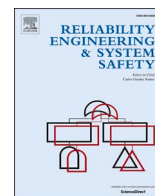


Contents lists available at [ScienceDirect](https://www.sciencedirect.com)

# Reliability Engineering and System Safety

journal homepage: [www.elsevier.com/locate/ress](http://www.elsevier.com/locate/ress)

## Generalized matrix-based Bayesian network for multi-state systems

Ji-Eun Byun<sup>a,1</sup>, Junho Song<sup>b,\*</sup>

<sup>a</sup> Department of Civil, Environmental and Geomatic Engineering, University College London, London, UK

<sup>b</sup> Department of Civil and Environmental Engineering, Seoul National University, Seoul, S. Korea

### ARTICLE INFO

#### Keywords:

Bayesian network (BN)  
Matrix-based Bayesian network (MBN)  
Multi-state system  
Large-scale system  
System reliability analysis  
BN inference

### ABSTRACT

To achieve a resilient society, the reliability of core engineering systems should be evaluated accurately. However, this remains challenging due to the complexity and large scale of real-world systems. Such complexity can be efficiently modelled by Bayesian network (BN), which formulates the probability distribution through a graph-based representation. On the other hand, the scale issue can be addressed by the matrix-based Bayesian network (MBN), which allows for efficient quantification and flexible inference of discrete BN. However, the MBN applications have been limited to binary-state systems, despite the essential role of multi-state engineering systems. Therefore, this paper generalizes the MBN to multi-state systems by introducing the concept of *composite state*. The definitions and inference operations developed for MBN are modified to accommodate the composite state, while formulations for the parameter sensitivity are also developed for the MBN. To facilitate applications of the generalized MBN, three commonly used techniques for decomposing an event space are employed to quantify the MBN, i.e. utilizing event definition, branch and bound (BnB), and decision diagram (DD), each being accompanied by an example system. The numerical examples demonstrate the efficiency and applicability of the generalized MBN. The supporting source code and data can be download at <https://github.com/jieunbyun/Generalized-MBN-multi-state>.

### 1. Introduction

The modern society relies on various types of complex engineering systems, which are characterized by their complex mechanisms and large scales, e.g. power systems, transportation networks, and oil distribution systems. Despite the importance of securing their functionality in normal and disaster situations, accurate evaluation of the reliability of those systems still remains a challenging task as system events call for collective consideration of the components, rather than conventional component-wise evaluations. To this end, researchers developed advanced system reliability methods that can identify and utilize the characteristics of the system event definitions [1,2,3]. However, these methods mostly focus on evaluating the marginal distribution of the system event, while taking into account only the component and system events. Furthermore, they often require specialized data structures and inference algorithms. As a result, they are not congenial either to incorporating additional variables, e.g. hazards and deterioration, or to deriving other inference tasks, e.g. conditional probability and parameter sensitivity.

Bayesian network (BN) can provide an efficient probabilistic model of multiple variables using its graph-based representation that can translate their real-world causal relationships into mathematical formulations [4]. In BN, the variables are represented by nodes, and each directed arrow stands for the causal relationship between a pair of variables. This strategy makes it straightforward to formulate the influence of the external factors on the component events. In addition, the dependence of a system event upon the component events can be explained simply by introducing arrows that head from the component nodes to the system node. Once a BN is modeled and quantified, BN inference algorithms can be applied to address various types of inference tasks. Especially for discrete BNs, i.e. BNs whose variables are all discrete, general-purpose inference algorithms are well-developed to facilitate the development of off-the-shelf software programs.

However, the conventional BN cannot handle large-scale systems effectively as the approach requires specifying the probability values for all possible joint states of the components. Since the number of such states exponentially increases with that of components, the computer memory demand quickly becomes insurmountable when one attempts

\* Corresponding author.

E-mail address: [junhosong@snu.ac.kr](mailto:junhosong@snu.ac.kr) (J. Song).

<sup>1</sup> formerly, Department of Civil and Environmental Engineering, Seoul National University, Seoul, S. Korea.

<https://doi.org/10.1016/j.ress.2021.107468>

Received 28 June 2020; Received in revised form 15 November 2020; Accepted 13 January 2021

Available online 16 January 2021

0951-8320/© 2021 The Authors.

Published by Elsevier Ltd.

This is an open access article under the CC BY-NC-ND license

(<http://creativecommons.org/licenses/by-nc-nd/4.0/>).

to quantify the probability mass function (PMF) of the system event. In order to avoid such exhaustive quantification, and thereby, to analyze large-scale systems within the BN methodology, the matrix-based Bayesian network (MBN) was developed by proposing a new matrix-based quantification of PMFs [5]. Thereby, the MBN can facilitate not only exploiting the regularities in the definition of the system event to achieve memory-efficiency, but also if necessary, performing approximate inference in the framework of BN to handle even larger systems that exact methods cannot address.

Despite its general applicability, the MBN has been applied only to binary-state systems so far [5,6], leaving its applicability to multi-state systems unexplored. The previously defined states in the MBN are not general enough to handle more than two states. Still, the reliability analysis of multi-state systems is essential as they cover a wide class of real-world systems, including mechanical systems [2], transportation networks [7], and utility distribution networks [8]. Although various methods have been proposed to efficiently quantify and inference multi-state systems [2,8], their complicated procedures makes it challenging for them to be combined with BN, which hampers performing comprehensive reliability analysis of those systems.

While multi-state systems show more rapid increase in memory demand than binary ones, both types of systems suffer the same issue, i.e. exponentially increasing memory demand with regard to the number of components, and therefore, can be addressed by similar approaches. Accordingly, this paper aims to generalize the MBN for multi-state systems by proposing the concept of *composite states* in contrast to the conventional states, namely *basic states*. The definitions and inference operations that were previously developed for the MBN are thus modified to be compatible with the proposed composite state. In addition, the MBN can also be employed for computing parameter sensitivity using the formulations newly developed in this paper.

On the other hand, while the MBN enables efficient quantification, the methodology itself does not suggest a specific way of practical implementation for given systems and inference tasks. While there is no general rule for this, this paper aims to provide insight for such development by presenting concrete applications of the MBN. For quantification, three most commonly used techniques are employed to deterministically decompose the event space, i.e. decomposition based on event definition, branch and bound (BnB), and decision-diagram (DD). The applications are demonstrated in detail by the example system events accompanying each approach, i.e. multi-state series-parallel (MS-SP) system, flow network, and multi-state *k-out-of-N:G* (MS-kN:G) system. For inference, various tasks are addressed in the numerical examples, which include system failure probability, component importance measure, parameter sensitivity, and distribution update. The applications also illustrate the approximate inference using the MBN.

The paper is organized as follows. Section 2 provides the theoretical background of BN. Then, Section 3 presents the proposed MBN methodology, including the previous development, generalized definitions and inference operations for multi-state systems, and formulations for parameter sensitivity. Section 4 illustrates how the MBN can be quantified using the existing techniques for decomposing event space. The discussions are illustrated by the numerical examples in Section 5 and summarized in Section 6.

In the following discussions, the upper and lower cases, e.g.  $X$  and  $x$ , denote the random variables (r.v.'s) and the assigned values over them, respectively, while the particular assignment  $X = k$  is abbreviated as  $x^k$ . The bolded letters are used when referring to a set of variables, and the cardinality of a set  $\mathbf{X}$  is denoted as  $|\mathbf{X}|$ . Also, the subtraction of sets,  $\mathbf{X} - \mathbf{Y}$  indicates  $\mathbf{X} \cap \mathbf{Y}^c$ .

## 2. Bayesian network

Bayesian network (BN) formulates a probability distribution based on a directed acyclic graph (DAG). In a BN, the random variables (r.v.'s)

are represented by circular nodes while their statistical dependence is indicated by directed arrows [4]. In this paper, the terms *node* and *variable* are used synonymously. The nodes connected to the tails and the heads of the arrows are respectively called *parent* and *child* nodes. Each node  $X$  is quantified by a probability distribution conditioned on the set of its parent nodes,  $Pa_X$ , i.e.  $P(X|Pa_X)$ . Thereby, the joint distribution  $P(\mathbf{X})$  over the variables  $\mathbf{X}$  in a BN can be described as the product of those conditional probability distributions, i.e.

$$P(\mathbf{X}) = \prod_{X \in \mathbf{X}} P(X|Pa_X) \tag{1}$$

where  $P(X|Pa_X)$  becomes a marginal distribution if  $Pa_X = \emptyset$ . Such graphical representation facilitates BN to exploit intuitive causal relationships between variables for formulating complex and high-dimensional joint probability distributions. Although Eq. (1) holds for both continuous and discrete distributions, the scope of this paper is limited to discrete ones. Therefore, in the following discussions, the notation  $P(\cdot)$  refers to only a probability mass function (PMF).

For instance, consider the flow capacity of a (physical) network in Fig. 1(a) where the uncertain flow capacities of the arcs are represented by the r.v.'s  $\mathbf{X}_N = \{X_1, \dots, X_N\}$  where  $N = 4$ . Fig. 1(b) shows the corresponding BN in which the r.v.'s in  $\mathbf{X}_N$  have  $H$  as their parent node. The r.v.  $H$  represents the hazard scenario of interest, while the arrows from  $H$  towards  $\mathbf{X}_N$  indicate that  $H$  affects the states of the component events  $\mathbf{X}_N$ . On the other hand, the system event  $X_{N+1}$  ( $= X_5$ ) survives if the states of  $\mathbf{X}_N$  are large enough to deliver the target demand  $d$  from nodes 1 to 4. Accordingly, the dependency of  $X_{N+1}$  on  $\mathbf{X}_N$  is reflected in the BN by the converging arrows from  $\mathbf{X}_N$  to  $X_{N+1}$ .

As also observed in Fig. 1(b), system events are characterized by the converging structure between component events and a system event in Fig. 2, which arises from their inherent dependency. Such structure makes it challenging for BN to handle large-scale systems because the PMF  $P(X_{N+1}|\mathbf{X}_N)$  has an exponentially increasing number of possible assignments over  $\{X_{N+1}\} \cup \mathbf{X}_N$  as that of components,  $N$  increases. As a result, while it is conventional to store the probabilities of all assignments using *table*-based data structure, namely conditional probability table (CPT), such conventional approach becomes infeasible even for a modest value of  $N$ . This raises the need for a new data structure that can make BN applicable for large-scale systems.

## 3. Matrix-based Bayesian network

### 3.1. Background, advantages, and limitation of previous development

To address the issue of exponential increase discussed in Section 2, Byun et al. [5] proposed the matrix-based Bayesian network (MBN) as an alternative data structure of BN. The key strategy is to introduce a *matrix*-based data structure for quantifying the conditional PMFs of a BN – namely, conditional probability matrices (CPMs). This approach is inspired by the matrix-based system reliability (MSR) method which separately stores the assignments and the probabilities using matrices so as to facilitate complicated probability computations [9]. Similarly, given a conditional PMF of a node, a CPM utilizes two matrices to store the assignments and the probabilities separately, namely event matrix  $\mathbf{C}$  and probability vector  $\mathbf{p}$ . In  $\mathbf{C}$  and  $\mathbf{p}$ , each row quantifies each of given instances, while each column of  $\mathbf{C}$  stands for each of the variables by storing the assignments over the corresponding variables. This strategy

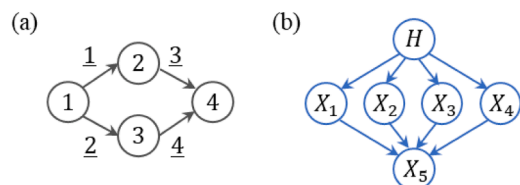


Fig. 1. (a) Topology of an example physical network, and (b) the corresponding BN

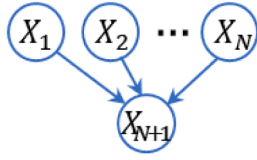


Fig. 2. Converging structure between component events and system event

allows for memory-efficient quantification and flexible BN inference, remaining insensitive to the BN graph such as converging structure [5].

The MBN achieves memory efficiency in two perspectives. First, the separate representation of assignments allows for defining the assignments more flexibly, which facilitates encoding the regularity in the definition of a system event. For instance, it is observed in the example network of Fig. 1 that the system fails (being disconnected) if arcs 1 and 2 are disabled, regardless of the states of arcs 3 and 4. Using the MBN, this observation can be quantified by a single row by introducing a state that stands for all states of  $X_3$  and  $X_4$ , e.g. defined as “-1” state in Byun *et al.* [5]. If  $X_3$  and  $X_4$  have three possible states each, then the single row can represent  $3^2 = 9$  instances that a conventional CPT would specify and store individually. Second, the row-wise storage allows the CPM to include only the instances of interest, e.g. those with positive probabilities and in case of optimization, those with positive utilities. For example, in the aforementioned event, i.e. the joint failures of arcs 1 and 2 result in the system failure, the system survival event is not stored in the CPM as it has a zero probability.

Moreover, the row-wise storage also exempts the MBN from quantifying the entire event space, i.e. the probabilities of the instances in CPM sum to less than one. This feature is essential for making the BN-based inferences applicable for approximate inference as well, which is inevitable as the number of components increases. Byun *et al.* [5] showed that the existing BN inference algorithms remain applicable even for such CPMs by developing the basic inference operations in terms of MBN, i.e. conditioning, sum, and product. These operations can be utilized to apply the existing algorithms such as variable elimination and clique trees [4]. In the paper, the formulations for error estimation were also provided for both cases where the instances are selected deterministically or stochastically.

Despite the potentially wide applicability of the MBN, the previous studies investigated only binary-state systems [5,6]. This is mainly due to the limitation of the previously introduced “-1” state, which can represent only the set of all states but not a subset of them. However, a wide class of real-world systems require modelling as multi-state systems for effective representations. Therefore, Sections 3.2 and 3.3 extend the MBN to multi-state systems by modifying the definitions and operations introduced in Byun *et al.* [5]. For completeness, the discussions also include the definitions and algorithms that do not require modifications. Thereafter, Section 3.4 develops formulations so as to apply the MBN to compute parameter sensitivity.

### 3.2. Generalized definitions for multi-state systems

As illustrated in Section 3.1, the MBN allows for flexibly defining the states of variables. In other words, one can collectively represent a subset of original states, namely *basic states*, by introducing some artificial state, namely *composite state*. In the followings, for a variable  $X$ , the set of basic states and that of *all states* (i.e. both basic and composite states) are respectively denoted as  $BS(X)$  and  $Val(X)$ . On the other hand, the set of basic states represented by a state  $x^k$  is denoted by  $B_X(k)$ . For illustrative purpose, the physical quantity that is represented by a basic state  $x^k$  is denoted as  $v(x^k)$ .

For instance, suppose in the BN of Fig. 1, the component events  $X_n \in \mathbf{X}_N$  have three possible states  $x_n^1, x_n^2,$  and  $x_n^3$  that represent the  $n$ -th arc having the flow capacity of 0, 20, and 30, respectively, i.e.  $v(x_n^1) = 0, v(x_n^2) = 20,$  and  $v(x_n^3) = 30$ . Meanwhile, the system event  $X_{N+1}$  reflects

whether the arc capacities are large enough to deliver the target flow  $d = 50$  from nodes 1 to 4, which leads  $X_{N+1}$  to take a binary-state of survival  $x_{N+1}^1$  and failure  $x_{N+1}^0$ . This setting results in the sets  $BS(X_n) = \{1, 2, 3\}$  for  $X_n \in \mathbf{X}_N$  and  $BS(X_{N+1}) = \{0, 1\}$ . In addition, for efficient quantification, composite states are introduced for  $X_n \in \mathbf{X}_N$  as  $B_{x_n}(4) = \{1, 2\}, B_{x_n}(5) = \{2, 3\},$  and  $B_{x_n}(6) = \{1, 2, 3\},$  resulting in  $Val(X_n) = \{1, 2, \dots, 6\}$

To enable such extended definition of states, the rows of the CPM are utilized as the quantification unit in the MBN, instead of the individual basic instances that the conventional approach utilizes. This distinctive unit is referred to as *rules*  $\mu = \langle c; p \rangle$  where assignments  $c$  and probability  $p$  respectively stand for the corresponding rows of  $C$  and  $p$ . The set of variables over which the given PMF is defined, i.e. those represented by the columns of  $C$ , is called *scope* which is denoted by  $Scope[\mu]$ . The formal definition of a rule is as follows

**Definition 1. (Rule in the MBN):** A “rule” in the MBN,  $\mu$  is a pair  $\langle c; p \rangle$  where  $c$  is a vector representing an assignment over a set of variables  $\mathbf{X}$ , and  $p \in [0, 1]$  is the corresponding probability. Inversely,  $X$  is defined as the *scope* of  $\mu$ , denoted by  $Scope[\mu]$ .

Then, a rule  $\mu = \langle c; p \rangle$  defined over a PMF  $P(\mathbf{X})$  implies that  $P(\mathbf{x}) = p$  for any assignment  $\mathbf{x}$  such that  $B(\mathbf{x}(X)) \subseteq B(c(X)), \forall X \in \mathbf{X}$

where  $c(X)$  denotes the assignment to  $X$  in  $c$ . For instance, in the aforementioned example, consider the event where the system fails, i.e.  $x_{N+1}^0$ , given the arcs 1 and 2 having zero capacity, i.e.  $x_1^1$  and  $x_2^2$ . In this case, the system always fails regardless of the states of  $X_3$  and  $X_4$ , which can be encoded by the states  $x_3^6$  and  $x_4^6$ . This produces the rule  $\mu = \langle c; p \rangle = \langle (0, 1, 1, 6, 6); 1 \rangle$  where the elements of  $c$  denote the states of  $X_{N+1}$  and  $X_1, \dots, X_4$  in sequence; and  $p = 1$  due to the deterministic definition of the system event. Inversely, the rule indicates that  $P(x_{N+1}^0, x_1^1, x_2^2, x_3^k, x_4^l) = 1$  for any  $k, l \in \{1, 2, 3\}$ . The scope of the rule is  $Scope[\mu] = \{X_{N+1}, X_1, \dots, X_4\}$

In order to make the rules coherent with the existing BN methods, the *compatibility* between two assignments having composite states should be defined. Two assignments are regarded being compatible if they indicate some common events. When the assignments consist only of basic states, the inspection is straightforward as the compatibility holds only when the assignments are identical for all variables in the common scope. However, composite states require some extra consideration since they can represent more than one instance: In this case, two assignments are compatible if and only if for all variables in their mutual scope, the assignments share at least one basic state. The formal definition is as follows:

**Definition 2. (Compatibility in the MBN):** An assignment  $c_1$  to  $\mathbf{X}$  is compatible with an assignment  $c_2$  to  $\mathbf{Y}$  if it holds that for all  $Z \in \mathbf{X} \cap \mathbf{Y}$ ,

$$B_Z(c_1(Z)) \cap B_Z(c_2(Z)) \neq \emptyset$$

If  $c_1$  is compatible with  $c_2$ , it is denoted by  $c_1 \sim c_2$ .

For example, consider the two assignments  $c_1 = (0, 1, 1, 6, 6)$  over  $\{X_5, X_1, \dots, X_4\}$  and  $c_2 = (1, 4)$  over  $\{X_2, X_3\}$ . Then, in regards to their mutual scope  $\{X_2, X_3\}$ , it is observed that  $B_{X_2}(1) \cap B_{X_2}(1) = \{1\}$  and  $B_{X_3}(6) \cap B_{X_3}(4) = \{1, 2\}$ ; and it is concluded that  $c_1 \sim c_2$ . On the other hand,  $c_3 = (2, 4)$  over  $\{X_2, X_3\}$  is not compatible with  $c_1$  for  $B_{X_2}(1) \cap B_{X_2}(2) = \emptyset$ .

The formal definition of CPM, which is the main data structure of PMFs in the MBN, can be established based on Definitions 1 and 2. As illustrated in Section 3.1, a CPM  $\mathcal{M} = \langle C; p \rangle$  quantifies a given PMF  $P(\mathbf{X}|\mathbf{U})$  by using the matrices  $C$  and  $p$ : The rows of the two matrices represent each rule, while the columns of  $C$  stand for each variable in the scope  $\mathbf{X} \cup \mathbf{U}$ . This leads to the following mathematical definition:

**Definition 3. (Conditional probability matrix):** A CPM of PMF  $P(\mathbf{X}|\mathbf{U})$  is a set of  $k$  rules  $\mathcal{M} = \{\langle c_1; p_1 \rangle, \langle c_2; p_2 \rangle, \dots, \langle c_k; p_k \rangle\}$  introduced such that:

- Each rule  $\mu \in \mathcal{M}$  has  $Scope[\mu] = \mathbf{X} \cup \mathbf{U}$ , which is also defined as  $Scope[\mathcal{M}]$ .

- For any assignment  $(\mathbf{x}, \mathbf{u})$  to  $\mathbf{X} \cup \mathbf{U}$  with only basic states, i.e. no composite states, there is either only one rule  $\langle \mathbf{c}; p \rangle \in \mathcal{M}$  such that  $\mathbf{c} \sim (\mathbf{x}, \mathbf{u})$ , in which case  $P(\mathbf{x}|\mathbf{u}) = p$ ; or no rule in  $\mathcal{M}$ , in which case it is considered  $P(\mathbf{x}|\mathbf{u}) = 0$ .
- CPM  $\mathcal{M}$  is represented as a pair  $\langle \mathbf{C}; \mathbf{p} \rangle$  where the rows of the matrix  $\mathbf{C}$  and the corresponding elements in the vector  $\mathbf{p}$  are respectively  $\mathbf{c}$  and  $p$  of the rules  $\langle \mathbf{c}; p \rangle \in \mathcal{M}$ .

It is noted that while a CPM is a set of rules in the mathematical point of view, it is a pair of matrices in the implementation point of view. In the followings, the subscript of a CPM denotes the child node being quantified, i.e.  $\mathcal{M}_X = \langle \mathbf{C}_X; \mathbf{p}_X \rangle$  quantifies  $P(X|Pa_X)$ . For simplicity, when there is no confusion, the CPM is quoted without the tuple representation, i.e.  $\langle \mathbf{C}; \mathbf{p} \rangle$  for the two matrices always inherit the subscript of the CPM.

In **Definition 3**, the only necessary condition for a valid CPM is the second bullet which requires all pairs of rules to be incompatible. Such condition is intuitive since any pair of compatible rules in a CPM would result in double-counting their common events. The only way to confirm this important condition for a given CPM is to check every pair of rules, which is not practical. A more efficient approach would be to ensure the condition while quantifying the CPM in the first place. This can be achieved by designing the quantification procedure in a way to generate disjoint sets, which is discussed in details in **Section 4**.

It should be also noted that **Definition 3** does not require a CPM to include the exhaustive set of instances, i.e. the sum of the probabilities of the rules may be less than one, which allows for *non-exhaustive* CPMs. This is particularly useful when the number of components is so large that the exhaustive quantification is unaffordable. In this case, a non-exhaustive CPM can be utilized to implement approximate inference into a BN. While the subset of rules can be selected either deterministically or stochastically depending on the employed methods, the way of selection determines the formulation of error bounds, for which the comprehensive illustrations can be found in Byun *et al.* [5].

As an example of quantifying a CPM,  $\mathcal{M}_{X_{N+1}}$  that represents the PMF  $P(X_{N+1}|\mathbf{X}_N)$  in **Fig. 1(b)** can be quantified using the two rules  $\langle (0, 1, 1, 6, 6); 1 \rangle$  and  $\langle (1, 3, 5, 3, 5); 1 \rangle$  over  $\{X_{N+1}, X_1, \dots, X_N\}$  as

$$\mathbf{C}_{X_{N+1}} = \begin{bmatrix} 0 & 1 & 1 & 6 & 6 \\ 1 & 3 & 5 & 3 & 5 \end{bmatrix} \text{ and } \mathbf{p}_{X_{N+1}} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \quad (2)$$

where the columns of  $\mathbf{C}_{X_{N+1}}$  sequentially denote the states of  $X_{N+1}$  and  $X_1, \dots, X_4$ . Although the rules are not representative of the complete set of the event  $\{X_{N+1}\} \cup \mathbf{X}_N$ , the CPM is still valid for they are not compatible with each other.

### 3.3. Generalized inference operations for multi-state systems

In this section, the basic operations of BN inference, i.e. *conditioning*, *sum*, and *product*, are developed for the MBN so that the existing BN inference algorithms, e.g. variable elimination, clique trees, and conditioning [4,5] can be applied to CPMs. To this end, given two assignments  $\mathbf{c}_1$  and  $\mathbf{c}_2$ , the *intersection* assignment  $\mathbf{c} = \mathbf{c}_1 \cap \mathbf{c}_2$  that represents their mutual instances, needs to be computed as follows:

**Definition 4. (Intersection assignment):** Consider an assignment  $\mathbf{c}_1$  to  $\mathbf{X}$  and an assignment  $\mathbf{c}_2$  to  $\mathbf{Y}$ . Then, the intersection assignment of  $\mathbf{c}_1$  and  $\mathbf{c}_2$ , denoted as  $\mathbf{c} = \mathbf{c}_1 \cap \mathbf{c}_2$ , has the scope  $\mathbf{X} \cup \mathbf{Y}$ , and the assigned values are computed for each  $Z \in \text{Scope}[\mathbf{c}]$  as

$$\mathbf{c}(Z) = \begin{cases} \mathbf{c}_1(Z), & Z \in \mathbf{X} - \mathbf{Y} \\ \mathbf{c}_2(Z), & Z \in \mathbf{Y} - \mathbf{X} \\ B_Z^{-1}(B_Z(\mathbf{c}_1(Z)) \cap B_Z(\mathbf{c}_2(Z))), & Z \in \mathbf{X} \cap \mathbf{Y} \end{cases}$$

where the inverse operation  $B_Z^{-1}(\mathbf{B}) = z$  such that  $B_Z(z) = \mathbf{B}$  for some  $\mathbf{B} \subseteq \text{BS}(Z)$  and  $z \in \text{Val}(Z)$ .

If  $\mathbf{c}_1$  and  $\mathbf{c}_2$  are not compatible, i.e. there is some  $Z \in \mathbf{X} \cap \mathbf{Y}$  such that  $B_Z(\mathbf{c}_1(Z)) \cap B_Z(\mathbf{c}_2(Z)) = \emptyset$ , above operation returns a null assignment. A null assignment implies that the rule is not included in the CPM.

For example, consider the assignments  $\mathbf{c}_1 = (0, 1, 1, 6, 6)$  over  $\{X_5, X_1, \dots, X_4\}$  and  $\mathbf{c}_2 = (1, 4)$  over  $\{X_2, X_3\}$ . Then, the intersection assignment  $\mathbf{c} = \mathbf{c}_1 \cap \mathbf{c}_2$  has the scope  $\{X_5, X_1, \dots, X_4\} = \{X_5, X_1, \dots, X_4\} \cup \{X_2, X_3\}$ . For the variables  $\{X_5, X_1, X_4\} = \{X_5, X_1, \dots, X_4\} - \{X_2, X_3\}$ , the intersection  $\mathbf{c}$  inherits the assignments in  $\mathbf{c}_1$ ; on the other hand, for the variables  $\{X_2, X_3\} = \{X_5, X_1, \dots, X_4\} \cap \{X_2, X_3\}$ ,  $\mathbf{c}$  has the assignments identified as  $B_{X_2}^{-1}(\{1\} \cap \{1\}) = 1$  over  $X_2$  and  $B_{X_3}^{-1}(\{1, 2, 3\} \cap \{1, 2\}) = 4$  over  $X_3$ . Accordingly, the intersection is obtained as  $\mathbf{c} = (0, 1, 1, 4, 6)$  over  $\{X_5, X_1, \dots, X_4\}$ .

Then, the first BN operation to be defined is conditioning which conditions a BN on a given assignment. In other words, this operation leaves in the CPM only the instances that are compatible with a given context  $\mathbf{e}$ , being defined as follows:

**Definition 5. (Conditioning operation in the MBN):** Consider a CPM  $\mathcal{M} = \langle \mathbf{C}; \mathbf{p} \rangle$  with  $\text{Scope}[\mathcal{M}] = \mathbf{X}$  and a context  $\mathbf{E} = \mathbf{e}$ . Then,  $\mathcal{M}$  is conditioned on  $\mathbf{E} = \mathbf{e}$ , by setting

$$\mathcal{M}[\mathbf{e}] = \{ \langle \mathbf{c}^*(\mathbf{X}); p \rangle : \mathbf{c}^* = \mathbf{c} \cap \mathbf{e} \text{ for } \langle \mathbf{c}; p \rangle \in \mathcal{M} \}$$

The sum operation aims to marginalize a variable out from a distribution. To this end, given a set of rules that are compatible over all variables but the variable to be summed out, the operation unifies these rules into a single rule with the probability being the sum of their probabilities. Although the definition has not been modified from the one proposed in Byun *et al.* [5], it is presented for completeness:

**Definition 6. (Sum operation in the MBN):** Let  $Y$  be a variable and  $\mu_i$ ,  $i = 1, 2, \dots, k$ , be the  $k$  rules of the form  $\mu_i = \langle \mathbf{c}, Y = y_i; p_i \rangle$ . Then for  $\mathcal{M} = \{ \mu_1, \mu_2, \dots, \mu_k \}$ , the sum is defined as  $\sum_Y \mathcal{M} = \langle \mathbf{c}; \sum_{i=1}^k p_i \rangle$ .

This definition leads to **Algorithm 1** for sum operation [4,5].

Finally, the product operation computes the product of two rules, i.e. the intersection of the two events. In other words, this operation returns a rule whose assignment and probability are respectively the intersection assignment and the product of the two given probabilities. The formal definition is as follows:

**Definition 7. (Product operation in the MBN):** Let  $\mu_1 = \langle \mathbf{c}_1; p_1 \rangle$  and  $\mu_2 = \langle \mathbf{c}_2; p_2 \rangle$  be two rules respectively with scopes  $\mathbf{X}$  and  $\mathbf{Y}$ . Then, their product

$$\mu^* = \mu_1 \cdot \mu_2 = \langle \mathbf{c}_1 \cap \mathbf{c}_2; p_1 \cdot p_2 \rangle \text{ with } \text{Scope}[\mu^*] = \mathbf{X} \cup \mathbf{Y}$$

This definition leads to **Algorithm 2** for the product operation.

The inference using the MBN requires the overheads, compared to the conventional CPT approach, of identifying the compatibility between the rules that may include composite states. However, when handling large-scale systems, the major bottleneck for BN analysis is not the computational cost for inference, but rather the memory required to store the individual instances whose number exponentially increases with that of component events. Moreover, the MBN has another advantage of making BN applicable for approximate inference, which allows for handling even larger systems.

### 3.4. Parameter sensitivity

Parameter sensitivity of probability can be computed using the MBN by replacing the elements of  $\mathbf{p}$  with the derivatives of interest [9]. Specifically, consider a parameter  $\theta$  and variables  $X_\theta \in \mathbf{X}_\theta$  for which the probabilities  $P(X_\theta|Pa_{X_\theta})$  are the functions of  $\theta$ . Then, the sensitivity with regard to the parameter  $\theta$  can be computed by replacing the CPM  $\mathcal{M}_{X_\theta} = \langle \mathbf{C}_{X_\theta}; \mathbf{p}_{X_\theta} \rangle$  with  $\mathcal{M}_{X_\theta, \theta} = \langle \mathbf{C}_{X_\theta}; \partial \mathbf{p}_{X_\theta} / \partial \theta \rangle$ . Since in BN, the joint distribution  $P(\mathbf{X})$  is the product of the distributions over the individual nodes as in **Eq. (1)**, its derivative is derived as

**Algorithm 1**  
Sum using CPMs.

---

```

Procedure CPM-sum (
     $\mathcal{M}$  // CPM
     $\mathbf{Y}$  // r.v.'s to be summed out
)
1    $\mathcal{M}^+ \leftarrow \mathcal{M}$ 
2    $\mathcal{M}^- \leftarrow \emptyset$ 
3   while  $\mathcal{M}^+ \neq \emptyset$ 
4       Select  $\mathcal{M}_c \subseteq \mathcal{M}^+$  such that
5            $\mathcal{M}_c = \{(c, \mathbf{Y} = \mathbf{y}_1; p_1), \dots, (c, \mathbf{Y} = \mathbf{y}_k; p_k)\}$ 
           (no other  $\mu \in \mathcal{M}^+$  is compatible with  $c$ )
6        $\mathcal{M}^- \leftarrow \mathcal{M}^- \cup \sum_{\mathbf{Y}} \mathcal{M}_c$ 
7        $\mathcal{M}^+ \leftarrow \mathcal{M}^+ - \mathcal{M}_c$ 
8   return  $\mathcal{M}^-$ 

```

---

**Algorithm 2**  
Product using CPMs.

---

```

Procedure CPM-product (
     $\mathcal{M}^1, \mathcal{M}^2$  // CPMs to be product
)
1    $\mathcal{M}^- \leftarrow \emptyset$ 
2   for each  $\mu_i = (c_1; p_1) \in \mathcal{M}^1$ 
3        $\mathcal{M}^- \leftarrow \mathcal{M}^- \cup \{\mu_i \cdot \mu : \mu \in \mathcal{M}^2\}$ 
4   return  $\mathcal{M}^-$ 

```

---

$$\frac{\partial P(\mathbf{X})}{\partial \theta} = \sum_{x_\theta \in X_\theta} \left( \frac{\partial P(X_\theta | Pa_{X_\theta})}{\partial \theta} \cdot \prod_{x \in X - \{X_\theta\}} P(X | Pa_X) \right) \quad (3)$$

The equation is different from Eq. (1) only for the derivative term, which can be accounted for by the replaced CPM  $\mathcal{M}_{x_\theta, \theta}$ . As indicated by the summation in the equation, if there is more than one  $X_\theta$ , the inference should be performed by replacing  $\mathcal{M}_{X_\theta}$  with  $\mathcal{M}_{x_\theta, \theta}$  for each  $X_\theta \in \mathbf{X}_\theta$  at a time, whereby the final result can be computed by summing up those individual evaluations.

#### 4. Deterministic MBN quantification for system events

The MBN provides an efficient means to encoding the system event  $P(X_{N+1} | \mathbf{X}_N)$  that is illustrated in Fig. 2. However, to this end, a specific method should be developed based on the characteristics of a given system event each time. This issue is particularly relevant to the case of deterministic quantification where the instances are identified deterministically in contrast to the sampling techniques that make random selection. In general, the quantification process is two-fold: First, it decomposes the event space  $\mathbf{X}_N$  into disjoint subsets consisting of the instances that lead to the same system state; then, each subset can be translated into a rule by using composite states of the component events. The smaller the number of the disjoint subsets is, the more the method is considered efficient.

For a concrete illustration of MBN quantification, in this section, the three most commonly used decomposition techniques of the event space are employed for MBN, each being accompanied by an example system type. During the implementation, it is noted that since CPMs do not need to include the exhaustive set of events, the quantification can be terminated without identifying all instances but only after achieving the desired precision. For such premature termination, the subsequent inference provides bounds on the query instead of the exact value, in

which case, the convergence of the bounds can be accelerated by decomposing the sets with larger probabilities with a priority. It is noteworthy that while, in this paper, only deterministic system events (i.e. where a combination of components states leads to a single system event with probability of one) are considered, the MBN can also be utilized for probabilistic system events (i.e. where a joint state of components may lead to several system states with certain probabilities) as long as a subset of component events lead to the same system event with the same probability.

##### 4.1. Decomposition based on system event definition

###### 4.1.1. Framework

Decomposing an event space can be facilitated by using the verbal definition of a given system event. For example, consider an event of reading the number on the upper face of a dice. Then, the decomposition becomes straightforward by describing the events in terms of such numbers, e.g. decomposition into three events by the numbers on the upper face such that  $\{1, 2\}$ ,  $\{3, 4\}$ , and  $\{5, 6\}$

###### 4.1.2. Example quantification: multi-state series-parallel system

A series-parallel system, illustrated in Fig. 3, is a series system of subsystems, each of which is a parallel system of components. This type of systems is representative of modular systems such as power systems and software programs. In order to allow the subsystems and the system to have multiple states, the system definition can be extended to a multi-state series-parallel (MS-SP) system [2]. In an MS-SP system, each component serves some specific capacity and takes a binary-state of fulfilling either the full or zero capacity. Then, each subsystem, as a parallel system, has the capacity equal to the sum of the capacities of the surviving components. The system event, on the other hand, being a series system, has the capacity as the minimum capacity of the

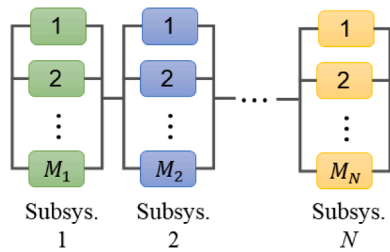


Fig. 3. Series-parallel system

subsystems. In this section, a decomposition algorithm is developed for an MS-SP system where in each subsystem, the components have the same capacity. The algorithm decomposes the event space based on verbal definitions.

In the followings,  $N$  and  $M_n$ ,  $n = 1, \dots, N$ , respectively denote the number of subsystems and that of components in the  $n$ -th subsystem. Then, the state of the  $m$ -th component in the  $n$ -th subsystem,  $m = 1, \dots, M_n$  and  $n = 1, \dots, N$ , is represented by the r.v.  $Y_{nm}$ , where the survival and the failure of the component event are respectively denoted by  $y_{nm}^1$  and  $y_{nm}^0$ . Finally, the capacities of the  $n$ -th subsystem and the system are represented by the r.v.'s  $X_n \in \mathbf{X}_N$  and  $X_{N+1}$ .

The event of an MS-SP system can be decomposed by defining the disjoint subsets as the events where “the  $n$ -th subsystem has the smallest capacity of  $C$ .” Then, the events are disjoint to each other for different  $n$  and  $C$ . In order to make the generated subsets strictly disjoint, though, the definition needs elaboration as “the  $n$ -th subsystem has the capacity  $C$  which is smaller than the capacities of the  $1, \dots, (n - 1)$ -th subsystems and smaller than or equal to the capacities of the  $(n + 1), \dots, N$ -th subsystems.” By considering all possible combinations of  $n$  and  $C$ , this definition can lead to quantifying the exhaustive set of the event. Since there are at most  $|BS(X_n)|$  possible values of  $C$  for the  $n$ -th subsystem, the maximum number of such combinations is  $\sum_{n=1}^N |BS(X_n)|$ . In contrast, it is noted that the conventional CPT would specify  $\prod_{n=1}^N |BS(X_n)|$  instances.

**Algorithm 3**

MBN quantification of MS-SP system.

**Procedure** MBN-quant-MSSP (

```

 $\mathbf{X}_N$  // set of subsystem events  $X_n$ ,  $n = 1, \dots, N$ 
)
1  $\mathcal{M}_{X_{N+1}} \leftarrow \emptyset$ 
2 for  $n = 1, \dots, N$ 
3   for each  $k \in BS(X_n)$ 
4      $v \leftarrow v(x_n^k)$ 
5      $c \leftarrow$  null-assignment over  $\{X_{N+1}, X_1, \dots, X_N\}$ 
6      $c(X_{N+1}) \leftarrow k'$  where  $v(x_{N+1}^{k'}) = v$ 
7     for  $n' = 1, \dots, (n - 1)$  and  $(n + 1), \dots, N$ 
8       if  $n' < n$  then
9          $K' \leftarrow \{x_{n'} \in BS(X_{n'}) : v(x_{n'}) > v\}$ 
10      else if  $n' > n$  then
11         $K' \leftarrow \{x_{n'} \in BS(X_{n'}) : v(x_{n'}) \geq v\}$ 
12      if  $K' \neq \emptyset$  then
13        Let  $k' \in Val(X_{n'})$  be a state such that  $B_{X_{n'}}(k') = K'$ 
14         $c(X_{n'}) \leftarrow k'$ 
15      else
16        break
17     $\mathcal{M}_{X_{N+1}} \leftarrow \mathcal{M}_{X_{N+1}} \cup \{(c; 1)\}$ 
18 return  $\mathcal{M}_{X_{N+1}}$ 

```

For the event corresponding to a pair of  $n$  and  $C$ , the subsystems other than the  $n$ -th one should have the capacities greater than (or equal to, depending on the index)  $C$ . Therefore, by utilizing the proper composite states that collectively represent such states, each event can be quantified by a rule. Algorithm 3 summarizes the quantification procedure. In the algorithm, the function  $v(\cdot)$  for the states of  $X_n \in \mathbf{X}_N$  ( $X_{N+1}$ ) denote the corresponding capacity of the subsystem (system).

As an illustrative example, consider an MS-SP system with  $N = 3$ ,  $M_1 = 3$ ,  $M_2 = 2$ , and  $M_3 = 1$ . The capacities of a single component in the subsystems 1, 2, and 3 are 10, 15, and 20 respectively, while the capacities of the surviving components are summed up to determine the capacities of the subsystems as summarized in Table 1. Then, following Algorithm 3, the rules for  $\mathcal{M}_{X_{N+1}}$  are created by going through  $n = 1, 2, 3$  and each  $k \in BS(X_n)$ . For example, for  $n = 1$  and  $k = 2$ , i.e.  $x_1^2$  with  $v(x_1^2) = 10$ , a rule is created over the assignments  $\{x_2^2, x_2^3\}$  and  $\{x_3^3\}$ , where the assignments indicate the capacities equal to or greater than 10. This leads to the rule in  $\mathcal{M}_{X_{N+1}}$

$$\langle (2, 2, 4, 2); 1 \rangle \tag{4}$$

where the assignments sequentially stand for the r.v.'s  $X_{N+1}$  and  $X_1, \dots, X_N$ ; and the system state  $x_{N+1}^2$  denotes the system having the capacity of 10. In the equation, the composite state  $x_2^4$  is introduced to represent the assignments  $\{x_2^2, x_2^3\}$ , i.e.  $B_{X_2}(4) = \{2, 3\}$ . In contrast, consider the assignment  $x_2^3$  where  $v(x_2^3) = 30$ . This assignment does not create any rule for there is no state of  $X_1$  corresponding to a greater capacity than

**Table 1**

$BS(X_n)$  and  $v_{X_n}(\cdot)$ ,  $n = 1, 2, 3$ , in the MS-SP system with  $N = 3$

$n$	1	2	3
$BS(X_n)$	{1, 2, 3, 4}	{1, 2, 3}	{1, 2}
$v(\cdot)$	$v(x_1^1) = 0;$ $v(x_1^2) = 10;$ $v(x_1^3) = 20;$ $v(x_1^4) = 30$	$v(x_2^1) = 0;$ $v(x_2^2) = 15;$ $v(x_2^3) = 30$	$v(x_3^1) = 0;$ $v(x_3^2) = 20$

30. Upon the completion of the algorithm, the MBN  $\mathcal{M}_{X_{N+1}} = \langle C_{X_{N+1}}; \mathbf{p}_{X_{N+1}} \rangle$  is quantified as

$$C_{X_{N+1}} = \begin{bmatrix} 1 & 1 & 5 & 3 \\ 2 & 2 & 4 & 2 \\ 4 & 3 & 3 & 2 \\ 1 & 6 & 1 & 3 \\ 3 & 5 & 2 & 2 \\ 1 & 6 & 4 & 1 \\ 4 & 4 & 3 & 2 \end{bmatrix} \text{ and } \mathbf{p}_{X_{N+1}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (5)$$

Table 2 summarizes the composite states of  $X_N$  and the states of  $X_{N+1}$  introduced for the quantification.

#### 4.2. Decomposition by branch and bound

##### 4.2.1. Framework

Branch and bound (BnB) is a decomposition technique whose procedure can be illustrated by a directed tree, i.e. for any node, there is exactly one path from the root node. In such graphical representation, the root node stands for the universal set of a given event, while other nodes correspond to the subset events created by the decomposition. In the process, decomposing a node is denoted by the directed arrows heading from the node to the next generated nodes (see Fig. 4 for an example). For a node, the decomposition is terminated if it is *specified*, i.e. the corresponding event consists of the instances that lead to the same state of the system. By construction, the final subsets are the end nodes of all branches.

After terminating the decomposition, while there can be some unspecified nodes, the specified ones among the end nodes can be used to generate the rules of  $\mathcal{M}_{X_{N+1}}$ , i.e.  $\mathcal{M}_{X_{N+1}}$  consists of as many rules as the number of the specified nodes. Such translation into rules would require introducing proper composite states.

##### 4.2.2. Example quantification: flow network

This section illustrates quantifying a flow network where the component events  $X_N$  represent the flow capacities of the  $N$  arcs, e.g. the example network in Fig. 1(a). While the events  $X_N$  can have multiple states, the system event  $X_{N+1}$  takes a binary state of survival ( $x_{N+1}^1$ ) if  $X_N$  have flow capacities large enough to deliver the target demand  $d$  from the source node  $s$  to the terminal node  $t$ ; and of failure ( $x_{N+1}^0$ ) otherwise. In the following illustrations,  $v(x_n^k)$  for  $X_n \in X_N$  denotes the corresponding arc capacity. It is also assumed without loss of generality that a higher state of  $X_n \in X_N$  indicates a larger capacity.

In the BnB, the decomposed sets are denoted by the upper and lower bounds of the component states,  $\mathbf{u} = (u_1, \dots, u_N)$  and  $\mathbf{l} = (l_1, \dots, l_N)$ . Thereby, if a set has  $\mathbf{u}$  that cannot deliver  $d$ , the set is specified as a failure set; or else if it has  $\mathbf{l}$  that can deliver the target flow  $d$ , it is specified as a survival set. On the other hand, an unspecified set can be efficiently decomposed by utilizing a  $d$ -flow  $\mathbf{f}^d = (f_1^d, \dots, f_N^d)$  [1]. A  $d$ -flow is a vector of flow values on the arcs when exactly the target flow  $d$  takes place from  $s$  to  $t$ . Such flow can be computed by introducing a new terminal node  $t'$  and an arrow with capacity  $d$  that heads from  $t$  to  $t'$ , to the (physical) network having the arc capacities as  $\mathbf{u}$ . With this modified network,  $f^d$  can be computed by performing maximum flow analysis from  $s$  to  $t'$ . For detailed discussion, readers are referred to Jane and Laih

[1].

The decomposition utilizes the fact that the subset with lower bound  $\mathbf{l}$  equal to or greater than  $\mathbf{f}^d$  can be specified as a survival set, aiming to arrive at this survival set at the end of the process (see Lines 5-28 in Algorithm 5). Specifically, let  $X_{n_1}, \dots, X_{n_m}$  be the decomposition ordering of  $X_n \in X_N$ . Following the order, the process separates out a set with the capacity of  $X_{n_i}$  smaller than  $f_{n_i}^d$ , retaining the set with the capacities equal to or greater than the  $d$ -flow value; then, it moves on to the next arc to further decompose the remaining set. This procedure leads the final remaining set to be specified as a survival set, while other sets produced during the decomposition require further flow analysis with their  $\mathbf{u}$  and  $\mathbf{l}$  to determine whether they can be specified. The numerical experiments suggest that the decomposition can become efficient by first considering the arcs with larger difference between the capacities corresponding to the lower bounds and the reference states for decomposition.

For instance, recall the network in Fig. 1(a) where the target demand  $d = 50$  from nodes 1 to 4; for  $X_n \in X_N$ ,  $v(x_n^1) = 0$ ,  $v(x_n^2) = 20$ , and  $v(x_n^3) = 30$ ; and  $B_{X_n}(4) = \{1, 2\}$ ,  $B_{X_n}(5) = \{2, 3\}$ , and  $B_{X_n}(6) = \{1, 2, 3\}$ . As illustrated in Fig. 4, the BnB starts with the universal set with upper bound  $(3, 3, 3, 3)$  and lower bound  $(1, 1, 1, 1)$ . While  $\mathbf{f}^d$  is not unique, suppose a  $d$ -flow has been obtained as  $(30, 20, 30, 20)$ . Then, the reference states for decomposition are evaluated as  $x_1^3, x_2^2, x_3^3$ , and  $x_4^2$ . Since  $v(x_n) - v(l_n)$  is 30, 20, 30, and 20 for  $n = 1, \dots, 4$ , the decomposition ordering is set as  $X_1, X_3, X_2$ , and  $X_4$ .

Consequently, as illustrated in the figure, the set is decomposed into the five disjoint sets with bounds  $\langle \mathbf{u}; \mathbf{l} \rangle$  such that  $\langle (2, 3, 3, 3); (1, 1, 1, 1) \rangle$ ,  $\langle (3, 3, 2, 3); (3, 1, 1, 1) \rangle$ ,  $\langle (3, 1, 3, 3); (3, 1, 3, 1) \rangle$ ,  $\langle (3, 3, 3, 1); (3, 2, 3, 1) \rangle$ , and  $\langle (3, 3, 3, 3); (3, 2, 3, 2) \rangle$ . While the last set is a survival set by construction, the maximum flow analysis over the upper bounds of the other sets reveals that the third and fourth sets are failure sets as their upper bounds show the maximum flow smaller than  $d = 50$ . Therefore, there are now two unspecified sets. In the figure, the further decomposition is made on the first set using the  $d$ -flow  $\mathbf{f}^d = (20, 30, 20, 30)$ . While there remains one unspecified set, one may continue or terminate the decomposition process depending on the desired level of precision and computational cost/memory. If one decides to terminate the process, the CPM  $\mathcal{M}_{X_{N+1}}$  is quantified as

$$C_{X_{N+1}} = \begin{bmatrix} 0 & 3 & 1 & 3 & 6 \\ 0 & 3 & 5 & 3 & 1 \\ 1 & 3 & 5 & 3 & 5 \\ 0 & 4 & 4 & 6 & 6 \\ 0 & 4 & 3 & 6 & 4 \\ 0 & 1 & 3 & 6 & 3 \\ 0 & 2 & 3 & 1 & 3 \\ 1 & 2 & 3 & 5 & 3 \end{bmatrix} \text{ and } \mathbf{p}_{X_{N+1}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (6)$$

where the columns of  $C_{X_{N+1}}$  stand for the states of  $X_{N+1}$  and  $X_1, \dots, X_4$ , in sequence. For instance, the survival set with  $\langle \mathbf{u}; \mathbf{l} \rangle = \langle (3, 3, 3, 3); (3, 2, 3, 2) \rangle$  identified by the first decomposition is quantified by the rule  $\langle (1, 3, 5, 3, 5); 1 \rangle$ , which is in the third row of the CPM.

Algorithm 4 summarizes the BnB-based MBN quantification of a flow network. In the algorithm,  $P(\mathbf{B})$  denotes the probability  $\sum_{\mathbf{b} \in \mathbf{B}} P(\mathbf{b})$  where  $\mathbf{B}$  and  $P(\mathbf{b})$  are respectively a set of decomposed events  $\mathbf{b}$  and the probability of  $\mathbf{b}$ .

**Table 2**  
Composite states of  $X_N$  and states of  $X_{N+1}$  in the MS-SP system with  $N = 3$

$n$	1	2	3
$B_{X_n}(\cdot)$	$B_{X_1}(5) = \{3, 4\};$ $B_{X_1}(6) = \{2, 3, 4\}$	$B_{X_2}(4) = \{2, 3\};$ $B_{X_2}(5) = \{1, 2, 3\}$	$B_{X_3}(3) = \{1, 2\}$
$v(\cdot)$	$v(x_{N+1}^1) = 0; v(x_{N+1}^2) = 10; v(x_{N+1}^3) = 15; v(x_{N+1}^4) = 20$		

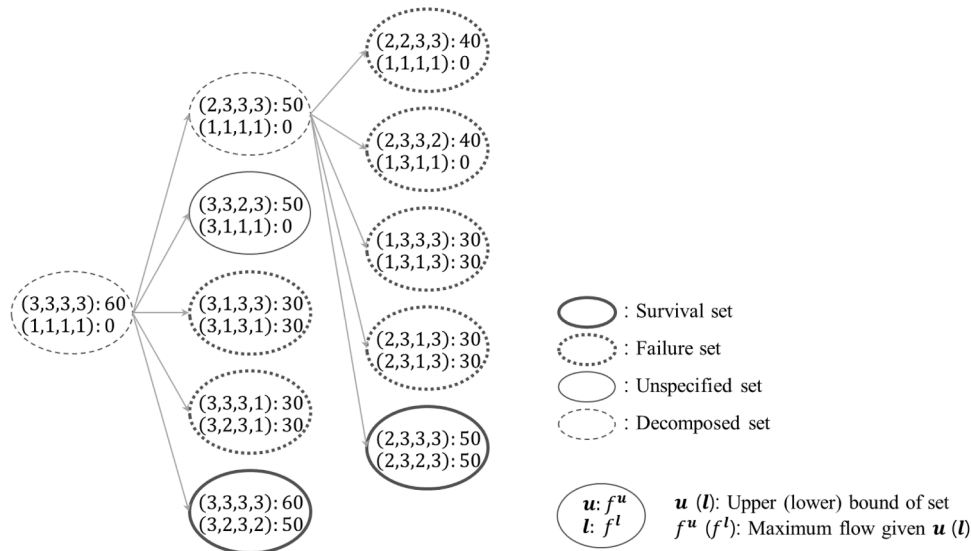


Fig. 4. BnB procedure for example flow network in Section 4.2.2

4.3. Decomposition by decision diagram

4.3.1. Framework

Similar to BnB, decision diagram (DD) procedure can be represented by a graph where the nodes and the arrows respectively stand for the decomposed subsets and the process of decomposition. Then, a node stops being decomposed when the node is specified, while the specified end nodes constitute the rules in  $\mathcal{M}_{X_{N+1}}$ . However, DD is distinguished from BnB in that it allows the paths to converge, i.e. a node can have more than one incoming arrow (for example, see Fig. 7 that illustrates an example DD procedure). Accordingly, in contrast to BnB which can be illustrated by a directed tree, DD is represented by a DAG [10].

As a trade-off of the converging paths, the DD-based decomposition requires the BN to introduce additional r.v.'s that account for the intermediate nodes on the paths in DD. As an example, Fig. 5 illustrates the modified BN from the one in Fig. 2 where the sequential counting of the component events  $X_1, \dots, X_N$  produces the intermediate nodes  $\bar{X}_1, \dots, \bar{X}_{N-1}$ . Then, each arrow heading to the next intermediate stage (the final stage) produces a rule in the CPMs  $\mathcal{M}_{\bar{X}_n} (\mathcal{M}_{X_{N+1}})$ . Accordingly, DD is preferred when the intermediate quantification results can be efficiently summarized into a handful of cases. When there is no such way of effective summary, BnB should be preferred for it does not require modification on BN.

4.3.2. Example quantification: multi-state k-out-of-N:G system

In this section, a k-out-of-N:G system is quantified where the system survives given that there are equal to or more than k surviving components among a total of N components. In order to extend this definition so that the component events can have multiple states, Huang and Zuo [8] proposed a multi-state k-out-of-N:G (MS-kN:G) system by introducing additional properties such that: (1) given M component states, each component state is associated with a demand k, which brings about k values as many as component states, i.e.  $k_m, m = 1, \dots, M$ ; and (2) the system survives given that for all m, there are at least  $k_m$  component events whose states are equal to or greater than m. Accordingly, the component events  $X_n, n = 1, \dots, N$ , have basic states  $BS(X_n) = \{0, 1, \dots, M\}$  where  $x_n^0$  indicates that none of the M states is achievable. On the other hand, the system event  $X_{N+1}$  takes the binary-state of either survival ( $x_{N+1}^1$ ) or failure ( $x_{N+1}^0$ ). An MS-kN:G system is representative of systems where the component states have some hierarchy. For instance, consider the oil distribution network in Fig. 6 where the component state is defined as the index of the farthest station that a pipeline can

reach [8]. Such definition creates a hierarchy between the states for a farther station can be reached only once all of the closer stations have been reached.

In order to quantify an MS-kN:G system, a DD-based approach can be employed where the component events  $X_n, n = 1, \dots, N$ , are sequentially counted, producing the intermediate nodes  $\bar{X}_n, n = 1, \dots, (N-1)$ , that represent the fulfilled demands by  $X_1, \dots, X_n$  [3]. During the quantification, the states  $\bar{x}_n^1$  and  $\bar{x}_n^0$  are reserved to represent the system survival and failure, respectively, for the case where the system event can be specified at the intermediate stage without counting the remaining component events.

As an illustrative example, consider an MS-kN:G system with  $N = 4, M = 2$ , and  $\mathbf{k} = (3, 2)$ . The DD for quantifying this system is given in Fig. 7. In the DD, each node stands for an assignment of  $\bar{X}_n, n = 1, \dots, (N-1)$ , which can be specified by the two vectors  $\mathbf{q}$  and  $\bar{\mathbf{q}}$ . In the figure, the upper vector  $\mathbf{q} = (q_1, \dots, q_M)$  denotes the number of components, among the counted ones, whose states are equal to or greater than m,  $m = 1, \dots, M$ . On the other hand, the lower vector  $\bar{\mathbf{q}} = (\bar{q}_1, \dots, \bar{q}_M)$  indicates the number of components that are additionally required to meet the demand at the state m, i.e.  $\bar{q}_m = k_m - q_m$ . By definition, the sum of the two vectors is equal to the demand  $\mathbf{k}$ , i.e.  $\mathbf{q} + \bar{\mathbf{q}} = \mathbf{k}$ . The two vectors can be utilized as a reference for the decomposition. For a node  $\bar{x}_n$ , if  $\mathbf{q}$  is observed such that  $\mathbf{q} \geq \mathbf{k}$ , i.e.  $q_m \geq k_m$  for all m, the state  $\bar{x}_n$  is specified as  $\bar{x}_n^1$  for the node always leads to the system survival event regardless of the states of the uncounted component events. In contrast, when a node has some m for which  $\bar{q}_m$  is greater than the number of remaining components, the node is specified as  $\bar{x}_n^0$  since it can never fulfill the demand  $k_m$ .

In Fig. 7, the DD starts with the universal set  $\Omega$  with  $\mathbf{q} = (0, 0)$ . Then, the set is decomposed into three subsets by the three states of  $X_1$ , where the subsets are represented by the states  $\bar{x}_1^2, \bar{x}_1^3$ , and  $\bar{x}_1^4$ . For example, by the state  $x_1^0, \Omega$  is decomposed into the event of  $\bar{x}_1^2$  where  $\mathbf{q} = (0, 0) + (0, 0) = (0, 0)$  for  $X_1$  cannot satisfy any of the states. On the other hand, given  $x_1^1$ , the event is decomposed to that of  $\bar{x}_1^3$  where  $\mathbf{q} = (0, 0) + (1, 0) = (1, 0)$ . This process is continued until all component events are counted.

In the figure, it is noted that the intermediate result, i.e. the number of components that fulfill each state, can be summarized into a few states of  $\bar{X}_n$ , which makes the DD efficient for this type of systems. Furthermore, a number of subsets can be specified in the middle of the quantification, which makes the quantification even more efficient by eliminating the need for counting the remaining components. This is



**Algorithm 4**

MBN quantification of a flow network.

```

Procedure MBN-quant-Flow (
 $\mathcal{G}^d$  // augmented (physical) network for  $d$ -flow
 $X_n, n = 1, \dots, N$  // component events
 $P(X_N)$  // probability distribution over  $X_N$ 
 $\bar{p}$  // threshold of unspecified probability
)
1  $S \leftarrow \emptyset$  // set of survival sets
2  $F \leftarrow \emptyset$  // set of failure sets
3 Set  $\mathbf{u}$  and  $\mathbf{l}$  be the largest and smallest states of  $X_N$ 
4  $B \leftarrow \{\langle \mathbf{u}; \mathbf{l} \rangle\}$  // set of unspecified sets
// Decomposition by BnB
5 while  $P(U) > \bar{p}$ 
6   Select  $\mathbf{b} = \langle \mathbf{u}; \mathbf{l} \rangle \in B$  with largest  $P(\mathbf{b})$ 
7    $B \leftarrow B - \{\mathbf{b}\}$ 
8   Evaluate  $d$ -flow  $f^d$  using  $\mathbf{u}$  and  $\mathcal{G}^d$ 
9    $X_{N'} \leftarrow \emptyset$ 
10  for  $n = 1, \dots, N$ 
11    if  $f_n^d > l_n$  then
12       $X_{N'} \leftarrow X_{N'} \cup \{X_n\}$ 
13       $\mathbf{x}'(X_n) \leftarrow \min\{x_n \in BS(X_n): v(x_n) \geq f_n^d\}$ 
14      Order  $X_{N'}$  by the descending order of  $v(x_{n'}) - v(l_{n'})$ 
15       $\mathbf{u}' \leftarrow \mathbf{u}, \mathbf{l}' \leftarrow \mathbf{l}, B' \leftarrow \emptyset$ 
16      for each  $X_{n'} \in X_{N'}$ 
17         $u_{n'}' \leftarrow x_{n'}' - 1$ 
18         $B' \leftarrow B' \cup \{\langle \mathbf{u}'; \mathbf{l}' \rangle\}$ 
19         $u_{n'}' \leftarrow u_{n'}$ 
20         $l_{n'}' \leftarrow x_{n'}'$ 
21       $S \leftarrow S \cup \{\langle \mathbf{u}'; \mathbf{l}' \rangle\}$ 
22      for each  $\mathbf{b} \in B'$ 
23        if  $\mathbf{l}$  can deliver  $d$  from  $s$  to  $t$  then
24           $S \leftarrow S \cup \{\mathbf{b}\}$ 
25        else if  $\mathbf{u}$  cannot deliver  $d$  from  $s$  to  $t$  then
26           $F \leftarrow F \cup \{\mathbf{b}\}$ 
27        else
28           $B \leftarrow B \cup \{\mathbf{b}\}$ 
// MBN quantification
29  $\mathcal{M}_{X_{N+1}} \leftarrow \emptyset$ 
30 for each  $\mathbf{b} = \langle \mathbf{u}; \mathbf{l} \rangle \in S$ 
31   for  $n = 1, \dots, N$ 
32      $\mathbf{x}(X_n) \leftarrow B_{X_n}^{-1}(\{l_n, \dots, u_n\})$ 
33      $\mathcal{M}_{X_{N+1}} \leftarrow \{(1, \mathbf{x}); 1\}$ 
34 for each  $\mathbf{b} = \langle \mathbf{u}; \mathbf{l} \rangle \in F$ 
35   for  $n = 1, \dots, N$ 
36      $\mathbf{x}(X_n) \leftarrow B_{X_n}^{-1}(\{l_n, \dots, u_n\})$ 
37      $\mathcal{M}_{X_{N+1}} \leftarrow \{(0, \mathbf{x}); 1\}$ 
38 return  $\mathcal{M}_{X_{N+1}}$ 

```

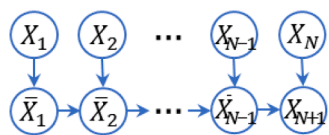


Fig. 5. Modified BN for DD-based quantification

indicated by the arrows connected to  $\bar{x}_n^1$  and  $\bar{x}_n^0, n = 1, \dots, N - 1$ . For instance,  $\bar{x}_2^0$  has only one remaining demand at  $m = 1$ , by which the node can be specified as a system survival event given that the state of  $X_3$  is equal to or greater than 1, i.e.  $x_3^1$  and  $x_3^2$ . On the other hand,  $\bar{x}_2^2$  has to fulfill two more demands for  $m = 2$  while there are only two remaining components. Accordingly, if the state of  $X_3$  is smaller than 2, i.e.  $\bar{x}_3^0$  and  $\bar{x}_3^1$ , the subsequent event becomes a system failure event.

To quantify the MBN using the DD result, the first step is to quantify

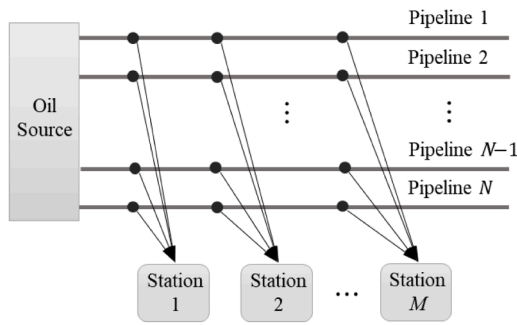


Fig. 6. Oil distribution network adopted from Huang and Zuo [8]

the intermediate nodes  $\bar{X}_n, n = 1, \dots, (N-1)$ . For example, the left column of nodes in Fig. 7 leads to the MBN  $\mathcal{M}_{\bar{X}_1} = \langle C_{\bar{X}_1}; \mathbf{p}_{\bar{X}_1} \rangle$  for  $P(\bar{X}_1|X_1)$  such that

$$C_{\bar{X}_1} = \begin{bmatrix} 2 & 0 \\ 3 & 1 \\ 4 & 2 \end{bmatrix} \text{ and } \mathbf{p}_{\bar{X}_1} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad (7)$$

where the first and the second columns of  $C_{\bar{X}_1}$  respectively represent the states of  $\bar{X}_1$  and  $X_1$ . Then, the middle column of nodes quantifies  $\mathcal{M}_{\bar{X}_2} = \langle C_{\bar{X}_2}; \mathbf{p}_{\bar{X}_2} \rangle$  for  $P(\bar{X}_2|\bar{X}_1, X_1)$  as

$$C_{\bar{X}_2} = \begin{bmatrix} 0 & 0 & 2 \\ 2 & 1 & 2 \\ 3 & 2 & 2 \\ 2 & 0 & 3 \\ 4 & 1 & 3 \\ 5 & 2 & 3 \\ 3 & 0 & 4 \\ 5 & 1 & 4 \\ 6 & 2 & 4 \end{bmatrix} \text{ and } \mathbf{p}_{\bar{X}_2} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (8)$$

where the columns of  $C_{\bar{X}_2}$  represent the states of  $\bar{X}_2, X_2,$  and  $\bar{X}_1$  in order. Thereafter, the right column leads to  $\mathcal{M}_{\bar{X}_3} = \langle C_{\bar{X}_3}; \mathbf{p}_{\bar{X}_3} \rangle$  such that

$$C_{\bar{X}_3} = \begin{bmatrix} 0 & 5 & 0 \\ 0 & 3 & 2 \\ 2 & 2 & 2 \\ 0 & 0 & 3 \\ 2 & 1 & 3 \\ 3 & 2 & 3 \\ 0 & 3 & 4 \\ 4 & 2 & 4 \\ 2 & 0 & 5 \\ 4 & 1 & 5 \\ 1 & 2 & 5 \\ 3 & 0 & 6 \\ 1 & 4 & 6 \end{bmatrix} \text{ and } \mathbf{p}_{\bar{X}_3} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (9)$$

where the columns of  $C_{\bar{X}_3}$  sequentially denote the states of  $\bar{X}_3, X_3,$  and  $\bar{X}_2$ . For the quantification, the composite states of component events are set up as  $B_{X_n}(3) = \{0, 1\}, B_{X_n}(4) = \{1, 2\},$  and  $B_{X_n}(5) = \{0, 1, 2\}$ . Finally, the system event  $X_{N+1}$  is quantified by the CPM  $\mathcal{M}_{X_{N+1}} = \langle C_{X_{N+1}}; \mathbf{p}_{X_{N+1}} \rangle$  that represents  $P(X_{N+1}|X_N, \bar{X}_{N-1})$  as

$$C_{X_{N+1}} = \begin{bmatrix} 0 & 5 & 0 \\ 0 & 3 & 2 \\ 1 & 2 & 2 \\ 0 & 0 & 3 \\ 1 & 4 & 3 \\ 0 & 3 & 4 \\ 1 & 2 & 4 \\ 1 & 5 & 1 \end{bmatrix} \text{ and } \mathbf{p}_{X_{N+1}} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (10)$$

where the columns of  $C_{X_{N+1}}$  sequentially stand for  $X_{N+1}, X_N,$  and  $\bar{X}_{N-1}$ .

Algorithm 5 summarizes the quantification process for the general MS-kN:G system. In the algorithm, the vector  $\bar{\mathbf{I}}_{M,k}$  denotes the  $M$ -dimensional vector where the first,  $\dots, k$ -th elements are 1 while the  $(k +$

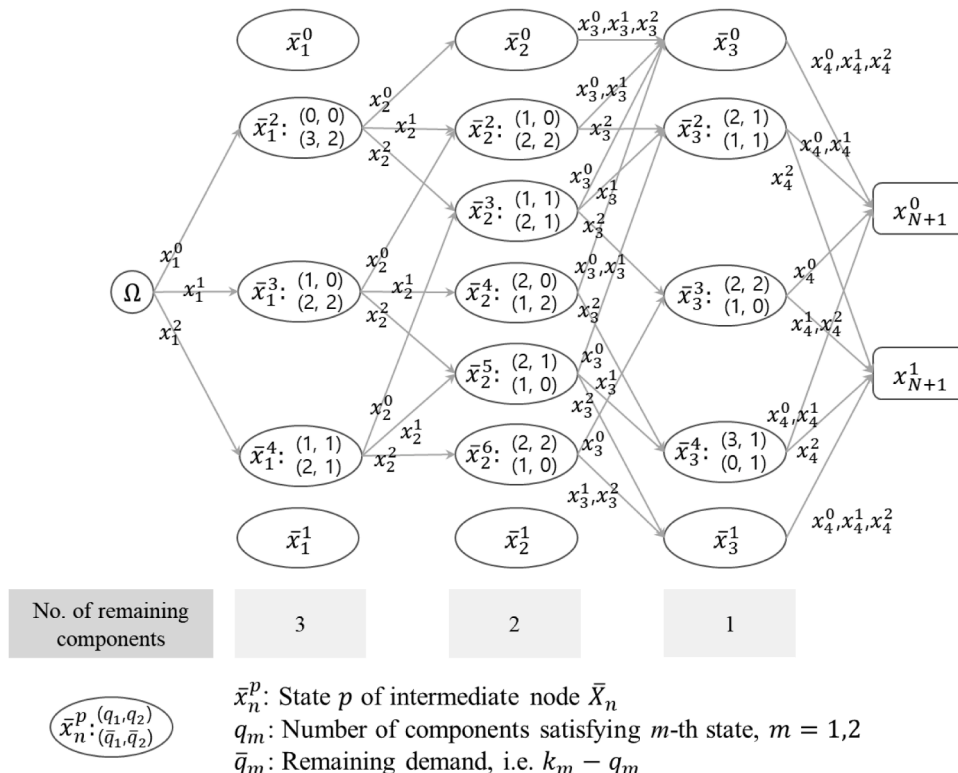


Fig. 7. DD procedure for example kN-MS:G system in Section 4.3.2

**Algorithm 5**

MBN quantification of a MS-kN:G system.

---

**Procedure** MBN-quant-MSkN (

$k$  // target demand

)

- 1  $\mathcal{M}_{\bar{x}_0}, BS(\bar{X}_0), \bar{Q}^0 \leftarrow \text{Initialize-CPM-MSkN}(k)$
- 2 **for**  $n = 1, \dots, N$
- 3      $\mathcal{M}_{\bar{x}_n}, BS(\bar{X}_n), \bar{Q}^n$
- 4      $\leftarrow \text{MBN-Quant-MSkN-Intermediate}(\mathcal{M}_{\bar{x}_{n-1}}, \bar{Q}^{n-1}, k)$
- 5 **return**  $\mathcal{M}_{X_{N+1}}, \{\mathcal{M}_{\bar{x}_n}, BS(\bar{X}_n)\}_{n=1}^{N-1}$

**Procedure** Initialize-CPM-MSkN(

$k$  // target demand

)

- 1 Let  $M$  be the length of  $k$
- 2 Set  $BS(\bar{X}_0)$  and  $\bar{Q}^0$  to include a null state  $\bar{x}_0$  and the associated vector  $\bar{q}^{\bar{x}_0} = k$ , respectively
- 3  $\mathcal{M}_{\bar{x}_0} \leftarrow \{(2); 1\}$
- 4 **return**  $\mathcal{M}_{\bar{x}_0}, BS(\bar{X}_0), \bar{Q}^0$

**Procedure** MBN-Quant-MSkN-Intermediate(

$\bar{x}_{n-1}$  //  $(n - 1)$ -th intermediate node

$\mathcal{M}_{\bar{x}_{n-1}}$  // CPM of  $\bar{x}_{n-1}$

$\bar{Q}^{n-1}$  // set of  $\bar{q}^{\bar{x}_{n-1}}$ , for  $\bar{x}_{n-1} \in BS(\bar{X}_{n-1})$

$k$  // target demand

$X_n$  // r.v. of  $n$ -th component event

)

- 1 Let  $M$  be the length of  $k$
  - 2  $\mathcal{M}_{\bar{x}_n} \leftarrow \emptyset, BS(\bar{X}_n) \leftarrow \emptyset, \bar{Q}^n \leftarrow \emptyset$
  - 3 **for each**  $k \in BS(\bar{X}_{n-1})$
  - 4     **if**  $k = 0$  or  $k = 1$  **then**
  - 5          $BS(\bar{X}_n) \leftarrow BS(\bar{X}_n) \cup \{k\}$
  - 6         Let  $x_n \in Val(X_n)$  be a state such that  $B_{X_n}(x_n) = \{0, 1, \dots, M\}$
  - 7          $\mathcal{M}_{\bar{x}_n} \leftarrow \mathcal{M}_{\bar{x}_n} \cup \{(k, x_n, k); 1\}$
  - 8     **else**
  - 9         **if**  $\bar{q}^k$  have all elements equal to or smaller than 1 **then**
  - 10             Let  $s$  be the largest index of elements that are equal to 1
  - 11             Let  $x_n \in Val(X_n)$  be a state such that  $B_{X_n}(x_n) = \{s, s + 1, \dots, M\}$
  - 12              $\mathcal{M}_{\bar{x}_n} \leftarrow \mathcal{M}_{\bar{x}_n} \cup \{(1, x_n, k); 1\}$
  - 13         **else**
  - 14              $s \leftarrow M + 1$
  - 15             **if**  $\bar{q}^k$  has some elements equal to  $N - (n - 1)$  **then**
  - 16                 Let  $t$  be the largest index among such elements
  - 17                 Let  $x_n \in Val(X_n)$  be a state such that  $B_{X_n}(x_n) = \{0, 1, \dots, t - 1\}$
  - 18                  $\mathcal{M}_{\bar{x}_n} \leftarrow \mathcal{M}_{\bar{x}_n} \cup \{(0, x_n, k); 1\}$
  - 19             **else**
  - 20                  $t \leftarrow 0$
  - 21             **for**  $x_n = t, \dots, s - 1$
  - 22                 Update  $BS(\bar{X}_n)$  and  $\bar{Q}^n$  to include a state  $\bar{x}_n$  with  $\bar{q}^{\bar{x}_n} = \bar{q}^k - \mathbf{1}_{M, x_n}$
  - 23                  $\mathcal{M}_{\bar{x}_n} \leftarrow \mathcal{M}_{\bar{x}_n} \cup \{(\bar{x}_n, x_n, k); 1\}$
  - 24 **return**  $\mathcal{M}_{\bar{x}_n}, BS(\bar{X}_n), \bar{Q}^n$
-

1)-th, ...,  $M$ -th elements are 0.

**5. Numerical examples: probabilistic inference of multi-state systems by MBN**

In the following examples, the MBN has been inferred using standard BN algorithms such as conditioning, variable elimination, and junction tree algorithm [11,4]. While the MBN quantification is illustrated in detail in the following subsections, the reader who is interested in the MBN inference is referred to the shared source code (which can be downloaded at <https://github.com/jieunbyun/Generalized-MBN-multi-state>) and the preceding study [5].

**5.1. Four-subsystem power system**

Consider a power system that can be described as a series system of different subsystems (modules) where the subsystem capacities are equivalent to the sum of the surviving components [2], i.e. the system is an MS-SP system described in Section 4.1.2. There are four subsystems, i.e.  $N = 4$ , each of which consists of 5, 7, 10, and 3 components. In the  $n$ -th subsystem,  $n = 1, \dots, N$ , the components have the identical reliability  $r_n$  and the capacity value  $q_n$ ; in addition, they are subject to a common cause failure with the probability  $p_{f,n}$ , which may arise from environmental loads, maintenance errors, and design flaws [12]. The assumed values of parameters are summarized in Table 3.

The graphical structure of BN is illustrated in Fig. 8 where the box represents the individual subsystems. In the box, the r.v.  $C_n$ ,  $n = 1, \dots, N$ , represents the number of surviving components, whereby the event is equivalent to a (binary-state)  $k$ -out-of- $N$ :G system [13]. Accordingly,  $P(C_n)$  can be computed by the binomial distribution, producing the rules  $\langle (c_n); p_n \rangle$  for the CPM  $\mathcal{M}_{C_n}$  as

$$p_n = \binom{M_n}{c_n} \cdot r_n^{c_n} (1 - r_n)^{M_n - c_n} \quad \text{for } c_n = 0, 1, \dots, M_n \quad (11)$$

On the other hand,  $F_n$  describes whether in the  $n$ -th subsystem, the common cause failure takes place ( $f_n^1$ ) or not ( $f_n^0$ ). The CPM  $\mathcal{M}_{F_n}$  can be quantified using the parameters  $p_{f,n}$  in Table 3.

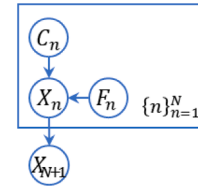
Then, as illustrated in Section 4.1.2, the r.v.  $X_n \in \mathbf{X}_N$  represents the capacity of the  $n$ -th subsystem, which, in this example, has  $F_n$  and  $C_n$  as parent nodes. The CPM  $\mathcal{M}_{X_n}$  for  $P(X_n|F_n, C_n)$  can be quantified based on the setting such that given  $f_n^1$ , the capacity is zero for all states of  $C_n$  while for  $f_n^0$ , the capacity is  $c_n \cdot q_n$  for each  $c_n \in BS(C_n)$ . It is noted that such deterministic relationship leads  $\mathbf{p}_{X_n}$  to have all elements being one. For instance, the CPM  $\mathcal{M}_{X_1}$  is quantified as

$$\mathbf{C}_{X_1} = \begin{bmatrix} 0 & 1 & 6 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \\ 2 & 0 & 2 \\ 3 & 0 & 3 \\ 4 & 0 & 4 \\ 5 & 0 & 5 \end{bmatrix} \quad \text{and} \quad \mathbf{p}_{X_1} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \quad (12)$$

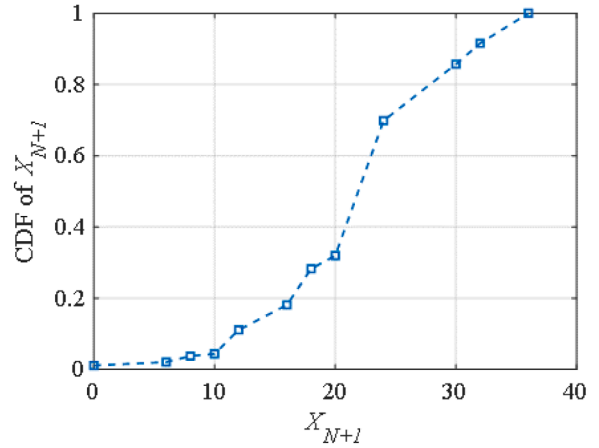
where the columns of  $\mathbf{C}_{X_1}$  stand for  $X_1$ ,  $F_1$ , and  $C_1$  from the left to right; the state  $c_1 \in BS(C_1)$  denotes that there are  $c_n$  surviving components,

**Table 3**  
Parameters of example MS-SP power system

$n$	1	2	3	4
Number of components, $M_n$	5	7	10	3
Component reliability, $r_n$	0.8	0.6	0.5	0.9
Component capacity, $q_n$	10	8	6	12
Probability of common cause failure, $p_{f,n} (\times 10^{-3})$	1	3	2	1



**Fig. 8.** BN for example power system



**Fig. 9.** CDF of  $P(X_{N+1})$  in example power system

**Table 4**  
Parameters and results of sensitivity analysis of example power system

$n$	1	2	3	4
Variation by unit cost, $\Delta r_n$	0.05	0.07	0.10	0.30
Derivative, $\partial P_f / \partial r_n (\times 10^{-1})$	-3.79	-5.75	-7.49	-2.48
Upgrade worth, $I_{p,r_n} (\times 10^{-2})$	1.89	4.02	7.49	0.743

while the composite state  $c_1^5$  is introduced as  $B_{C_1}(6) = \{0, 1, \dots, 5\}$ ; and the states  $x_1^0, x_1^1, \dots, x_1^5$  represent the subsystem capacities of 0, 10, ..., 50, respectively. Finally, the distribution of the system event,  $P(X_{N+1}|\mathbf{X}_N)$  can be quantified by Algorithm 3. It is noted that to quantify the distribution, the naïve approach would quantify  $\prod_{n=1}^N (M_n + 1) = 2,112$  instances, while the MBN requires only 20 rules for  $\mathcal{M}_{X_{N+1}}$ .

Using the CPMs quantified over the BN in Fig. 8, the marginal probability of the system event,  $P(X_{N+1})$  has been computed using variable elimination algorithm. Thereby, the cumulative distribution function (CDF) of  $P(X_{N+1})$  can be computed as shown in Fig. 9. By defining the system failure event as the system capacity being smaller than 10, the system failure probability is computed as  $3.74 \times 10^{-2}$  from the result in the figure.

The importance of components can be also measured by calculating parameter sensitivity of probability using the quantified CPMs. First, the sensitivity of the system failure probability with respect to the component reliability  $r_n$ ,  $n = 1, \dots, N$ , is computed using the proposed method. Since the values of  $r_n$  determine the probabilities of  $P(C_n)$  (see Eq. (11)), the CPM  $\mathcal{M}_{C_n}$  is replaced by  $\mathcal{M}_{C_n, r_n}$  whose rules  $\langle c_n; \partial p_n / \partial r_n \rangle$  are derived as

$$\frac{\partial p_n}{\partial r_n} = \binom{M_n}{c_n} \cdot (c_n \cdot r_n^{c_n-1} (1-r_n)^{M_n-c_n} + (M_n-c_n) \cdot r_n^{c_n} (1-r_n)^{M_n-c_n-1}) = \binom{M_n}{c_n} \cdot r_n^{c_n-1} (1-r_n)^{M_n-c_n-1} (c_n(1-r_n) + (M_n-c_n)r_n), \quad (13)$$

$c_n = 0, 1, \dots, M_n$

Next, based on the parameter sensitivity, the ‘‘upgrade worth’’ is computed to quantify the importance of components [9,14]. This measure quantifies the worth of fixed cost upgrade of the corresponding component by

$$I_\theta = - \left[ \frac{\partial P_f}{\partial \theta} \right] \Delta \theta \quad (14)$$

where  $\theta (= r_n$  in this example),  $\Delta \theta$ , and  $P_f$  respectively denote the parameter of interest, the variation in  $\theta$  that can be achieved by a unit cost increment, and the system failure probability. The assumed values of  $\Delta r_n$  are summarized in Table 4. Thereby, the values of  $\partial P_f / \partial r_n$  and  $I_{P_f, r_n}$  are computed as shown in Table 4. The result suggests that the greatest worth is expected for upgrading the component reliability of the third subsystem.

### 5.2. Sioux Falls benchmark transportation network

This section investigates the Sioux Falls benchmark transportation network consisting of 38 arcs (roads) and 24 nodes as illustrated in Fig. 10, assumed subject to seismic hazard [15]. In this example, the system event  $X_{N+1}$  survives if the target flow  $d = 4$  can be delivered from the nodes  $s$  to  $t$  that are marked in the figure. As illustrated in Section 4.2.2, the flow capacities of the arcs are represented by the component events  $X_n \in \mathbf{X}_N$ , while  $X_n$  can take three states  $x_n^1, x_n^2$ , and  $x_n^3$  that respectively correspond to the flow capacity of 0, 2, and 3.

The BN in Fig. 11 is constructed to represent the system where the seismic hazard is represented by the r.v.’s  $M$  and  $L$  that respectively stand for the moment magnitude and the epicenter location. On the other hand, the box represents the individual arcs, in which  $D_n, I_n$ , and  $X_n$  respectively represent the deterioration, the inspection result, and the flow capacity of the  $n$ -th arc,  $n = 1, \dots, N$ . Specifically,  $D_n$  takes a binary state of being either deteriorated ( $d_n^1$ ) or not ( $d_n^0$ ). Similarly, the binary state of  $I_n$  denotes the inspection result of being deteriorated ( $i_n^1$ ) or not ( $i_n^0$ ). This setting leads  $X_n$  to have parent nodes of  $M, L$ , and  $D_n$ , while  $D_n$

is also the parent node of  $I_n$ .

For quantification, the magnitude  $M$  is assumed to follow the truncated exponential distribution [16] with the probability density function (PDF)

$$f_M(m) = \begin{cases} \frac{\beta \exp[-\beta(m_p - m_0)]}{1 - \exp[-\beta(m_p - m_0)]}, & \text{for } m_0 \leq m \leq m_p \\ 0 & \text{otherwise} \end{cases} \quad (15)$$

where  $\beta = 0.76, m_0 = 6.0$ , and  $m_p = 8.5$ . To quantify the CPM  $\mathcal{M}_M$ , the values of  $M$  are uniformly discretized into 5 intervals such that [6.0, 6.5), [6.5, 7.0), [7.0, 7.5), [7.5, 8.0), and [8.0, 8.5] whose representative values are set as the midpoint values during subsequent analysis. On the other hand, the location  $L$  is set to follow the uniform distribution over the five locations  $(-2, -2), (-1, -3), (0, -4), (1, -5)$ , and  $(2, -6)$  km. All arcs are assumed to have the probability of deterioration as 0.05, i.e.  $P(d_n^1) = 0.05$ , while the inspection has the rates of false positive and false negative as 0.05 and 0.2, respectively, i.e.  $P(i_n^1 | d_n^0) = 0.05$  and  $P(i_n^0 | d_n^1) = 0.2$ .

To quantify the distribution of  $X_n$ , the intensity measure (IM) of seismic hazards is defined as the peak ground accelerations (PGA) experienced at the centroids of the roads, which can be computed as

$$\ln(PGA) = \mu_{\ln PGA} + \varepsilon$$

where  $\mu_{\ln PGA} = -3.512 + 0.904m - 1.328 \ln \sqrt{r^2 + [0.149 \exp(0.647m)]^2} + [0.440 - 0.171 \ln r] S_{SR} + [0.405 - 0.222 \ln r] S_{HR} + [1.125 - 0.112 \ln r - 0.0957m] F$

(16)

In the equation, PGA has the unit  $g$ ;  $\mu_{\ln PGA}$  and  $\varepsilon$  are respectively the mean value and the error term of the logarithm of the PGA;  $m$  and  $r$  respectively denote the magnitude and the distance between the epicenter and the centroid of the road; and the parameters  $S_{SR}, S_{HR}$ , and  $F$  are set to zero by assuming firm soil and strike-slip type faulting [17]. The error term  $\varepsilon$  follows the normal distribution with zero mean and the standard deviation  $\sigma$  given as

$$\sigma = \begin{cases} 0.55, & A_H < 0.068g \\ 0.173 - 0.140 \ln(A_H), & 0.068g \leq A_H \leq 0.21g \\ 0.39, & A_H > 0.21g \end{cases} \quad (17)$$

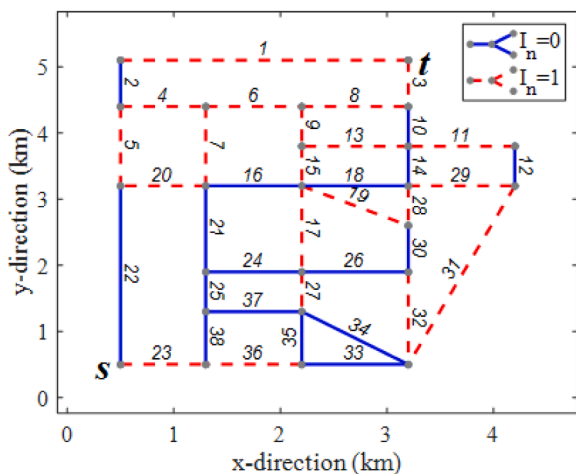


Fig. 10. Topology of Sioux Falls benchmark network and hypothetical inspection scenario

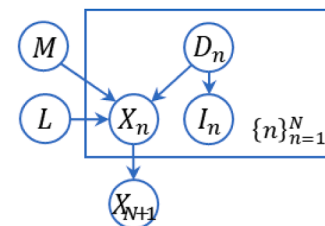


Fig. 11. BN for Sioux Falls benchmark network under seismic hazards

Table 5  
PGA capacity (g) of roads to maintain the flow capacity

Flow capacity	2 ( $c_2$ )	3 ( $c_3$ )
$d_n^0$	1.20	1.00
$d_n^1$	1.00	0.85

**Table 6**  
System failure probability of Sioux Falls benchmark network

	Bounds by BnB [upper, lower]	99% confidence interval by MCS	Bounds by BnB given $I_N$
System failure probability ( $\times 10^{-2}$ )	[2.69, 2.70]	[2.58, 3.01]	[3.89, 3.93]

On the other hand, the seismic capacity of the roads described in terms of PGA is considered deterministic. Accordingly, the probabilities of  $X_n \in \mathbf{X}_N$  are derived as

$$\begin{aligned}
 P(x_n^1) &= \Phi\left(-\frac{c_2 - \mu_{\ln PGA}}{\sigma}\right), \\
 P(x_n^2) &= \Phi\left(\frac{c_2 - \mu_{\ln PGA}}{\sigma}\right) - \Phi\left(\frac{c_3 - \mu_{\ln PGA}}{\sigma}\right), \\
 \text{and } P(x_n^3) &= \Phi\left(\frac{c_3 - \mu_{\ln PGA}}{\sigma}\right)
 \end{aligned} \tag{18}$$

where  $\mu_{\ln PGA}$  and  $\sigma$  can be evaluated respectively from Eqs. (16) and (17). The parameters  $c_2$  and  $c_3$  are the threshold values of PGA capacities for the flow capacity being 2 and 3, respectively, being dependent on  $D_n$  as illustrated in Table 5.

Finally, the CPM  $\mathcal{M}_{X_{N+1}}$  for the system event  $P(X_{N+1}|\mathbf{X}_N)$  can be quantified by Algorithm 4. It is noted that the algorithm requires the joint PMF  $P(\mathbf{X}_N)$  as an input, whose exact computation is expensive. Therefore, an alternative PMF  $Q(\mathbf{X}_N) = \prod_{n=1}^N P(X_n|m^5)$  is used where the state  $m^5$  corresponds to the last interval  $m \in [8.0, 8.5]$ . Such conditioning is found to be effective in most numerical experiments, having an effect analogous to the importance sampling (IS). Since the event space is too large for exhaustive quantification, the algorithm has been terminated when the probability of the unspecified sets becomes less than  $10^{-3}$ , i.e.  $\bar{p} = 1 \times 10^{-3}$  in Algorithm 4. As a result, the subsequent inference does not provide the exact results, but deterministic bounds.

For inference using the BN, the r.v.'s  $M$  and  $L$  are marginalized by conditioning, while other variables are marginalized by sum-product variable elimination algorithm (for details of conditioning, readers are referred to Chapter 9.5 of Koller and Friedman [4]). By performing the inference using the non-exhaustive CPM  $\mathcal{M}_{X_{N+1}}$ , the bounds on the system failure probability  $P(x_{N+1}^0)$  can be computed as

$$\tilde{P}(x_{N+1}^0) \leq P(x_{N+1}^0) \leq 1 - \tilde{P}(x_{N+1}^1) \tag{19}$$

where  $\tilde{P}(\cdot)$  denotes the probability whose computation involves any non-exhaustive CPMs. The inference result is summarized in Table 6 along with that by Monte Carlo Simulation (MCS). While the two results agree with each other, it is noted that the BnB-based decomposition quantifies  $\mathcal{M}_{X_{N+1}}$  with 1,007 rules (606 survival sets and 401 failure sets) while the MCS results in 38,670 rules in the CPM to attain the coefficient of variance (c.o.v.) of 0.03. This demonstrates that analytical decomposition methods are in general more efficient than sampling methods if there is an available method for a given system.

The BN in Fig. 11 can also be used to evaluate the relative importance of the components (roads). For instance, one can employ the conditional probability based importance measure (CPIM) proposed by Song and Kang [9], where the measure is defined as the component failure probability conditioned on the system failure. Since a component can have more than two states, the component failure event is regarded as the arc having capacity less than the greatest capacity it can have, i.e.  $x_n^4$  where  $B_{X_n}(4) = \{1, 2\}$ . To compute the CPIM, the bounds of the joint probability  $P(x_n^4, x_{N+1}^0)$ ,  $n = 1, \dots, N$ , are computed as

$$\tilde{P}(x_n^4, x_{N+1}^0) \leq P(x_n^4, x_{N+1}^0) \leq 1 - \tilde{P}(\{x_n^4, x_{N+1}^0\}^c) \tag{20}$$

where  $\{x_n^4, x_{N+1}^0\}^c$  is the set of states such that  $Val(X_n, X_{N+1}) - \{x_n^4, x_{N+1}^0\}$ . Accordingly, the bounds on the CPIMs of  $X_n$ , i.e.  $P(x_n^4|x_{N+1}^0)$  are

computed from Eqs. (19) and (20) as

$$\frac{\tilde{P}(x_n^4, x_{N+1}^0)}{1 - \tilde{P}(x_{N+1}^1)} \leq P(x_n^4|x_{N+1}^0) \leq \frac{1 - \tilde{P}(\{x_n^4, x_{N+1}^0\}^c)}{\tilde{P}(x_{N+1}^0)} \tag{21}$$

In Fig. 12, the deterministic bounds of the five largest CPIMs are illustrated along with the 99% confidence intervals evaluated using the MCS samples. While the results by the two approaches agree with each other, it is noted that the deterministic decomposition provides much narrower intervals. On the other hand, the list of the five roads with the highest CPIMs is as expected for they constitute the major routes from  $s$  to  $t$ . The CPIMs of other roads do not show a notable difference, all having the upper bounds less than 0.05.

Another advantage of BN is that it is straightforward to update the probability distribution to reflect new information. To demonstrate this, a hypothetical inspection scenario is assumed as illustrated in Fig. 10. The blue solid and red dotted lines respectively indicate that the roads are inspected to be intact ( $i_n^0$ ) and deteriorated ( $i_n^1$ ). The scenario has been generated with the probability 0.5 for both assignments, which puts a higher probability of deterioration than the originally given  $P(D_n)$ . Consequently, the system failure probability rises by around 45% as illustrated in Table 6. The CPIMs can be also updated as illustrated in Fig. 12 which suggests that the inspection result of being deteriorated increases the CPIM. The stable bounds widths of the updated information suggest that approximate inference by deterministic decomposition is in general more robust against varying configurations compared to sampling approaches, which often shows unstable performance in such conditions.

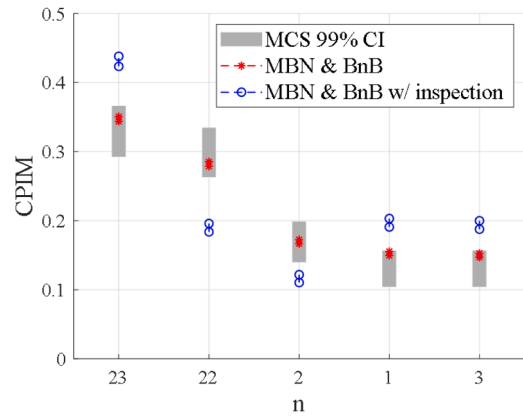


Fig. 12. Top five CPIMs of roads

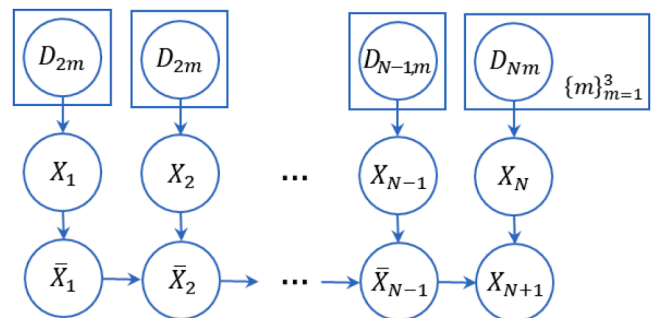


Fig. 13. BN for example oil distribution system

### 5.3. Three-station oil distribution system

Consider the oil distribution system in Fig. 6 with  $N = 20$  and  $M = 3$ , i.e. 20 pipelines and 3 stations. In addition to the component events and the system event, the r.v.'s  $D_{nm}$ ,  $n = 1, \dots, N$  and  $m = 1, 2, 3$ , are introduced to account for the deterioration of the  $m$ -th segment of the  $n$ -th pipeline where the  $m$ -th segment is the part that connects the  $m$ -th station from the previous one. This leads to the BN in Fig. 13 where the boxes represent the individual segments. As indicated by the BN, the component event  $X_n$  depends on  $D_{nm}$ ,  $m = 1, 2, 3$ , while the system event  $X_{N+1}$  and the intermediate nodes  $\bar{X}_1, \dots, \bar{X}_{N-1}$ , can be quantified by Algorithm 4.

The r.v.  $D_{nm}$  takes a binary state for being deteriorated ( $d_{nm}^1$ ) and being intact ( $d_{nm}^0$ ), respectively with the probabilities of 0.2 and 0.8. The state of  $D_{nm}$  affects the failure likelihood of the corresponding segment as given  $d_{nm}^1$ , the segment fails with probability 0.3 while given  $d_{nm}^0$ , it fails with probability 0.1. These parameters can be used to quantify the PMF  $P(X_n | D_{n3}, D_{n2}, D_{n1})$  (recall from Section 4.2.2 that the r.v.  $X_n$  represents the farthest station that the  $n$ -th pipeline can reach).

To evaluate the system reliability, three target demands are considered:  $\mathbf{k}^1 = (10, 7, 4)$ ,  $\mathbf{k}^2 = (5, 11, 5)$ , and  $\mathbf{k}^3 = (4, 7, 10)$ . The system failure probability  $P(X_{N+1}^0)$  has been computed by the junction tree algorithm, which is obtained as  $7.36 \times 10^{-5}$ ,  $1.85 \times 10^{-2}$ , and  $6.94 \times 10^{-2}$  respectively for  $\mathbf{k}^1$ ,  $\mathbf{k}^2$ , and  $\mathbf{k}^3$ . As expected, the more demand the farther station requires, the larger the failure probability becomes. It is noted that while the event  $X_N \cup \{X_{N+1}\}$  consists of  $(M+1)^N \approx 2.20 \times 10^{12}$  instances, the numbers of rules used to quantify the CPMs  $\mathcal{M}_{X_{N+1}}$  and  $\mathcal{M}_{\bar{X}_1}, \dots, \mathcal{M}_{\bar{X}_{N-1}}$  are 15,662, 14,250, and 10,140 respectively for  $\mathbf{k}^1$ ,  $\mathbf{k}^2$ , and  $\mathbf{k}^3$ .

## 6. Conclusions

The matrix-based Bayesian network (MBN [5]), an alternative data structure of discrete Bayesian network (BN) recently proposed to handle large-scale systems, was generalized in this paper to handle multi-state systems representing a wide range of real-world systems. To achieve this goal, the concept of *composite state* was proposed to represent a subset of states collectively, as a generalization of “-1” state introduced in Byun et al. [5].

The definitions of the MBN and the basic probability operations (conditioning, sum, and product) were renewed so that the composite state can be consistent with the existing BN inference algorithms. In addition, an MBN-based formulation was proposed to compute parameter sensitivity of probabilities of interest. The paper also presents concrete illustrations of MBN applications. To this end, the MBN was quantified using three commonly used techniques that deterministically decompose event space, i.e. utilization of verbal definition of events, branch and bound (BnB), and decision diagram (DD), each being accompanied by an example system, i.e. multi-state series parallel (MS-SP) system, flow network, and multi-state  $k$ -out-of- $N$ :G (MS-kN:G) system, respectively. Furthermore, various inference tasks were demonstrated in the numerical examples, which includes evaluating system failure probability and component importance measure, sensitivity analysis, and updating distributions. The approximate inference using the MBN was also explored, which suggests that the approximation based on deterministic quantification tends to show better performance in terms of efficiency and robustness, given that there exists an available method for a given type of system. The supporting source code of the MBN application and the data of the numerical examples can be download at <https://github.com/jieunbyun/Generalized-MBN-multi-state>.

The MBN quantification can be further developed to advance the system reliability analysis based on BN. To this end, MBN quantification methods can be developed for the systems whose efficient quantification

has not been addressed yet. Such framework can be developed even for general systems so that any arbitrary systems can be efficiently quantified. Another type of systems worth investigating is probabilistic system events where a combination of component states may lead to several system states with certain probabilities. Moreover, the MBN can be extended for dynamic systems, in which the presence of time variables can make the quantification and inference more challenging.

## CRedit authorship contribution statement

**Ji-Eun Byun:** Conceptualization, Methodology, Software, Validation, Writing - original draft, Visualization. **Junho Song:** Conceptualization, Resources, Writing - review & editing, Supervision, Project administration, Funding acquisition.

## Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgement

This work was supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant 21SCIP-B146946-04). The second author is supported by the Institute of Construction and Environmental Engineering at Seoul National University.

## References

- [1] Jane CC, Lai YW. A practical algorithm for computing multi-state two-terminal reliability. *IEEE Transactions on Reliability* 2008;57(2):295–302.
- [2] Levitin G, Lisnianski A, Ben-Haim H, Elmakis D. Redundancy optimization for series-parallel multi-state systems. *IEEE Transactions on Reliability* 1998;47(2):165–72.
- [3] Mo Y, Xing L, Amari SV, Bechta Dugan J. Efficient analysis of multi-state k-out-of-n systems. *Reliability Engineering and System Safety* 2015;133:95–105.
- [4] Koller D, Friedman N. *Probabilistic Graphical Models: Principles and Techniques*. Foundations. Cambridge, MA: MIT Press; 2009.
- [5] Byun JE, Zwirgmaier K, Straub D, Song J. Matrix-based Bayesian Network for efficient memory storage and flexible inference. *Reliability Engineering and System Safety* 2019;185:533–45.
- [6] Byun J-E, Song J. Efficient Probabilistic Multi-Objective Optimization of Complex Systems Using Matrix-based Bayesian Network. *Reliability Engineering and System Safety* 2020;200:106899.
- [7] Xu Z, Ramirez-Marquez JE, Liu Y, Xiahou T. A new resilience-based component importance measure for multi-state networks. *Reliability Engineering and System Safety* 2020;193:106591.
- [8] Huang J, Zuo MJ. Multi-state  $k$ -out-of- $n$  system model and its applications. In: *Proceedings of the Annual Reliability and Maintainability Symposium*. IEEE; 2000. p. 264–8.
- [9] Song J, Kang WH. System reliability and sensitivity under statistical dependence by matrix-based system reliability method. *Structural Safety* 2009;31(2):148–56.
- [10] Kam T, Villa T, Brayton R, Sangiovanni-Vincentelli A. *Synthesis of Finite State Machines: Functional Optimization*. Berlin: Springer; 1997.
- [11] Barber D. *Bayesian reasoning and machine learning*. Cambridge: Cambridge University Press; 2012.
- [12] Li Cyang, Chen X, Yi Xshan, Tao Jyong. Heterogeneous redundancy optimization for multi-state series-parallel systems subject to common cause failures. *Reliability Engineering and System Safety* 2010;95(3):202–7.
- [13] Byun JE, Noh HM, Song J. Reliability growth analysis of  $k$ -out-of- $N$  systems using matrix-based system reliability method. *Reliability Engineering and System Safety* 2017;165:410–21.
- [14] Der Kiureghian A, Ditlevsen OD, Song J. Availability, reliability and downtime of systems with repairable components. *Reliability Engineering and System Safety* 2007;92(2):231–42.
- [15] Lee Y-J, Song J, Gardoni P, Lim H-W. Post-hazard flow capacity of bridge transportation network considering structural deterioration of bridges. *Structure and Infrastructure Engineering* 2011;7(7–8):509–21.
- [16] Cosentino P, Ficarra V, Luzlo D. Truncated exponential frequency-magnitude relationship in earthquake statistics. *Bulletin of the Seismological Society of America* 1977;67(6):1623–5.
- [17] Campbell KW. Empirical near-source attenuation relationships for horizontal and vertical components of peak ground acceleration, peak ground velocity, and pseudo-absolute acceleration response spectra. *Seismological Research Letters* 1997;68(1):154–79.