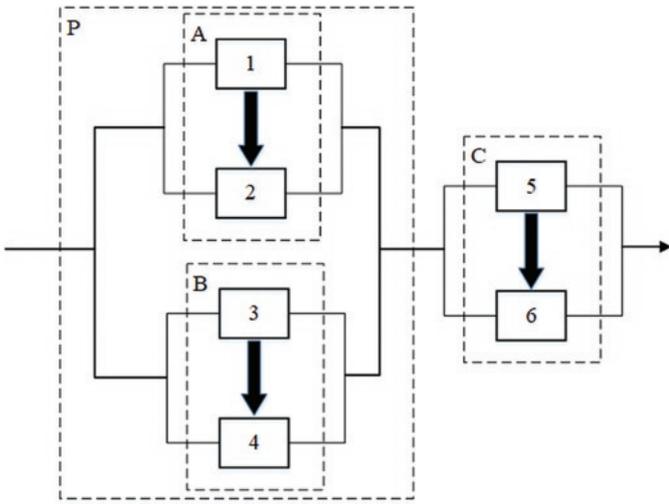| X_i	1	2	$\cdots$	j	$\cdots$	n_i
P	$\dfrac{H_{i1}}{H}$	$\dfrac{H_{i2}}{H}$	$\cdots$	$\dfrac{H_{ij}}{H}$	$\cdots$	$\dfrac{H_{in_i}}{H}$

Fig. 2. System logic diagram of the task processing system

The system consists of three independent computing blocks: A, B and C. Blocks A and B are constructed in parallel. Then, the parallel structure P and block C are arranged in series. Each block is composed of two processing units that differ in terms of priority. Elements 1, 3 and 5 have high priorities in blocks A, B and C. When the high-priority unit accesses a database, the low priority unit must to wait for the operation to be completed. Thus, the processing speed of the low-priority unit is affected by the load of the high-priority unit. The performance distributions of elements 1, 3 and 5 and the conditional performance distributions of elements 2, 4 and 6 can be found in [15].

The Bayesian network model of the task processing system is illustrated in Fig. 3. Nodes G_1, G_2, G_3, G_4, G_5, and G_6 represent processing units 1, 2, 3, 4, 5, and 6. Nodes A, B and C represent computing blocks A, B and C. Node P represents the subsystem that consists of blocks A and B. Node S represents the task processing system.

The BN is used to evaluate the performance in model learning from the data. In the next section, the logic sampling method [10] is used to randomly generate virtual instances to demonstrate the relationship between accuracy of the BN and the number of observations.
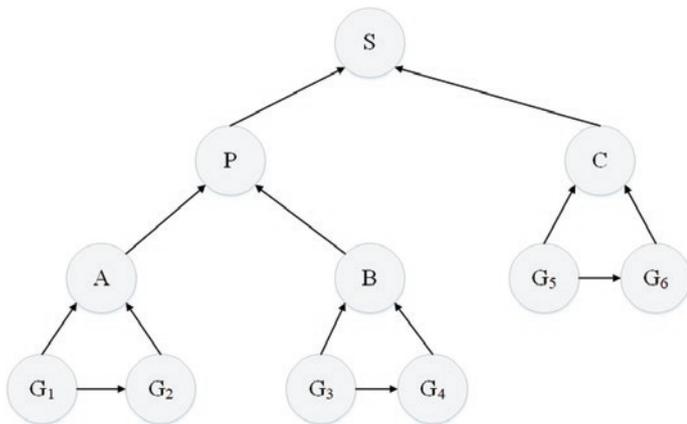

Fig. 3. Bayesian network of the task processing system

The performance of each element is discrete; hence, we treat the performance as the state of the element, namely, the performance and the state of the element have the same meaning in this paper.

The performance distributions of elements 1, 3 and 5 are presented in Table 2. The conditional performance distributions of elements 2, 4 and 6 are presented in Table 3, Table 4 and Table 5.

The conditional performance distributions of block A are presented in Table 6. Similarly, the conditional performance distributions of nodes B, C, P and S can be obtained.

The data obtained based on the monitoring and presented in Table 7, which collected 200 samples of the task processing system. The data set demonstrates some combinations of component states and the corresponding performance levels of the system. Though the data set is small, the relationships among the components and system can be inferred partially. The detail process is illustrated in the next subsection.

Table 2. *Performance distributions of elements 1, 3 and 5*

G_1	0	10	20	30	40	50
P_1	0.05	0.05	0.1	0.1	0.5	0.2
G_3	0	20	60			
P_3	0.1	0.7	0.2			
G_5	0	80	100			
P_5	0.1	0.2	0.7			

Table 3. *Conditional performance distribution of element 2*

G_1		0	10	20	30	40	50
	0	0.05	0.05	0.05	0.05	0.1	0.1
G_2	15	0.15	0.15	0.55	0.55	0.9	0.9
	30	0.8	0.8	0.4	0.4	0	0

Table 4. *Conditional performance distribution of element 4*

G_3		0	20	60
	0	0.05	0.05	0.05
G_4	15	0.15	0.35	0.95
	30	0.8	0.6	0

Table 5. *Conditional performance distribution of element 6*

G_5		0	80	100
	0	0.05	0.1	0.1
G_6	30	0.15	0.4	0.6
	50	0.8	0.5	0.3

4.2. Reliability estimation of the task processing system

According to the system logic diagram in Fig. 2, the ranking of the nodes depends on the hierarchy of the nodes, which will affect the efficiency of the K2 algorithm. The ranks and the number of performance value of the components are listed in Table 10.

Using the K2 algorithm, the associations among the components, subsystems and system can be identified and the score function values that correspond to parents can be obtained. The results are listed in

Table 6. Conditional probability distribution of block A

G_1	0			10			20			30			40			50		
G_2	0	15	30	0	15	30	0	15	30	0	15	30	0	15	30	0	15	30
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
30	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
50	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
60	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
65	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
70	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
80	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Table 7. Data obtained based on the monitoring of the task processing system

No.	G_1	G_2	A	G_3	G_4	B	P	G_5	G_6	C	S
1	40	15	55	60	15	0	55	100	30	130	55
2	40	15	55	20	30	50	105	0	50	50	50
3	50	15	65	20	15	35	100	0	50	50	50
4	20	30	50	0	30	30	80	100	30	130	80
5	40	15	55	60	15	75	115	80	50	130	115
6	10	30	40	20	30	50	90	80	30	110	80
7	20	15	35	0	30	30	65	80	50	130	65
8	10	30	40	20	30	50	90	80	30	110	90
9	0	15	15	20	30	50	65	100	0	100	65
10	0	0	0	20	30	50	50	0	50	50	50
…	…	…	…	…	…	…	…	…	…	…	…
200	30	30	60	20	30	50	110	100	50	150	110

Table 10. Ranks and the number of performance value of the components

Rank	Component/Subsystem	The number of performance value
1	G_1	6
2	G_2	3
3	A	15
4	G_3	3
5	G_4	3
6	B	9
7	P	33
8	G_5	3
9	G_6	3
10	C	8
11	S	30

Table 11. Associations and score function values

Component/Sub-system	Parent set	Score function value
G_1	ϕ	−292.63
G_2	$\{G_1\}$	−112.85
A	$\{G_1, G_2\}$	−184.89
G_3	ϕ	−165.17
G_4	$\{G_3\}$	−142.15
B	$\{G_3, G_4\}$	−93.49
G_5	ϕ	−173.42
G_6	$\{G_5\}$	−191.04
C	$\{G_5, G_6\}$	−107.88
P	$\{A, B\}$	−427.34
S	$\{P\}$	−429.06

Table 11 and the raw system structure inferred from the data is shown in Fig. 4. The system structure lacks of the relationship between component C and the system and the reason is that the data in Table 7 just represents partial state combinations of the system. The correlation of accuracy and data volume is analyzed in the next section.
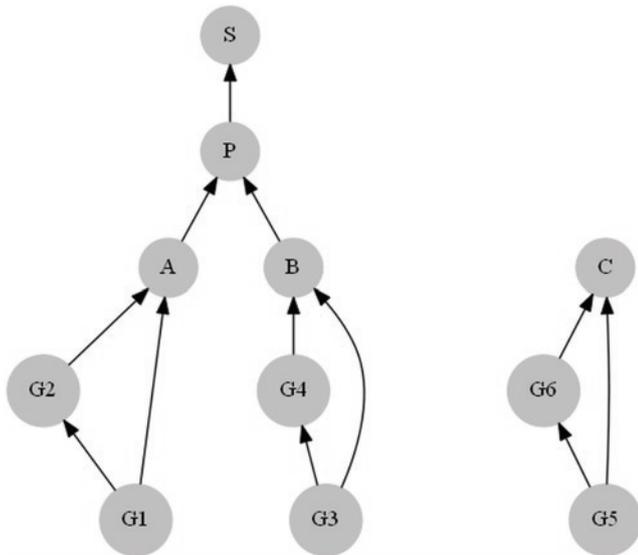
Fig. 4. The system structure inferred from the data

Table 12. Probability estimator of component G_1

G_1	0	10	20	30	40	50
p	0.04	0.05	0.06	0.11	0.515	0.225

Table 13. Conditional probability distribution estimator of component G_2

	G_1	0	10	20	30	40	50
	0	0	0	0	0	0.1165	0.1111
G_2	15	0.25	0.2	0.25	0.5	0.8835	0.8889
	30	0.75	0.8	0.75	0.5	0	0

The performance distribution parameters of the components or subsystems can be estimated by the data obtained based on the monitoring. The approximate performance distributions of elements G_1 and G_2 in the data are presented in Table 12 and Table 13. The performance distributions for the other elements can be obtained via the same approach.

Then, the reliability of the task processing system with consideration of the demand can be evaluated and the system reliability as a function of the demand is plotted in Fig. 5. The result is compared with the reliability that was calculated via the UGF method in Fig. 5.
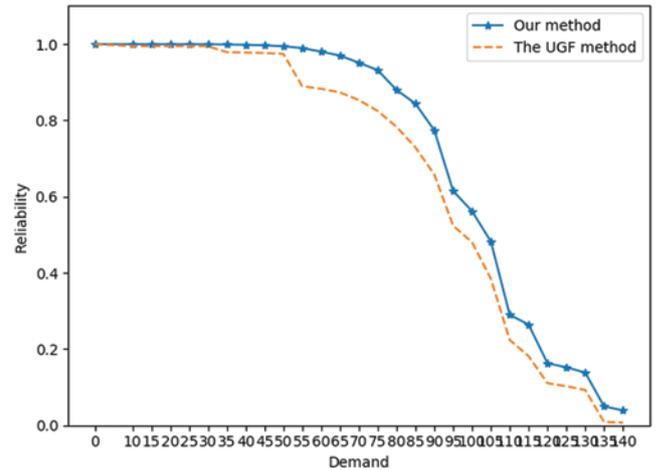

Fig. 5. Reliability comparison between our method and the UGF method

5. Experimental analysis

5.1. Generating random instances

The logic sampling generates an instance by randomly selecting values from the probability tables or conditional probability tables. The nodes are visited from the root nodes to the leaves; hence, all nodes in the BN can be instantiated once all nodes have been traversed. The detailed algorithm is available in [10].

Next, let us work through one round of simulation for the task processing system. Consider the root node G_1 of this network as an example. The random number generator produces a value between 0 and 1. Assume that the random number is 0.56. Then, the corresponding performance of unit 1 is 40, as illustrated in Fig. 6.
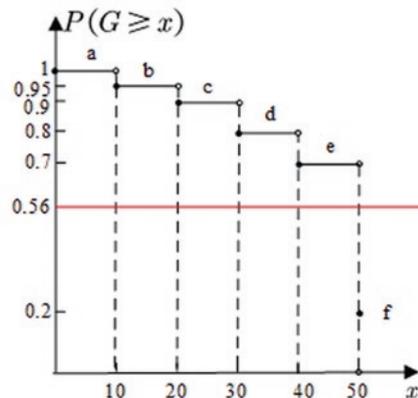

Fig. 6. Cumulative performance curve for processing unit 1

Table 7. *Performance values of all root nodes*

G$_1$	G$_3$	G$_5$
40	20	80

Table 8. *Performance instances of units 2, 4 and 6*

G$_2$	G$_4$	G$_6$
15	30	30

Table 9. *Full combinations of values in this simulation round*

G$_1$	G$_2$	G$_3$	G$_4$	G$_5$	G$_6$	A	B	C	P	S
40	15	20	30	80	30	55	50	110	105	105

Fig. 7. *BN models for various numbers of instances*

The performance values of other root nodes can be obtained as well. The performance instances of all root nodes are listed in Table 7.

Next, the value for child node G_2 should be generated. According to Table 3, the conditional cumulative performance curve for processing unit 2 can be calculated. The random number is 0.72; hence, the performance value of unit 2 is 15. The performance instances of units 2, 4 and 6 are listed in Table 8.

Then, the performance values of other nodes in the BN can be computed via the method that is demonstrated above. Finally, the full combinations of values in this simulation round are listed in Table 9.

The simulation is repeated 20000 times and 20000 sets of cases are generated as the operation data of the system. These cases form the training data set.

5.2. The correlation between accuracy and data volume

The number of observations that are used to discover the associations among nodes affects the efficiency of the K2 algorithm and the accuracy of the constructed BN. The error rate that is defined in [4] was used here to analyze the relationship between accuracy of the BN and the number of observations. The error rate can be calculated as follows:

$$\rho = \frac{A_{FP} + A_{FN}}{A_T} \tag{5}$$

A_{FP}: the number of associations that do not exist in the actual BN.

A_{FN}: the number of associations in the actual BN that are missed.

A_T: the number of the associations in the constructed BN.

As the amount of data increases, structure learned from data approximates the actual BN with increasing accuracy. The BN models that correspond to various numbers of instances are illustrated in Fig. 6.

The last Bayesian network has same structure as the model that is illustrated in Fig. 3. The BN that is constructed using a dataset with 1000 instances is similar to the actual BN of the task processing system. The BN that was built with 2500 instances has the same associations as the BN with 1000 instances. Hence, the additional data do not provide more information regarding the associations between block
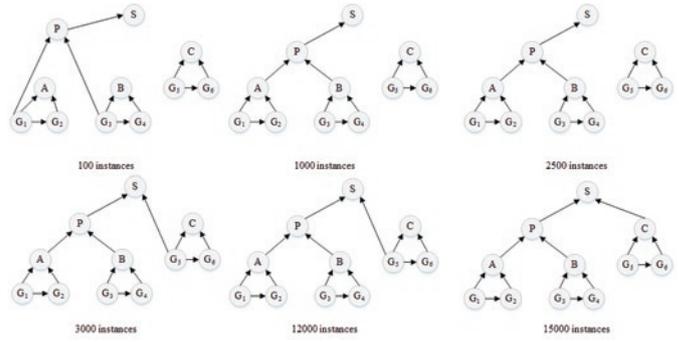
C and the system. With the increase in the amount of data, node G_5 becomes associated with node S. However, the performance of the system depends on block C and not on element 5. Although the additional data provide new information about the relation between the system and block C, the associations do not represent the real relationship. Moreover, the error rates of the BNs with 3000 and 12000 instances become higher than that of the BN with 2500 instances. In the course of training, the error rate may fluctuate with the increase in the amount of data before finally converging. The error rate curve is plotted in Fig. 7
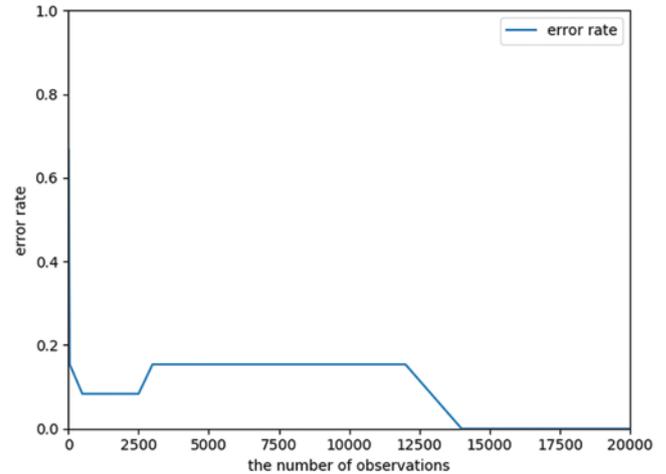

Fig. 8. *Error rate versus the number of observations*

According to Fig. 6 and Fig. 7, the dependent relationships of computing blocks A and B and the system are easily constructed; however, more data are required for accurately establishing the associations of computing block C and the system. That is because the task processing system is a series system and its performance depends on the minimal performance of subsystem P and block C. Hence, the accuracy may vary with the structure of the system.

6. Conclusions and Discussion

Traditional reliability methods strongly depend on the ability and experience of the engineers, which results in differences among reliability models that model the same system. With the development of data acquisition and data mining techniques, the operational data of

the systems can be monitored and analyzed for reliability estimation and system optimization. In this paper, a new method that is based on the K2 algorithm is proposed for constructing a reliability model of a system and for estimating the parameters of components from data. In the illustrative example, the Bayesian network model that is learned from the data is the same as the Bayesian network model that we constructed. Moreover, comparing with the universal generating function method, the results of the two methods are very close. Hence, this approach is effective.

According to experimental results, the efficiency of the method may depend on the structure of the systems. Thus, determining how to integrate the structure of the systems as prior knowledge into the process of structure learning will constitute our future work. This pa-per does not consider the scenario of missing data, which is common in practice. Determining how to handle missing data will be left for future work.

In conclusion, this method is suitable for reliability estimation of complex systems while considering multi-state and dependency behavior. Facilitated by BN tools, reliability modeling and reliability estimation can be conducted automatically without human intervention.

References

1. Bhagavatula A, Tao J, Dunnett S, et al. A new methodology for automatic fault tree construction based on component and mark libraries. Safety and Reliability 2016; 36(2): 62-76, https://doi.org/10.1080/09617353.2016.1219934.
2. Bucci P, et al. Construction of event-tree/fault-tree models from a Markov approach to dynamic system reliability. Reliability Engineering & System Safety 2008; 93(11): 1616-1627, https://doi.org/10.1016/j.ress.2008.01.008.
3. Cooper G F, Herskovits E. A Bayesian method for the induction of probabilistic networks from data. Machine Learning 1992; 9(4): 309-347, https://doi.org/10.1007/BF00994110.
4. Doguc O, Ramirez-Marquez J E. A generic method for estimating system reliability using Bayesian networks. Reliability Engineering & System Safety 2009; 94(2): 542-550, https://doi.org/10.1016/j.ress.2008.06.009.
5. Fan J, Ren Y, Liu L L. A new GO methodology algorithm based on BDD. Advanced Materials Research 2013; 791-793:1134-1138, https://doi.org/10.4028/www.scientific.net/AMR.791-793.1134.
6. François, Olivier & Leray, Philippe. BNT Structure Learning Package: Documentation and Experiments. 2008.
7. G. Point and A. Rauzy. AltaRica: Constraint automata as a description language. European Journal on Automation 1999; 33(8-9): 1033-1052.
8. Hao Z, Guo J, Zeng S. Fully Bayesian reliability assessment of multi-state systems with overlapping data. Journal of Systems Engineering and Electronics 2017; 28(1): 187-198, https://doi.org/10.21629/JSEE.2017.01.21.
9. Jackson C, Mosleh A. Bayesian inference with overlapping data: Reliability estimation of multi-state on-demand continuous life metric systems with uncertain evidence. Reliability Engineering & System Safety 2016; 145: 124-135, https://doi.org/10.1016/j.ress.2015.09.006.
10. Jensen F V. Bayesian Artificial Intelligence, second edition. Chapman & Hall/crc Boca Raton Fl 2010.
11. Liu X, Ren Y, Wang Z, et al. Modeling method of SysML-based reliability block diagram. Proceedings 2013 International Conference on Mechatronic Sciences 2013: 206-209, https://doi.org/10.1109/MEC.2013.6885073.
12. Liu L, Fan D, Wang Z, et al. Enhanced GO methodology to support failure mode, effects and criticality analysis. Journal of Intelligent Manufacturing 2019; 30(2): 1451-1468, https://doi.org/10.1007/s10845-017-1336-0.
13. Levashenko V, Zaitseva E, Kvassay M, et al. Reliability estimation of healthcare systems using Fuzzy Decision Trees. 2016 Federated Conference on Computer Science and Information Systems 2016: 331-340, https://doi.org/10.15439/2016F150.
14. Levitin, Gregory & Xing, Liudong. Multi-state systems. Reliability Engineering & System Safety 2017; 166: 1-2, https://doi.org/10.1016/j.ress.2017.06.008.
15. Levitin G. A universal generating function approach for the analysis of multi-state systems with dependent elements. Reliability Engineering & System Safety 2004; 84(3): 285-292, https://doi.org/10.1016/j.ress.2003.12.002.
16. Lesanovsky, A. Multistate Markov models for systems with dependent units. IEEE Transactions on Reliability 1988; 37(5): 505-511, https://doi.org/10.1109/24.9872.
17. Li K, Yi R, Ma Z. Reliability analysis of dynamic reliability blocks through conversion into dynamic bayesian networks. 2016 IEEE International Conference on Industrial Engineering and Engineering Management 2016: 1330-1334, https://doi.org/10.1109/IEEM.2016.7798094.
18. Matsuoka T, Kobayashi M, Takemura K. The GO-FLOW Methodology: A Reliability Analysis of the Emergency Core Cooling System of a Marine Reactor Under Accident Conditions. Nuclear Technology 1989; 84(3): 285-295, https://doi.org/10.13182/NT89-A34212.
19. Mi J, Li Y, Huang H Z, et al. Reliability analysis of multi-state systems with common cause failure based on Bayesian Networks. Eksploatacja i Niezawodnosc – Maintenance and Reliability.2013; 15(2): 169-175, https://doi.org/10.1109/ICQR2MSE.2012.6246417
20. Majdara A, Wakabayashi T, Soares C G. Component-based modeling of systems for automated fault tree generation. Reliability Engineering & System Safety 2009; 94(6): 1076-1086, https://doi.org/10.1016/j.ress.2008.12.003.
21. Montani S, Portinale L, Bobbio A, et al. RADYBAN : A tool for reliability analysis of dynamic fault trees through conversion into dynamic Bayesian networks. Reliability Engineering & System Safety 2008; 93(7), 922–932, https://doi.org/10.1016/j.ress.2007.03.013.
22. Nagayama, Shinobu & Sasao, Tsutomu & Butler, Jon & Thornton, Mitch & Manikas, Theodore. Analysis Methods of Multi-state Systems Partially Having Dependent Components Using Multiple-Valued Decision Diagrams. 2014 IEEE 44th International Symposium on Multiple-Valued Logic 2014: 190-195. https://doi.org/10.1109/ISMVL.2014.41.
23. Prosvirnova T. The AltaRica 3.0 Project for Model-Based Safety Assessment. 2013 11th IEEE International Conference on Industrial Informatics (INDIN) 2013: 741-746, https://doi.org/10.1109/INDIN.2013.6622976.
24. Peng R. Reliability of interdependent networks with cascading failures. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2018; 20 (2): 273–277, http://doi.org/10.17531/ein.2018.2.13.
25. Ren Y, Fan, Dongming, Wang, Zili, et al. System Dynamic Behavior Modelling based on Extended GO Methodology. IEEE Access 2018; 6: 22513-22523, http://doi.org/10.1109/ACCESS.2018.2816165.

26. Song X , Zhai Z , Liu Y , et al. A Stochastic Approach for the Reliability Evaluation of Multi-State Systems with Dependent Components. Reliability Engineering & System Safety 2018; 170: 257-266, https://doi.org/10.1016/j.ress.2017.10.015.
27. Sun Y, Ma L, Mathew J, et al. An analytical model for interactive failures. Reliability Engineering & System Safety 2006; 91(5): 495-504, https://doi.org/10.1016/j.ress.2005.03.014.
28. WANG H, DUAN F, MA J. Reliability analysis of complex uncertainty multi-state system based on Bayesian network. Eksploatacja i Niezawodnosc – Maintenance and Reliability 2019; 21(3): 419–429, http://dx.doi.org/10.17531/ein.2019.3.8.
29. Yingkui G , Jing L . Multi-State System Reliability: A New and Systematic Review. Procedia Engineering 2012; 29: 531-536, https://doi.org/10.1016/j.proeng.2011.12.756.
30. Zio, Enrico. Some Challenges and Opportunities in Reliability Engineering. IEEE Transactions on Reliability 2016; 65(4): 1769-1782, http://dx.doi.org/10.1109/TR.2016.2591504.
31. Zaitseva E , Levashenko V . Construction of a Reliability Structure Function Based on Uncertain Data. IEEE Transactions on Reliability 2016; 65 (4): 1710-1723, http://dx.doi.org/10.1109/TR.2016.2578948.
32. Zaitseva, E., Levashenko, V., Kvassay, M., & Rabcan, J. Application of Ordered Fuzzy Decision Trees in Construction of Structure Function of Multi-State System. In International Conference on Information and Communication Technologies in Education, Research, and Industrial Applications 2016: 56–75, http://dx.doi.org/10.1007/978-3-319-69965-3_4.

Zhifeng LI
Zili WANG
Yi REN
Dezhen YANG
Xing LV
School of Reliability and Systems Engineering
Beihang University
No.37 Xueyuan RD. Haidian, 100191, Beijing, China

E-mails: lizhifeng@buaa.edu.cn, wangzili@buaa.edu.cn, renyi@buaa.edu.cn, dezhenyang@buaa.edu.cn, lvxing@buaa.edu.cn