Reliability Engineering and System Safety 185 (2019) 533–545

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Matrix-based Bayesian Network for efficient memory storage and flexible inference

Ji-Eun Byun[a], Kilian Zwirglmaier[b], Daniel Straub[b], Junho Song[a,*]

[a] Department of Civil and Environmental Engineering, Seoul National University, Seoul, South Korea
[b] Engineering Risk Analysis Group, Technische Universität München, München, Germany

ARTICLE INFO

Keywords:
Discrete Bayesian Network
Bayesian Network
Exact inference
Approximate inference
Complex systems
Matrix-based modeling

ABSTRACT

For real-world civil infrastructure systems that consist of a large number of functionally and statistically dependent components, such as transportation systems or water distribution networks, the Bayesian Network (BN) can be a powerful tool for probabilistic inference. In a BN, the statistical relationship between multiple random variables (r.v.'s) is modeled through a directed acyclic graph. The complexity of inference in the BN depends not only on the number of r.v.'s, but also the graphical structure. As a consequence, the application of standard BN techniques may become infeasible even with a moderate number of r.v.'s as the size of an event set exponentially increases with the number of r.v.'s. Moreover, when the exhaustive set that is required for full quantification of a discrete BN node becomes intractably large, only approximate inference algorithms are feasible, which do not require the full (explicit) description of all BN nodes. We address both issues in discrete BNs by proposing a matrix-based Bayesian Network (MBN) that facilitates efficient modeling of joint probability mass functions and flexible inference. The MBN is developed for exact as well as approximate BN inference. The efficiency and applicability of the MBN are demonstrated by numerical examples. The supporting source code and data are available for download at https://github.com/jieunbyun/GitHub-MBN-code.

1. Introduction

The disaster-resilience of urban communities relies heavily on the post-disaster performance of urban infrastructure systems [1]. Decision-making to manage the disaster-resilience of such systems requires probabilistic modeling and analysis because of significant uncertainties in the system and component performances encountered in the aftermath of a disaster. However, real-world civil systems such as transportation systems or water distribution networks make probabilistic modeling and analysis challenging because of a large number of components and their statistical dependence.

Bayesian Networks (BNs) have been proposed as a tool for probabilistic modeling and reliability analysis of infrastructure systems, in particular for post-disaster decision support [2,3]. BNs can provide an intuitive way for model representation, by using a directed acyclic graph (DAG) in which random variables (r.v.'s) are represented through nodes and their statistical dependences through links. In addition, BNs allow for efficient Bayesian updating when observations become available on parts of a system, e.g. detection of damages. The complexity of BN modeling and inference depends on the number of r.v.'s and the graphical structure, which limits the BN's applicability to real-world problems involving a large number of r.v.'s and complex interdependencies [4].

In order to make modeling and inference of BNs less sensitive to graphical structures, various methodologies have been proposed to exploit the regularity in the state of a r.v. that is a function of other r.v.'s, which is often encountered in problems with a large number of r.v.'s [5–8]. Although these methodologies may provide efficient alternatives, they still have some limitations in that the modeling methodologies have been developed for a limited class of problems, and the specialized formulations may hamper their applications to various types of BN inference. Another difficulty arising from the large number of r.v.'s is that the information may not be available to fully quantify the entire event space. In this case, these alternative methodologies are inapplicable as their formulations are valid only when the exhaustive set of an event is known. To address these two issues and extend the applicability of BNs, this paper proposes the matrix-based Bayesian Network (MBN), in which the conditional probability *matrices* (CPMs) are introduced to model discrete BNs and perform probabilistic inference. The development of MBN can be regarded as the unification of

* Corresponding author.
E-mail address: junhosong@snu.ac.kr (J. Song).

https://doi.org/10.1016/j.ress.2019.01.007
Received 12 September 2018; Received in revised form 7 January 2019; Accepted 10 January 2019
Available online 12 January 2019

the previous efforts that have been proposed to address these issues separately into one methodology which is more general and compatible with standard BN techniques.

The rest of the paper is organized as follows. Section 2 provides a brief introduction of the BN methodology in general and raises central issues addressed in this paper, along with the overview of the related works. Section 3 then develops definitions for the proposed CPM concept, and illustrates its efficient procedure of BN construction. New inference algorithms are developed for the proposed MBN in Section 4, and the applicability and efficiency of the MBN are demonstrated by numerical examples with various sizes in Section 5. Finally, Section 6 summarizes the paper with concluding remarks.

In this paper, an *assignment* refers to the specific values assigned to a set of r.v.'s corresponding to one of the possible outcomes. A r.v. and its assignment are respectively denoted by upper and lower cases, e.g. X and x, which are set in bold to denote a set of r.v.'s. For simplicity, x^k and $P(x^k)$ respectively denote the assignment of value k to the r.v. X and the probability of the assignment $X = k$, i.e. $P(X = k)$. The set of values that a r.v. X can take is denoted $Val(X)$. It is also noted that as this paper mainly considers discrete BNs, the notation $P(\cdot)$ refers to a probabilistic mass function (PMF), not a probabilistic density function (PDF). There is an exception in Section 2.1 where the general BN methodology is discussed, and $P(\cdot)$ applies to both PMFs and PDFs.

2. Background and related work

2.1. Bayesian Network

BNs are based on DAGs in which the r.v.'s are represented by nodes while their statistical dependence is depicted by directed arcs connecting the nodes, as illustrated in Fig. 1. The nodes from which the arcs are initiated and to which they are pointed, are respectively termed as the *parent* nodes (X_1 and X_2) and the *child* nodes (X_3). In the following discussions, the term *node* is used synonymously with *r.v.*.

In a BN, the joint distribution of the r.v.'s is formulated as the product of the conditional distributions given the parent nodes. If there are no parents for a node, its marginal distribution is used instead. For instance, the distributions quantifying the BN of Fig. 1, are $P(X_1)$, $P(X_2)$, and $P(X_3|X_1, X_2)$, and the joint distribution $P(X_1, X_2, X_3)$ is the product of these three distributions. In general, the joint distribution represented by a BN over a set of r.v.'s $\boldsymbol{X}$ is

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

where Pa_X is the set of parent nodes of X.

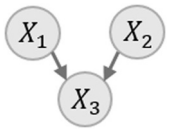


Fig. 1. Example BN.

2.2. Conditional probability tables (CPTs)

In discrete BNs, conditional probability tables (CPTs) are used as data structures for storing the conditional probability mass functions (PMFs) quantifying the nodes of a BN. CPTs store only the probability values while the corresponding assignments on the r.v.'s are inferred from their locations in the table. For example, consider the example BN in Fig. 1 and suppose that $Val(X_i) = \{0, 1\}$ for $i = 1, 2, 3$. Then, the CPT for the conditional PMF $P(X_3|X_1, X_2)$ can be constructed either as a third-order tensor [9]

$$p_{ijk} = P(x_3^i|x_1^j, x_2^k) \text{ for } i, j, k = 1, 2 \tag{2}$$

or arrays. A possible two-dimensional array [7] of p_{ijk} in Eq. (2) is

$$\begin{bmatrix} p_{111} & p_{112} & p_{121} & p_{122} \\ p_{211} & p_{212} & p_{221} & p_{222} \end{bmatrix} \tag{3}$$

while the one-dimensional array [10] is

$$\begin{bmatrix} p_{111} \\ p_{211} \\ p_{112} \\ p_{212} \\ p_{121} \\ p_{221} \\ p_{122} \\ p_{222} \end{bmatrix} \tag{4}$$

The straightforward construction and implementation make CPTs particularly favorable in developing general-purpose software programs of BN modeling. However, as indicated in Eqs. (2)–(4), the CPT formulations only implicitly assign the probability values to each outcome of the r.v. by their location in the tables. Such characteristic requires that the probabilities of all possible assignments be explicitly enumerated, which can be infeasible when a node has a large number of parents as the number of the assignments exponentially increases with the number of parents.

2.3. Converging structure in Bayesian Network and related research efforts

The table-based representation using CPTs is inefficient when a BN has a *converging* structure, i.e. a node X_S has a large number of parent nodes, which is frequently encountered when modeling a system whose state is determined by the joint state of its components. This is due to the exponential increase in the number of possible combinatorial assignments of the parent nodes, raising the following two major issues.

The first issue is that the memory required to store the parameters of $P(X_S|Pa_{X_S})$ quickly becomes unaffordable even with a moderate number of parent nodes. To address this issue, various approaches have been proposed to exploit the regularity and redundancy in the description of an event, which commonly takes place under the presence of a large number of r.v.'s. For instance, it was proposed to modify the graphical structure of BNs to reduce the maximum number of parent nodes in system reliability problems such as network connectivity problems, and series and parallel systems [5,11,12]. This approach is advantageous in that CPTs and the existing BN methodologies using CPTs, are still applicable after the modification. However, the global structure has to be modified to a less intuitive one, and the strategies for structure modifications have been developed only for a limited class of problems.

Another approach to deal with the memory issue is to exploit the local structure, i.e. the regularities within a joint PMF $P(X_S, Pa_{X_S})$, for efficient storage of PMFs. The network polynomial [8] factors the parameters of each assignment, and for inference, establishes an arithmetic circuit to carry out the algebraic operation of these factorized parameters. Boutilier et al. [6] proposed a tree-based representation, which models a PMF as a tree-structured diagram. Alternatively, Tien and Der Kiureghian [10] employ data compression techniques to reduce the memory required for storing PMFs. In these methodologies, the complexity of the inference can be reduced if the supplementary models are cleverly designed in a way to exploit the local structure. However, there is no general rule to design supplementary models, which may hamper general implementations for various types of BN inference.

As an alternative to exploiting the local structure, a rule-based representation [7] aims to store the outcomes more efficiently by explicitly storing the associated assignments together with probability values. This type of representation makes the modeling straightforward with the given information and facilitates developing a general inference rule. However, the data structure is not conducive to an efficient implementation. In Section 3.4, the approaches discussed above

are compared by means of a numerical example.

The second issue is the infeasible computational cost to analyze all possible outcomes in the joint space of X_S and all its parents. In this case, a rule-based representation is not applicable as the method requires that all possible outcomes be expressed, i.e. the probabilities of the considered outcomes must sum up to unity. As an alternative, various approximate methods have been proposed to carry out inference only with a subset of the joint outcome space of X_S and its parents in the context of reliability analysis. Those outcomes can be selected either deterministically [13–16] or stochastically [17–19]. Despite the extensive application of these methods, there have been only a few efforts to systematically connect the approximate methods with probabilistic inference [20–22] and a formal linkage to BN methodologies is not available.

This paper aims to propose a new data structure of PMFs, which addresses the representation of the full $P(X_S|Pa_{X_S})$. Approximate inference algorithms are formulated based on the proposed PMF representations, to enable applications of existing approximate inference methodologies to BNs representing a wide class of problems.

3. Proposed data structure of PMF: conditional probability matrices (CPMs)

The proposed MBN uses conditional probability *matrices* (CPMs) as an alternative data structure of the PMFs in a discrete BN. These CPMs have the same structure as the matrices introduced in the matrix-based system reliability (MSR) method [23,24]. In the MSR method, event and probability vectors of matrix form are introduced jointly to store the assignment and probability of each outcome separately. The BN modeling and inference of the MBN can be regarded as an extension of the rule-based representation, where rules are used as the bases to construct the modeling structure of BNs.

3.1. Definitions for conditional probability matrices

First, the following new definitions of rules are introduced to build the concept of CPMs in MBN:

Definition 1–1. (*Rule in the MBN*)**:** A rule in the MBN, μ is a pair $\langle \boldsymbol{c}; p\rangle$ where $\boldsymbol{c}$ is a vector representing an assignment to a set of r.v.'s $\boldsymbol{X}$ and $p \in [0, 1]$ is the corresponding probability. Inversely, $\boldsymbol{X}$ is defined as the *scope* of μ, denoted by $Scope[\mu]$.

In addition, the MBN can feature a "-1" *state* as explained below. In the following, for a set of r.v.'s $\boldsymbol{Y}$ and the assignment $\boldsymbol{c}$ defined for another set of r.v.'s $\boldsymbol{X}$, $\boldsymbol{c}\langle\boldsymbol{Y}\rangle$ denotes the subset of the assignment in $\boldsymbol{c}$ corresponding to the intersection of the r.v. sets, i.e. $\boldsymbol{X}\cap\boldsymbol{Y}$.

Definition 1–2. (*Rule in the MBN*)**:** For a PMF $P(\boldsymbol{X})$ and an associated assignment $\boldsymbol{x}$ over $\boldsymbol{X}$, suppose a subset of r.v.'s $\boldsymbol{Y}\subseteq\boldsymbol{X}$ leads to

$$P(\boldsymbol{x}\langle\boldsymbol{X}-\boldsymbol{Y}\rangle, \boldsymbol{y}) = p, \quad \forall\, \boldsymbol{y} \in Val(\boldsymbol{Y}) \tag{5}$$

for some $p \in [0, 1]$, where $\boldsymbol{X}-\boldsymbol{Y} = \boldsymbol{X}\cap\boldsymbol{Y}^{\mathbf{c}}$ and $\boldsymbol{x}\langle\boldsymbol{X}-\boldsymbol{Y}\rangle$ denotes the part of $\boldsymbol{x}$ corresponding to $\boldsymbol{X}-\boldsymbol{Y}$ following the aforementioned definition of bracket $\langle\cdot\rangle$ with assignments, i.e. $P(\boldsymbol{x}\langle\boldsymbol{X}-\boldsymbol{Y}\rangle, \boldsymbol{y}) = P(\boldsymbol{X}\cap\boldsymbol{Y}^{\mathrm{c}} = \boldsymbol{x}\langle\boldsymbol{X}\cap\boldsymbol{Y}^{\mathrm{c}}\rangle, \boldsymbol{Y} = \boldsymbol{y})$. Then, the vector $\boldsymbol{c}$ in the rule $\mu = \langle\mathbf{c}; p\rangle$ is defined by the following two subsets:

$$\boldsymbol{c}\langle\boldsymbol{X}-\boldsymbol{Y}\rangle = \boldsymbol{x}\langle\boldsymbol{X}-\boldsymbol{Y}\rangle \text{and} \boldsymbol{c}\langle\boldsymbol{Y}\rangle = -\mathbf{1}_{1\times|\boldsymbol{Y}|}, \forall\, Y \in \boldsymbol{Y} \tag{6}$$

where $-\mathbf{1}_{m\times n}$ is an $m \times n$ matrix with values "-1" and $|\boldsymbol{Y}|$ denotes the cardinality of the set $\boldsymbol{Y}$.

When a node has multiple parent nodes, the condition of Eq. (5) is often observed, i.e. under a specific assignment $\boldsymbol{x}$ to some r.v.'s, the assignment to the other r.v.'s $\boldsymbol{Y}$ has no effect on the probability value. Therefore, by skipping the set of assignments over $\boldsymbol{Y}$ altogether, the required memory can be significantly reduced. In the traditional rule-based representation, the assignments over $\boldsymbol{Y}$ in Eq. (5) are simply omitted while only those over $\boldsymbol{X}-\boldsymbol{Y}$ are stored. This strategy is inefficient for implementation as the rules in a set have assignments on the inconsistent sets of r.v.'s, i.e. the data structure is not optimal from a practical perspective. By contrast, CPMs impose the "-1" state on $\boldsymbol{Y}$, as illustrated in Eq. (6), by which all rules have assignments over all r.v.'s in $\boldsymbol{X}$ and as a result, can be collected in a matrix.

Additionally, the compatibility between assignments in MBN is defined as follows.

Definition 2. (*Compatibility in the MBN*)**:** An assignment $\mathbf{c}_1$ to $\boldsymbol{X}$ is compatible with an assignment $\mathbf{c}_2$ to $\boldsymbol{Y}$ if for $\boldsymbol{V} = \{V \in \boldsymbol{X}: \boldsymbol{c}_1\langle V\rangle \neq -1\}$ and $\boldsymbol{W} = \{W \in \boldsymbol{Y}: \boldsymbol{c}_2\langle W\rangle \neq -1\}$, the following equality holds:

$$\boldsymbol{c}_1\langle\boldsymbol{V}\cap\boldsymbol{W}\rangle = \boldsymbol{c}_2\langle\boldsymbol{V}\cap\boldsymbol{W}\rangle \tag{7}$$

In this paper, the compatibility between assignments $\boldsymbol{c}_1$ and $\boldsymbol{c}_2$ is denoted as $\boldsymbol{c}_1 \sim \boldsymbol{c}_2$. To check the compatibility between assignments, the index function can be used to check if the assigned values over $\boldsymbol{V}\cap\boldsymbol{W}$ are identical. Furthermore, **Definition 2** implies the equivalence between the following two statements: (1) the outcomes represented by two assignments are disjoint to each other; and (2) two assignments are not compatible.

Using the rules in **Definition 1**, the CPM is finally defined for conditional PMFs $P(\boldsymbol{X}|\boldsymbol{U})$ as follows. When a marginal distribution of $\boldsymbol{X}$ is considered, $\boldsymbol{U}$ becomes an empty set.

Definition 3. (*Conditional probability matrix*)**:** A CPM of PMF $P(\boldsymbol{X}|\boldsymbol{U})$ is a set of rules $\mathcal{M} = \{\langle\boldsymbol{c}_1; p_1\rangle, \langle\boldsymbol{c}_2; p_2\rangle, \cdots, \langle\boldsymbol{c}_k; p_k\rangle\}$ introduced such that:

- Each rule $\mu \in \mathcal{M}$ has $Scope[\mu] = \boldsymbol{X}\cup\boldsymbol{U}$ which is also defined as $Scope[\mathcal{M}]$.
- For each of the assignments $(\boldsymbol{x}, \boldsymbol{u})$ to $\boldsymbol{X}\cup\boldsymbol{U}$ with no "-1" states, there is either only one rule $\langle\boldsymbol{c}; p\rangle \in \mathcal{M}$ such that $\boldsymbol{c}$ is compatible with $(\boldsymbol{x}, \boldsymbol{u})$, in which case $P(\boldsymbol{x}|\boldsymbol{u}) = p$; or no rule in $\mathcal{M}$, in which case $P(\boldsymbol{x}|\boldsymbol{u}) = 0$.
- 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 $\boldsymbol{c}$ and p of the rules $\mu = \langle\boldsymbol{c}; p\rangle \in \mathcal{M}$.

In analogy to the matrix of the event vectors in the MSR method [25], each column of $\mathbf{C}$ represents the assignments to the corresponding r.v. in $Scope[\mathcal{M}]$.

The second condition of Definition 3 implies that the rules in a CPM must be mutually exclusive while they do not necessarily need to be exhaustive. Such exhaustiveness is required in the original rule-based representation. This relaxation of the necessary condition for the validity of a set of rules is required so that only the rules that affect the inference results can be stored. Specifically, the rules that always have zero probabilities do not appear in CPMs, which is particularly advantageous under the presence of deterministic functions, i.e. a single outcome has a conditional probability of one and all others have probability zero. In addition, as discussed in Section 4.2, CPMs are valid for the case of approximate inference as well, in which only a subset of an event is quantified. It is also noted that although a CPM is a set of rules, its data structure for implementation takes the form of matrices. Therefore, the union and subtraction of two CPMs, i.e. $\mathcal{M}_1 \cup \mathcal{M}_2$ and $\mathcal{M}_1 - \mathcal{M}_2$, refer to the addition and exclusion of rows in the corresponding matrices.

The extra overhead required for CPMs compared to CPTs is the storage of $\mathbf{C}$ required in addition to that of $\mathbf{p}$, and the use of the index function introduced to check compatibility during inference as illustrated in Section 4. Accordingly, when most instances have non-zero probabilities and the size of a given problem is small enough so that all instances can be explicitly enumerated, CPTs are preferable to the MBN. In contrast, when most instances have zero probabilities or the CPTs cannot be quantified due to the large size of a BN node, the gain from selective inclusion of events in quantifying PMFs often dominates the additional cost caused by the storage of $\mathbf{C}$. Moreover, in such large BNs, the high demand on memory is usually the principal problem rather than the computational cost required for inference, which makes MBN the preferable choice.

3.2. Strategy to identify disjoint rules for conditional probability matrices

To construct a valid CPM, the second condition of Definition 3 must be satisfied. However, the use of the "−1" state may obscure whether or not all rules in a CPM have exclusive assignments to each other. One possible way to check this is to examine every pair of the rules in a CPM and confirm that their assignments are not compatible with each other, following the definition of compatibility. Aside from such an elementary check, this section discusses an example strategy to identify a set of disjoint rules.

Using the "−1" state, a set of outcomes can be expressed by one rule. This strategy is classified as implicit enumeration, because individual outcomes are implicitly quantified. A number of methodologies have been proposed to this end, ranging from general methodologies such as fault tree [26], event tree [27], and branch and bound [28] to problem-oriented ones such as decomposition algorithms for network connectivity [15], network flow [29], and structural systems [14]. Underlying these methodologies is the common idea of partitioning an event into the smallest number of disjoint subsets for which the elements can be specified without further investigation.

For example, we here illustrate the branch and bound method, applied to the reliability block diagram (RBD) example of Fig. 2 [5,11]. The r.v.'s X_i, $i = 1, 2, \cdots, 8$ take value 0 when the i-th component has failed and value 1 when it is in operation, while the r.v. X_9 representing the system state, takes value 0 when the source and sink nodes are disconnected and 1 when they are connected, which is determined in function of $\boldsymbol{X}_c = \{X_1, X_2, \cdots, X_8\}$.

Fig. 3 illustrates the process of branch and bound to identify the disjoint subsets of the event associated with $P(X_9|\boldsymbol{X}_c)$. Each subset (marked as a node) arising during this process is branched (marked as an arrow) based on the assignment of each r.v. Specifically, for x_8^0, the RBD is disconnected, i.e. resulting in x_9^0, regardless of the assignments of $\boldsymbol{X}_c - \{X_8\}$. Therefore, the branching is terminated for the subset represented by the assignment $\boldsymbol{c}_1$ over $\boldsymbol{X}_c$ that specifies the failure of X_8. However, for x_8^1, the value of X_9 still depends on the assignment of the r.v.'s in $\boldsymbol{X}_c - \{X_8\}$, so the branching is continued. Since r.v. X_i cannot take values 0 and 1 simultaneously, this branching strategy guarantees that all generated subsets are disjoint. As portrayed in Fig. 3, a total of nine subsets, represented by $\boldsymbol{c}_i$ for $i = 1, 2, \cdots, 9$, suffice to cover the entire event set of $P(X_9|\boldsymbol{X}_c)$.

Consequently, the CPM $\mathcal{M} = \langle \mathbf{C}; \mathbf{p} \rangle$ with these rules $\mu_i = \langle (x_9, \mathbf{c}_i); p \rangle$, $i = 1, \cdots, 9$ is

$$\mathbf{C} = \begin{bmatrix} 0 & -1 & -1 & -1 & -1 & -1 & -1 & -1 & 0 \\ 0 & -1 & -1 & -1 & -1 & -1 & -1 & 0 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 0 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & 0 & 0 & 1 & -1 & -1 & -1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & -1 & -1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & -1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \text{ and } \mathbf{p} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{8}$$

where the columns of $\mathbf{C}$ sequentially correspond to the r.v.'s x_9 and x_i, $i = 1, \cdots, 8$. As should be evident from this simple example, the order of r.v.'s according to which the event is partitioned determines the efficiency of the decomposition, i.e. the smaller the set of components that determines the system state, the more efficient the quantification becomes. Multiple specialized algorithms have been developed to find the optimal order [14,29], e.g. the recursive decomposition algorithm (RDA) [15,16].

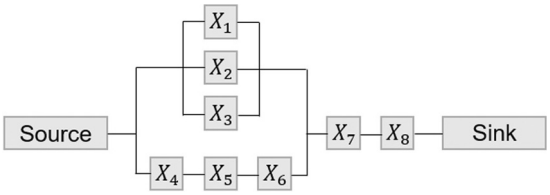


Fig. 2. RBD example.

3.3. Construction of conditional probability matrices for deterministic functions

In order to illustrate the construction of CPMs and demonstrate their efficiency, the CPMs for three basic types of deterministic functions are discussed – namely, conjunction, disjunction, and a general if-then logic. In the following, values of 1 and 0 respectively represent the true and false states. First, the conjunction event of a set of r.v.'s is true if and only if the states of all r.v.'s are true. This condition corresponds to series systems, which are in operation only when all components are in operation. With the branch and bound scheme, the event of $P(X_{N+1}|\boldsymbol{X}_c)$ where the r.v. X_{N+1} is the conjunction of the r.v.'s $\boldsymbol{X}_c = \{C_1, C_2, \cdots, C_N\}$, is decomposed as illustrated in Fig. 4.

As a result, the CPM $\mathcal{M} = \langle \mathbf{C}; \mathbf{p} \rangle$ of $P(X_{N+1}|\boldsymbol{X}_c)$ is modeled with the rules $\mu_i = \langle (x_{N+1}, \boldsymbol{c}_i); p \rangle$, $i = 1, \cdots, (N+1)$ as

$$\mathbf{C} = \begin{bmatrix} 0 & 0 & -1 & -1 & \cdots & -1 \\ 0 & 1 & 0 & -1 & \cdots & -1 \\ 0 & 1 & 1 & 0 & \cdots & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 1 & 1 & 1 & \cdots & 0 \\ 1 & 1 & 1 & 1 & \cdots & 1 \end{bmatrix} \text{ and } \mathbf{p} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \\ 1 \end{bmatrix} \tag{9}$$

where the columns of $\mathbf{C}$ correspond to the r.v.'s x_{N+1} and x_i, $i = 1, \cdots, (N+1)$ in sequence. It is noted that the number of parameters in the CPM of Eq. (9) is $(N+1) \times (N+2)$ while an exhaustive enumeration would require 2^N parameters. This is because an assignment with the "−1" state over a subset of r.v.'s $\boldsymbol{Y}$ stands for $\prod_{Y \in \boldsymbol{Y}} |Val(Y)|$ number of assignments.

When the r.v. X_{N+1} is the disjunction of the r.v.'s $\boldsymbol{X}_c$, X_{N+1} is false if and only if all r.v.'s in $\boldsymbol{X}_c$ are false. A parallel system corresponds to a disjunction event as the system fails only when all of its components fail. Since this logical relationship is equivalent to the negation of conjunction, the CPMs are modeled by switching 0 and 1 in $\{X_{N+1}\} \cup \boldsymbol{X}_c$ of Eq. (9).

For the if-then logic, consider the statement

$$\text{If } \boldsymbol{X}_{\text{if}} = \boldsymbol{x}_{\text{if}}, \text{ then } \boldsymbol{Y}_{\text{then}} = \boldsymbol{y}_{\text{then}}. \tag{10}$$

for two sets of r.v.'s $\boldsymbol{X}_{\text{if}} \subseteq \boldsymbol{X}$ and $\boldsymbol{Y}_{\text{then}} \subseteq \boldsymbol{Y}$. Eq. (10) leads to the rule $\mu = \langle \boldsymbol{c}; p \rangle$ associated with $P(\boldsymbol{Y}|\boldsymbol{X})$ as

$$\begin{aligned} &\boldsymbol{c}\langle \boldsymbol{X}_{\text{if}} \rangle = \boldsymbol{x}_{\text{if}}, \ \boldsymbol{c}\langle \boldsymbol{Y}_{\text{then}} \rangle = \boldsymbol{y}_{\text{then}}, \ \boldsymbol{c}\langle \boldsymbol{X} - \boldsymbol{X}_{\text{if}} \rangle = -\boldsymbol{1}_{1 \times |\boldsymbol{X} - \boldsymbol{X}_{\text{if}}|}, \\ &\boldsymbol{c}\langle \boldsymbol{Y} - \boldsymbol{Y}_{\text{then}} \rangle = -\boldsymbol{1}_{1 \times |\boldsymbol{Y} - \boldsymbol{Y}_{\text{then}}|}, \text{ and } p = 1 \end{aligned} \tag{11}$$

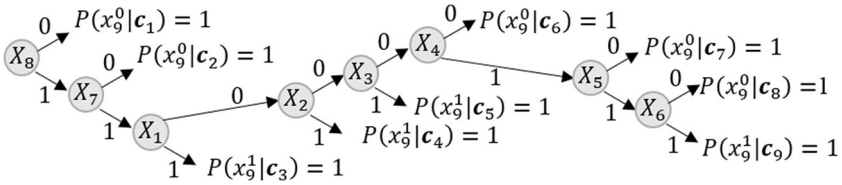
As illustrated by Eq. (11), any if-then statement can be summarized into a single rule while implying the outcomes amounting to $\prod_{Z \in \{(\boldsymbol{X} \cup \boldsymbol{Y}) - (\boldsymbol{X}_{\text{if}} \cup \boldsymbol{Y}_{\text{then}})\}} |val(Z)|$. For example, In Section 3.2, the logical statement corresponding to the first row in Eq. (8) is

$$\text{If } X_8 = 0, \text{ then } X_9 = 0. \tag{12}$$

which covers 2^7 outcomes. Since any deterministic function is equivalent to a set of if-then logical statements, Eq. (11) is applicable to general functions.

3.4. Application of other methodologies to the reliability block diagram example

In this section, the event of X_9 conditioned on $\boldsymbol{X}_c$ in the RBD example of Section 3.2 is modeled by four methodologies [5,6,8,32] other



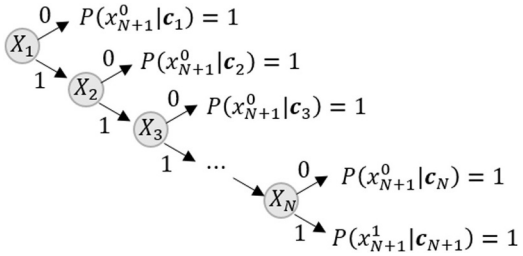


Fig. 4. Branch and bound method to decompose the conjunction event of $\boldsymbol{P}(X_{N+1}|\boldsymbol{X}_c)$.

than the MBN discussed in Section 2.3. The formulation by the MBN – Eq. (8) – and the general comparison between methodologies can be found in Section 2.3.

First, Fig. 5 illustrates the naïve, but most intuitive graphical structure of BN corresponding to the RBD. It is noted that in this naïve modeling, the node X_9 has eight parent nodes, resulting in the CPT quantified by 2^9 parameters, despite the clear regularity identified in Section 3.2. In order to exclude such redundancy, it has been proposed to modify the graphical structure to reduce the number of parent nodes [5,11]. To this end, "super components" are introduced: SC_1 and SC_2 are the series systems respectively of $\{X_4, X_5, X_6\}$ and $\{X_7, X_8\}$; and SC_3 is a parallel system of $\{X_1, X_2, X_3, SC_1\}$. Thereby, the system event becomes a series system of $\{SC_3, SC_2\}$ as illustrated by the modified BN structure in Fig. 6. This alternative representation can reduce the maximum number of parent nodes although the graphical representation becomes less intuitive.

In describing the following three methodologies, i.e. the network polynomial [8], tree-based PMFs [6], and rule-based PMFs [33], the disjoint cut- and link-sets identified in Section 3.2 are utilized. First, the network polynomial formulates the BN inference as the computation of a multilinear function. To this end, two types of variables are introduced – namely, evidence indicators λ_x and network parameters $\theta_{x|\boldsymbol{u}}$ – for each assignment $(x, \boldsymbol{u})$ over a r.v. X and its parent nodes $\boldsymbol{U}$. λ_x is 1 when the assignment x is compatible with a given context, and 0 otherwise, while $\theta_{x|\boldsymbol{u}}$ is the probability $P(x|\boldsymbol{u})$. The BN inference over the r.v.'s $\boldsymbol{X}$ is then formulated with these variables as

$$f_{NP} = \sum_{\boldsymbol{x} \in Val(\boldsymbol{X})} \prod_{(x,\boldsymbol{u}) \sim \boldsymbol{x}} \lambda_x \theta_{x|\boldsymbol{u}} \tag{13}$$

where f_{NP} is equivalent to the probability of any assignment for which λ_x are set. In order to efficiently compute Eq. (13), Darwiche [8] proposed to construct an arithmetic circuit to exploit the identified regularity in the given event.

For the RBD example, the circuit can be constructed as illustrated in Fig. 7. The circuit has become more efficient, i.e. involves less nodes and arrows, utilizing the knowledge of the event. Specifically, the circuit is directly connected to the evidence indicators of X_9 whenever the assignment of X_9 is confirmed, e.g. the multiplication node * of x_8^0 is

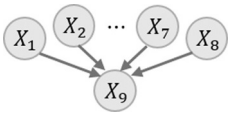


Fig. 5. Naïve BN modeling of the example RBD in Section 3.2.

Fig. 3. Branch and bound method to decompose the system event of the RBD example.

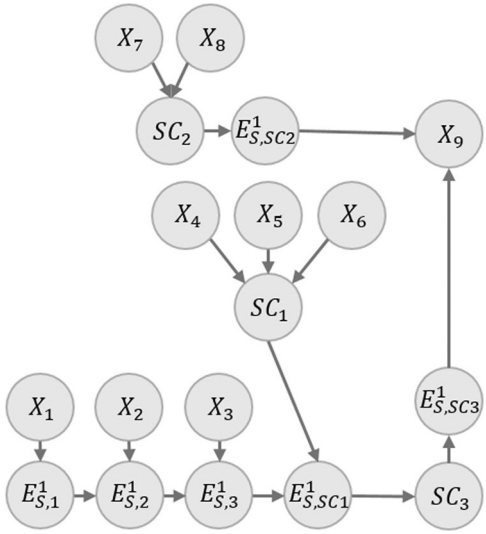


Fig. 6. BN of the RBD example proposed by Bensi [5].

directly connected to that of x_9^0 as x_8^0 immediately leads to x_9^0.

On the other hand, Boutilier et al. [6] proposed to represent a PMF by a tree structure as illustrated in Fig. 8. The tuples at each end of arrows denote $(P(x_9^0|\boldsymbol{c}), P(x_9^1|\boldsymbol{c}))$ for given context $\boldsymbol{c}$ over $\boldsymbol{X}_c$. In both methodologies of the network polynomial and tree-based representation, it is not always straightforward to construct an efficient structure based on the available knowledge. Moreover, their specialized inference methodologies limit their applicability.

Finally, in the original rule-based representation, the rules for $P(X_9|\boldsymbol{X}_c)$ are quantified as

$$\begin{pmatrix}
\rho_1\colon \langle x_8^0, x_9^0; 1\rangle \\
\rho_2\colon \langle x_8^0, x_9^1; 0\rangle \\
\rho_3\colon \langle x_8^1, x_7^0, x_9^0; 1\rangle \\
\rho_4\colon \langle x_8^1, x_7^0, x_9^1; 0\rangle \\
\rho_5\colon \langle x_8^1, x_7^1, x_1^1, x_9^0; 0\rangle \\
\rho_6\colon \langle x_8^1, x_7^1, x_1^1, x_9^1; 1\rangle \\
\rho_7\colon \langle x_8^1, x_7^1, x_1^0, x_2^1, x_9^0; 0\rangle \\
\rho_8\colon \langle x_8^1, x_7^1, x_1^0, x_2^1, x_9^1; 1\rangle \\
\rho_9\colon \langle x_8^1, x_7^1, x_1^0, x_2^0, x_3^1, x_9^0; 0\rangle \\
\rho_{10}\colon \langle x_8^1, x_7^1, x_1^0, x_2^0, x_3^1, x_9^1; 1\rangle \\
\rho_{11}\colon \langle x_8^1, x_7^1, x_1^0, x_2^0, x_3^0, x_4^0, x_9^0; 1\rangle \\
\rho_{12}\colon \langle x_8^1, x_7^1, x_1^0, x_2^0, x_3^0, x_4^0, x_9^1; 0\rangle \\
\rho_{13}\colon \langle x_8^1, x_7^1, x_1^0, x_2^0, x_3^0, x_4^1, x_5^0, x_9^0; 1\rangle \\
\rho_{14}\colon \langle x_8^1, x_7^1, x_1^0, x_2^0, x_3^0, x_4^1, x_5^0, x_9^1; 0\rangle \\
\rho_{15}\colon \langle x_8^1, x_7^1, x_1^0, x_2^0, x_3^0, x_4^1, x_5^1, x_6^0, x_9^0; 1\rangle \\
\rho_{16}\colon \langle x_8^1, x_7^1, x_1^0, x_2^0, x_3^0, x_4^1, x_5^1, x_6^0, x_9^1; 0\rangle \\
\rho_{17}\colon \langle x_8^1, x_7^1, x_1^0, x_2^0, x_3^0, x_4^1, x_5^1, x_6^1, x_9^0; 0\rangle \\
\rho_{18}\colon \langle x_8^1, x_7^1, x_1^0, x_2^0, x_3^0, x_4^1, x_5^1, x_6^1, x_9^1; 1\rangle
\end{pmatrix} \tag{14}$$

where $\rho_i = \langle \boldsymbol{c}; p\rangle$, $i = 1, 2, \cdots, 18$ denote the rules with the assignment $\boldsymbol{c}$ over $\{X_9\} \cup \boldsymbol{X}_c$ and the corresponding probability p. These rules are

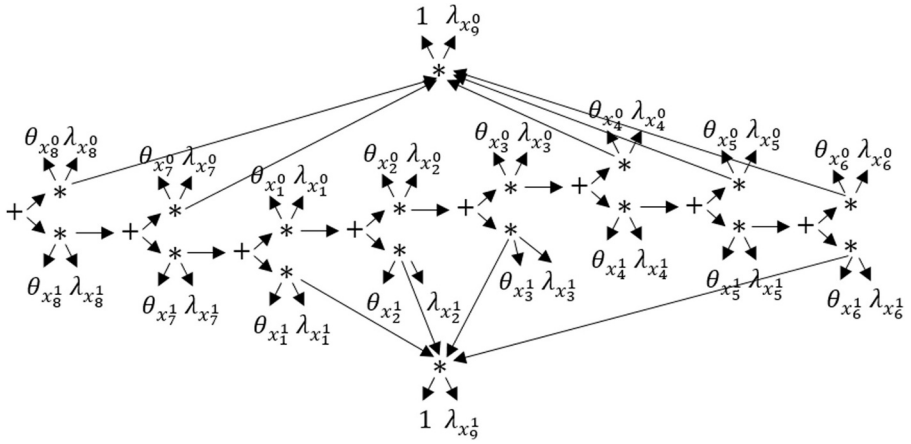


Fig. 7. Arithmetic circuit of the network polynomial for the example RBD.

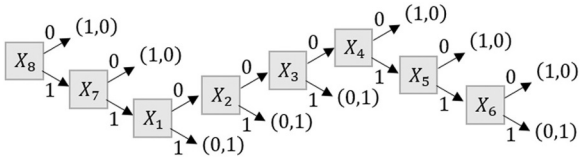


Fig. 8. Tree-based representation of $P(X_9|\boldsymbol{X}_c)$ of the RBD example.

defined similarly to those in the MBN, but also include those with zero probability, doubling the number of rules. Furthermore, the inconsistent lengths of those rules make the data structure inefficient for implementation.

These comparisons highlight the efficiency of the MBN in quantifying a given PMF, compared to the existing. Specifically, the CPM can be modeled with the least number of outcomes compared to the exhaustive quantification and the original rule-based representation. Furthermore, the modeling of the MBN reflecting the regularity in a system is straightforward compared to the other methodologies that employ certain specialized structures [5,6,8,11]. Finally, the data structure of matrix allows the efficient implementation of the MBN compared to the original rule-based representation.

4. Probabilistic inference using the matrix-based Bayesian Network

4.1. Basic operations for inference using the matrix-based Bayesian Network

In this section, basic operations, i.e. conditioning, sum, and product, are formulated for the MBN so that the proposed methodology becomes compatible with existing BN inference methodologies such as variable elimination (VE), clique tree (CT), or conditioning [7]. The developed concepts are similar to their counterparts in the original rule-based representation, but expanded to fit the definitions of rules introduced for the MBN.

The *conditioning* operation allows to introduce evidence (observations) $\boldsymbol{e}$ on nodes in the BN. The definition of conditioning for the MBN is as follows.

Definition 4. (*Conditioning operation in the MBN*): Consider a CPM $\mathcal{M} = \langle \mathbf{C}; \mathbf{p} \rangle$ with $Scope[\mathcal{M}] = \boldsymbol{X}$ and a context $\boldsymbol{E} = \boldsymbol{e}$. Let us consider a reduced CPM

$$\mathcal{M}^* = \{\mu = \langle \boldsymbol{c}; p \rangle \in \mathcal{M} : \boldsymbol{c} \sim \boldsymbol{e}\} = \langle \mathbf{C}^*; \mathbf{p}^* \rangle \tag{15}$$

Then, $\mathcal{M}$ is conditioned on $\boldsymbol{E} = \boldsymbol{e}$, by setting

$$\mathcal{M}[\boldsymbol{e}] = \{\langle \boldsymbol{c}^* \langle \boldsymbol{X} - \boldsymbol{E} \rangle, \boldsymbol{E} = \boldsymbol{e}; p^* \rangle : \langle \boldsymbol{c}^*; p^* \rangle \in \mathcal{M}^*\} \tag{16}$$

After sorting out the rules whose assignments are compatible with the given context $\boldsymbol{E} = \boldsymbol{e}$ as indicated by Eq. (15), Eq. (16) implies that the "−1" state assigned to the r.v.'s in $\boldsymbol{E}$ has to be altered to $\boldsymbol{e}$.

The *sum* operation marginalizes out a r.v. from a given distribution, and is defined in the MBN as follows.

Definition 5. (*Sum operation in the MBN*): Let Y be a variable and μ_i, $i = 1, 2, \cdots, k$ be a rule of the form $\mu_i = \langle \boldsymbol{c}, Y = y_i; p_i \rangle$. Then for $\mathcal{M} = \{\mu_1, \mu_2, \cdots, \mu_k\}$, the sum is defined as $\sum_Y \mathcal{M} = \langle \boldsymbol{c}; \sum_{i=1}^{k} p_i \rangle$.

The definition above leads to the following algorithm of sum operation.

Algorithm 1
Sum using CPMs.

```
    Procedure CPM-sum (
      M    // CPM
      Y    // r.v.'s to be summed out
    )
1   M+ ← M
2   M− ← ∅
3   while M+ ≠ ∅
4     Select Mc ⊆ M+ such that
5       Mc = {⟨c, Y = y1; p1⟩,⋯,⟨c, Y = yk; pk⟩}
6       (no other μ ∈ M+ is compatible with c)
7     M− ← M− ∪ ∑Y Mc
8     M+ ← M+ − Mc
9   return M−
```

Finally, the *product* operation of two rules in the MBN is defined as follows.

Definition 6. (*Product operation in the MBN*): Let $\mu_1 = \langle \boldsymbol{c}_1; p_1 \rangle$ and $\mu_2 = \langle \boldsymbol{c}_2; p_2 \rangle$ be two rules respectively with scopes $\boldsymbol{X}$ and $\boldsymbol{Y}$. If $\boldsymbol{c}_1 \sim \boldsymbol{c}_2$, then their product

$$\mu^* = \mu_1 \cdot \mu_2 = \langle \boldsymbol{c}; p_1 \cdot p_2 \rangle \text{ with } Scope[\mu^*] = \boldsymbol{X} \cup \boldsymbol{Y} \tag{17}$$

where for $Z \in \boldsymbol{X} \cup \boldsymbol{Y}$, $\boldsymbol{c}\langle Z \rangle = \begin{cases} \boldsymbol{c}_1\langle Z \rangle, & \text{if } Z \in \boldsymbol{X} \text{ and } \boldsymbol{c}_1\langle Z \rangle \neq -1 \\ \boldsymbol{c}_2\langle Z \rangle, & \text{if } Z \in \boldsymbol{Y} - \boldsymbol{X} \text{ and } \boldsymbol{c}_2\langle Z \rangle \neq -1 \\ -1, & \text{otherwise} \end{cases}$

The corresponding algorithm for the product operation is then designed as follows.

Algorithm 2
Product using CPMs.

```
Procedure CPM-product (
  M^1, M^2   // CPMs to be product
)
1  M^- ← ∅
2  for each μ_i = ⟨c_i; p_i⟩ ∈ M^1
3    Select M_c ⊆ M^2 such that
4      M_c = {⟨c_1; p_1⟩, ⋯, ⟨c_k; p_k⟩}
5      (no other μ ∈ M^2 is compatible with c_i)
6    M^- ← M^- ∪ {⟨c_1; p_1⟩·μ_i, ⋯, ⟨c_k; p_k⟩·μ_i}
7  return M^-
```

In **Algorithms 1** and **2**, the rules with compatible assignments must be identified for each iteration, which is required $O(|\mathcal{M}|^2)$ and $O(|\mathcal{M}^1|\cdot|\mathcal{M}^2|)$ times, respectively.

4.1.1. Example of the inference using the matrix-based Bayesian Network

Using the three basic operations developed in Section 4.1, the VE algorithm for the MBN can be developed as follows.

Algorithm 3
Sum-product variable elimination using CPMs.

```
Procedure CPM-sum-product-VE (
  M   // Set of CPMs
  Z   // Set of r.v.'s to be eliminated
  ≺   // Ordering on Z
)
1  Let Z_1, ⋯, Z_k be an ordering of Z such that
2    Z_i≺Z_j if and only if i < j
3  for i = 1, ⋯, k
4    M ← CPM-Sum-Product-Eliminate-Var (M, Z_i)
5  M* ← ∏_{M∈M} M
6  return M*

Procedure CPM-Sum-Product-Eliminate-Var (
  M   // Set of CPMs
  Z   // Variable to be eliminated
)
1  M' ← {M ∈ M: Z ∈ Scope[M]}
2  M'' ← M − M'
3  N ← ∏_{M∈M'} M
4  O ← Σ_Z N
5  return M'' ∪ {O}
```

Similarly, other BN inference algorithms such as CT or conditioning, can also be developed for the MBN by applying the newly defined basic operations to existing algorithms.

4.2. Approximate Bayesian Network inference using non-exhaustive conditional probability matrices

When only a subset of an event is used for inference due to either infeasible computational cost or excessive memory demand for storing all outcomes, approximate inference can be carried out to compute the bounds of the quantity of interest. The derivation of these bounds depends on how the subset has been chosen, i.e. either deterministically [13–16] or stochastically [16–18,29–31]. In this section, the methodologies for both cases are developed in the framework of the MBN. Their specific applications are illustrated by the RBD example (Fig. 2) later in Section 4.2.3, and a real-world benchmark network in Section 5.2. In the following, a non-exhaustive CPM refers to the CPM having some missing rules with non-zero probabilities.

4.2.1. Non-exhaustive CPMs by deterministic selection

When a subset of outcomes are selected deterministically, the bounds of an estimate is computed deterministically as well. A joint PMF derived under the presence of non-exhaustive CPMs, ends up being non-exhaustive, i.e. it does not sum up to one. Therefore, the result of inference with such joint PMF provides the lower bound of the quantity (Chapter 12.5 in [7]). Similarly, the upper bound can be computed by subtracting the lower bound of the complementary event from unity, leading to the bounds formulated as

$$\tilde{P}(\boldsymbol{x}) \le P(\boldsymbol{x}) \le 1 - \tilde{P}(\{\boldsymbol{x}\}^c) \tag{18}$$

where $\{\boldsymbol{x}\}^c = Val(\boldsymbol{X}) - \{\boldsymbol{x}\}$, and $P(\boldsymbol{x})$ and $\tilde{P}(\cdot)$ respectively refer to the probability of interest and the inference result under the presence of non-exhaustive CPMs. Likewise, the bounds for the posterior distribution

$$P(\boldsymbol{x}|\boldsymbol{e}) = \frac{P(\boldsymbol{x}, \boldsymbol{e})}{P(\boldsymbol{e})} \tag{19}$$

can be derived as follows using the bounds of $P(\boldsymbol{x}, \boldsymbol{e})$ and $P(\boldsymbol{e})$ computed based on Eq. (18):

$$\frac{\tilde{P}(\boldsymbol{x}, \boldsymbol{e})}{1 - \tilde{P}(\{\boldsymbol{e}\}^c)} \le P(\boldsymbol{x}|\boldsymbol{e}) \le \frac{1 - \tilde{P}(\{(\boldsymbol{x}, \boldsymbol{e})\}^c)}{\tilde{P}(\boldsymbol{e})} \tag{20}$$

It is noted that when the bounds either on $\tilde{P}(\{(\boldsymbol{x}, \boldsymbol{e})\}^c)$ or $\tilde{P}(\boldsymbol{e})$ are wide, the upper bound of Eq. (20) can exceed one, in which case it is set to one.

4.2.2. Non-exhaustive CPMs by stochastic selection

As a generalization of sampling using BNs, sampling and analytic inference can be concurrently applied respectively over two disjoint subsets of r.v.'s. When one of the two sets is empty, it corresponds to either sampling-based or exact inference of BNs. This strategy is called *Rao–Blackwellizing* [28], and the expectation of a function $f(\boldsymbol{x})$ of assignment $\boldsymbol{x} = (\boldsymbol{x}_p, \boldsymbol{x}_d, \boldsymbol{e})$ with regard to a posterior distribution $P(\boldsymbol{X}|\boldsymbol{E})$ is computed as

$$\begin{aligned} E_{P(\boldsymbol{x}|\boldsymbol{e})}[f(\boldsymbol{x})] &= \sum_{\boldsymbol{x}_p,\boldsymbol{x}_d} P(\boldsymbol{x}_p, \boldsymbol{x}_d|\boldsymbol{e}) f(\boldsymbol{x}_p, \boldsymbol{x}_d, \boldsymbol{e}) \\ &= \sum_{\boldsymbol{x}_p,\boldsymbol{x}_d} \frac{P(\boldsymbol{x}_p, \boldsymbol{x}_d, \boldsymbol{e})}{P(\boldsymbol{e})} f(\boldsymbol{x}_p, \boldsymbol{x}_d, \boldsymbol{e}) \\ &= \frac{1}{P(\boldsymbol{e})} \sum_{\boldsymbol{x}_p,\boldsymbol{x}_d} Q(\boldsymbol{x}_p) \frac{P(\boldsymbol{x}_p, \boldsymbol{x}_d, \boldsymbol{e})}{Q(\boldsymbol{x}_p)} f(\boldsymbol{x}_p, \boldsymbol{x}_d, \boldsymbol{e}) \\ &= \frac{1}{P(\boldsymbol{e})} \sum_{\boldsymbol{x}_p} Q(\boldsymbol{x}_p) \frac{P(\boldsymbol{x}_p, \boldsymbol{e})}{Q(\boldsymbol{x}_p)} \sum_{\boldsymbol{x}_d} P(\boldsymbol{x}_d|\boldsymbol{x}_p, \boldsymbol{e}) f(\boldsymbol{x}_p, \boldsymbol{x}_d, \boldsymbol{e}) \end{aligned} \tag{21}$$

where $\boldsymbol{X}_p$ is the set of r.v.'s for which samples are generated from the importance sampling density $Q(\boldsymbol{X}_p)$, and $\boldsymbol{X}_d = \boldsymbol{X} - \boldsymbol{X}_p$ are the set of r.v.'s over which the exact inference is carried out. Defining the weight of each sample as $\omega(\boldsymbol{x}_p) = P(\boldsymbol{x}_p, \boldsymbol{e})/Q(\boldsymbol{x}_p)$, Eq. (21) is written in short notation as

$$E_{P(\boldsymbol{x}|\boldsymbol{e})}[f(\boldsymbol{x})] = \frac{1}{P(\boldsymbol{e})} E_{Q(\boldsymbol{X}_p)}\left[\omega(\boldsymbol{X}_p) E_{P(\boldsymbol{x}_d|\boldsymbol{x}_p,\boldsymbol{e})}[f(\boldsymbol{x}_p, \boldsymbol{x}_d, \boldsymbol{e})]\right] \tag{22}$$

When the marginal distribution $P(\boldsymbol{e})$ cannot be computed without the CPMs whose scopes include some r.v.'s in $\boldsymbol{X}_p$, $P(\boldsymbol{e})$ in the denominator is derived as

$$P(\boldsymbol{e}) = \sum_{\boldsymbol{x}_p} P(\boldsymbol{x}_p, \boldsymbol{e}) = \sum_{\boldsymbol{x}_p} Q(\boldsymbol{x}_p) \frac{P(\boldsymbol{x}_p, \boldsymbol{e})}{Q(\boldsymbol{x}_p)} = E_{Q(\boldsymbol{X}_p)}[\omega(\boldsymbol{X}_p)] \tag{23}$$

leading to the formulation

$$E_{P(\boldsymbol{x}|\boldsymbol{e})}[f(\boldsymbol{x})] = \frac{E_{Q(\boldsymbol{X}_p)}\left[\omega(\boldsymbol{X}_p)E_{P(\boldsymbol{x}_d|\boldsymbol{x}_p,\boldsymbol{e})}[f(\boldsymbol{x}_p,\boldsymbol{x}_d,\boldsymbol{e})]\right]}{E_{Q(\boldsymbol{X}_p)}[\omega(\boldsymbol{X}_p)]} \tag{24}$$

which corresponds to normalized importance sampling (IS). Otherwise, when $P(\boldsymbol{e})$ can be analytically computed without interfering with $\boldsymbol{X}_p$, i.e. can be computed only with the CPMs whose scopes are the subsets of $\boldsymbol{X}_d$, Eq. (22) corresponds to the formulation of an unnormalized IS (Details about IS using probabilistic graphical models can be found in Chapter 12 of [7].)

In order to compute Eq. (22) or (24), the data set of M samples

$$\begin{aligned}&\mathcal{D} = \{(\boldsymbol{x}_p[m], \omega(\boldsymbol{x}_p[m]), \mu_d(\boldsymbol{x}_p[m]))\}_{m=1}^{M}\\ &\text{where } \mu_d(\boldsymbol{x}_p[m]) = E_{P(\boldsymbol{x}_d|\boldsymbol{x}_p[m],\boldsymbol{e})}[f(\boldsymbol{x}_p[m],\boldsymbol{x}_d,\boldsymbol{e})]\end{aligned} \tag{25}$$

needs to be generated by computing $\mu_d(\boldsymbol{x}_p[m])$ for each sample using any analytical BN inference methodology. To this end, the identical BN inference procedure over $\boldsymbol{X}_d$ should be applied to each sample with the MBN conditioned on the assignment $(\boldsymbol{x}_p, \boldsymbol{x}_d, \boldsymbol{e})$. This leads to the following algorithm for Rao–Blackwellizing using the MBN.

Algorithm 4
Rao–Blackwellizing in the MBN.

Procedure MBN-Rao–Blackwellizing (
  $\boldsymbol{P}$ // Set of M tuples $\langle \boldsymbol{x}_p; Q(\boldsymbol{x}_p)\rangle$
  $\boldsymbol{M}$ // Set of CPMs
  f // Function to be estimated
  $\boldsymbol{E} = \boldsymbol{e}$ // Context on $\boldsymbol{E}$
  $\boldsymbol{Z}$ // Set of r.v.'s to be eliminated (The r.v.'s that do not affect the value of f)
  $\prec$ // Ordering on $\boldsymbol{Z}$
)
1 Let $\boldsymbol{Y}$ be $Scope[\boldsymbol{M}] - \boldsymbol{X}_p - \boldsymbol{E} - \boldsymbol{Z}$
2 $\mathcal{D}^- \leftarrow \emptyset$
3 **for** each tuple $\langle \boldsymbol{x}_p, Q(\boldsymbol{x}_p)\rangle \in \boldsymbol{P}$
4 $\mathcal{M} \leftarrow$ Reduce-and-VE ($\boldsymbol{M}$, $\langle \boldsymbol{x}_p, \boldsymbol{e}\rangle$, $\boldsymbol{Z}$, $\prec$)
5 $\langle \omega, \mu_d\rangle \leftarrow$ Compute-particle ($\mathcal{M}$, $\boldsymbol{Y}$, $Q(\boldsymbol{x}_p)$, f)
6 $\mathcal{D}^- \leftarrow \mathcal{D}^- \cup \{\langle \omega, \mu_d\rangle\}$
7 **return** $\mathcal{D}^-$

Procedure Reduce-and-VE (
  $\boldsymbol{M}$ // Set of CPMs
  $\boldsymbol{E} = \boldsymbol{e}$ // Context on $\boldsymbol{E}$
  $\boldsymbol{Z}$ // Set of r.v.'s to be eliminated
  $\prec$ // Ordering on $\boldsymbol{Z}$
)
1 Reduce $\mathcal{M} \in \boldsymbol{M}$ as
2 $\boldsymbol{M} = \{\mathcal{M}_1[\boldsymbol{e}], \cdots, \mathcal{M}_k[\boldsymbol{e}]\}$
3 $\mathcal{M} \leftarrow$ CPM-sum-product-VE ($\boldsymbol{M}$, $\boldsymbol{Z}$, $\prec$)
4 **return** $\mathcal{M}$

Procedure compute-particle (
  $\mathcal{M}$ // CPM
  $\boldsymbol{Y}$ // Set of r.v.'s over which expectation is computed
  $Q(\boldsymbol{x}_p)$ // Probability of $\boldsymbol{x}_p$ being sampled from $Q(\boldsymbol{X}_p)$
  f // Function whose expectation is to be estimated
)
1 $P(\boldsymbol{x}_p, \boldsymbol{e}) \leftarrow$ CPM-Sum ($\mathcal{M}$, $\boldsymbol{Y}$)
2 $\omega \leftarrow P(\boldsymbol{x}_p, \boldsymbol{e})/Q(\boldsymbol{x}_p)$
3 $\mu_d \leftarrow 0$
4 **for** each rule $\mu = \langle \mathbf{c}; p\rangle \in \mathcal{M}$
5 $\mu_d \leftarrow \mu_d + p \cdot f(\mathbf{c})$
6 $\mu_d \leftarrow \mu_d / P(\boldsymbol{x}_p, \boldsymbol{e})$
7 **return** $\langle \omega, \mu_d\rangle$

Although the VE is employed in Algorithm 4 for the exact inference over $\boldsymbol{X}_d$, any exact BN inference method is applicable.

As a result, the expectation of Eq. (24) is estimated as

$$\hat{\mu} = \frac{\sum_{m=1}^{M}\omega(\boldsymbol{x}_p[m])\cdot\mu_d(\boldsymbol{x}_p[m])}{\sum_{m=1}^{M}\omega(\boldsymbol{x}_p[m])} \tag{26}$$

with the variance estimated as

$$\hat{\sigma}^2 = \frac{\sum_{m=1}^{M}\omega(\boldsymbol{x}_p[m])^2\{\mu_d(\boldsymbol{x}_p[m]) - \hat{\mu}\}^2}{\{\sum_{m=1}^{M}\omega(\boldsymbol{x}_p[m])\}^2} \tag{27}$$

On the other hand, when $P(\boldsymbol{e})$ can be analytically computed, the expectation of Eq. (22) can be estimated – note that each $\mu_d(\boldsymbol{x}_p[m])$ is already divided by $P(\boldsymbol{e})$ in the proposed algorithm – as

$$\hat{\mu} = \frac{1}{M}\sum_{m=1}^{M}\omega(\boldsymbol{x}_p[m])\cdot\mu_d(\boldsymbol{x}_p[m]) \tag{28}$$

with the sampling variance

$$\hat{\sigma}^2 = \frac{1}{M}\sum_{m=1}^{M}\{\omega(\boldsymbol{x}_p[m])\mu_d(\boldsymbol{x}_p[m]) - \hat{\mu}\}^2 \tag{29}$$

When there is a correlation between samples, as is the case for some sampling techniques, in particular Markov Chain Monte Carlo methods [17,19,31], the variances in Eqs. (27) and (29) need to include the auto-covariance terms additionally.

4.2.3. Approximate inference of the reliability block diagram example by MBN

This section demonstrates the approximate inference using non-exhaustive set of events by estimating $P(x_1^1|x_9^1)$ in the RBD example. In the following, the BN in Fig. 5 is assumed to have the CPMs $\langle \mathbf{C}; \mathbf{p}\rangle$ of $P(X_i)$, $i = 1, 2, \cdots, 8$ where

$$\mathbf{C} = \begin{bmatrix} x_i^0 \\ x_i^1 \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix} \tag{30}$$

First, as a non-exhaustive CPM with deterministically selected rules, consider the CPM of $P(X_9|\boldsymbol{X}_c)$ in Eq. (8) with the last two rows missing. The inference results of $P(x_1^1, x_9^1)$ and $P(x_9^1)$ are then computed as

$$\begin{aligned}&\tilde{P}(x_9^1, x_1^1) = 0.7290\\ &\tilde{P}(\{x_9^1, x_1^1\}^c) = 0.2703\\ &\tilde{P}(x_9^1) = 0.8092\\ &\tilde{P}(\{x_9^1\}^c) = 0.1901\end{aligned} \tag{31}$$

Accordingly, the bounds of $P(x_1^1|x_9^1)$ are evaluated by Eq. (20), i.e.

$$\frac{0.7290}{1 - 0.1901} = 0.9001 \le P(x_1^1|x_9^1) \le \frac{1 - 0.2703}{0.8092} = 0.9018 \tag{32}$$

To demonstrate the stochastic approach inference following Section 4.2.2, we consider the following 10 samples of $\boldsymbol{P} = \{\langle \boldsymbol{x}_p[m]; Q(\boldsymbol{x}_p[m])\rangle\}_{m=1}^{10}$ with $\boldsymbol{X}_p = \boldsymbol{X}_c$, where the assignments of $\boldsymbol{X}_c$ have been drawn from the binomial distribution with probability 0.9:

$$\boldsymbol{P} = \begin{pmatrix} \langle(x_1^0, x_2^1, x_3^1, x_4^1, x_5^1, x_6^0, x_7^1, x_8^1); 0.9^6\cdot0.1^2 = 5.314 \times 10^{-3}\rangle \\ \langle(x_1^1, x_2^1, x_3^1, x_4^1, x_5^0, x_6^1, x_7^1, x_8^1); 0.9^7\cdot0.1^1 = 4.783 \times 10^{-2}\rangle \\ \langle(x_1^1, x_2^1, x_3^1, x_4^1, x_5^1, x_6^1, x_7^1, x_8^1); 0.9^8\cdot0.1^0 = 4.305 \times 10^{-1}\rangle \\ \langle(x_1^1, x_2^1, x_3^1, x_4^1, x_5^1, x_6^0, x_7^1, x_8^1); 0.9^7\cdot0.1^1 = 4.783 \times 10^{-2}\rangle \\ \langle(x_1^1, x_2^1, x_3^1, x_4^1, x_5^1, x_6^0, x_7^1, x_8^1); 0.9^7\cdot0.1^1 = 4.783 \times 10^{-2}\rangle \\ \langle(x_1^1, x_2^1, x_3^1, x_4^1, x_5^1, x_6^1, x_7^1, x_8^1); 0.9^8\cdot0.1^0 = 4.305 \times 10^{-1}\rangle \\ \langle(x_1^1, x_2^1, x_3^1, x_4^1, x_5^1, x_6^1, x_7^0, x_8^1); 0.9^7\cdot0.1^1 = 4.783 \times 10^{-2}\rangle \\ \langle(x_1^1, x_2^1, x_3^1, x_4^1, x_5^1, x_6^1, x_7^1, x_8^1); 0.9^8\cdot0.1^0 = 4.305 \times 10^{-1}\rangle \\ \langle(x_1^1, x_2^1, x_3^1, x_4^1, x_5^1, x_6^1, x_7^1, x_8^1); 0.9^8\cdot0.1^0 = 4.305 \times 10^{-1}\rangle \\ \langle(x_1^1, x_2^1, x_3^1, x_4^1, x_5^1, x_6^1, x_7^1, x_8^0); 0.9^7\cdot0.1^1 = 4.783 \times 10^{-2}\rangle \end{pmatrix} \tag{33}$$

Under each $\boldsymbol{x}_p[m]$ for $m = 1, 2, \cdots, 10$, the assignment of $\boldsymbol{X}_d = \{X_9\}$ is determined by the observation whether the system is connected, leading to the CPM $\langle \mathbf{C}\langle X_9 \cup \boldsymbol{X}_c\rangle; \mathbf{p}\rangle$ for $P(X_9|\boldsymbol{X}_c)$ as

$$\mathbf{C} = \begin{bmatrix} 1 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}, \quad \mathbf{p} = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{34}$$

with the columns of **C** that sequentially represent the r.v.'s x_9 and x_i, i = 1, ⋯,8. The rules in Eq. (34) are reduced to five as some instances appear several times in $\boldsymbol{P}$ of Eq. (33). Following Algorithm 4 for the query $P(x_1^1|x_9^1)$, along with the set of CPMs $\boldsymbol{M}$ of the CPMs for X_i, $i = 1, 2, \cdots, 8$ in Eq. (30) and the CPM $\langle \mathbf{C}; \mathbf{p} \rangle$ for $P(X_9|\boldsymbol{X}_c)$ in Eq. (34), the other input variables are set as

$$\begin{aligned} &f = \mathrm{II}[(x_1^1, x_9^1)] \\ &\boldsymbol{E} = \{X_9\}, \ \ \boldsymbol{e} = \{x_9^1\} \\ &\boldsymbol{Z} = \{X_9, X_1, X_2, \cdots, X_8\} - \{X_9, X_1\} = \{X_2, X_3, \cdots, X_8\}, \text{ and} \\ &X_2 \prec X_3 \prec \cdots \prec X_8 \end{aligned} \tag{35}$$

where $\mathrm{II}[\cdot]$ is the index function, i.e. having the value 1 when the given statement is true and 0 otherwise. With this setting, the operation to get the CPM $\mathcal{M}_1' = \langle \mathbf{C}'_1; \mathbf{p}'_1 \rangle$ on $P(X_1, X_9)$ for the first sample

$$\mathcal{M}_1' \leftarrow \text{Reduce} - \text{and} - \text{VE}(\boldsymbol{M}, \langle \boldsymbol{e}, \boldsymbol{x}_p[1] \rangle, \boldsymbol{Z}, \prec) \tag{36}$$

leads to

$$\mathbf{C}'_1 = [0 \ \ 1] \text{ and } \mathbf{p}'_1 = [0.9^6 \cdot 0.1^2 = 5.314 \ \times 10^{-3}] \tag{37}$$

while the same operation for the second sample yields the CPM $\mathcal{M}_2' = \langle \mathbf{C}'_2; \mathbf{p}'_2 \rangle$ with

$$\mathbf{C}'_2 = [1 \ \ 1] \text{ and } \mathbf{p}'_2 = [0.9^7 \cdot 0.1^1 = 4.783 \ \times 10^{-2}] \tag{38}$$

Then, the next operation on the first sample of

$$\langle \omega(\boldsymbol{x}_p[1]), \mu(\boldsymbol{x}_p[1]) \rangle \leftarrow \text{Compute} - \text{Particle}(\mathcal{M}'_1, \varnothing, Q(\boldsymbol{x}_p[1])) \tag{39}$$

computes

$$\omega(\boldsymbol{x}_p[1]) = 0 \text{ and } \mu(\boldsymbol{x}_p[1]) = 0 \tag{40}$$

as there remain no rules compatible with (x_1^1, x_9^1), the result for the second sample is

$$\omega(\boldsymbol{x}_p[2]) = \frac{4.783 \ \times 10^{-2}}{4.783 \ \times 10^{-2}} = 1, \ \text{ and } \mu(\boldsymbol{x}_p[2]) = \frac{4.783 \ \times 10^{-2}}{4.783 \ \times 10^{-2}} = 1 \tag{41}$$

whose computation is also trivial as there is only one rule.

Consequently, in this simple example, the data set $\mathcal{D} = \{\langle \omega(\boldsymbol{x}_p[m]), \mu_d(\boldsymbol{x}_p[m]) \rangle\}_{m=1}^{10}$ is derived from the deterministic relationship between S and $\boldsymbol{C}$, as

$$\omega(\boldsymbol{x}_p[m]) = \begin{cases} 1, & \text{if } P(x_9^1|\boldsymbol{x}_p[m]) = 1 \\ 0, & \text{otherwise} \end{cases} \tag{42}$$

and

$$\mu(\boldsymbol{x}_p[m]) = \begin{cases} 1, & \text{if } P(x_9^1|\boldsymbol{x}_p[m]) = 1 \text{ and } \boldsymbol{x}_p[m]\langle X_1 \rangle = x_1^1 \\ 0, & \text{otherwise} \end{cases} \tag{43}$$

Finally, $P(x_1^1|x_9^1)$ is estimated as

$$\hat{\mu} = \frac{7}{8} = 0.8750 \tag{44}$$

with the sampling variance $\hat{\sigma}^2$ and coefficient of covariance $\hat{\delta}$

$$\begin{aligned} &\hat{\sigma}^2 = \frac{7 \cdot (1 - 0.8750)^2 + 1 \cdot (0 - 0.8750)^2}{8^2} = 0.01367 \text{ and} \\ &\hat{\delta} = \frac{1}{|\hat{\mu}|}\sqrt{\frac{\hat{\sigma}^2}{8}} = 0.04725 \end{aligned} \tag{45}$$

The final number of samples is decreased to eight as reflected in Eq. (45) as two samples are rejected because of their inconsistency with the context $\boldsymbol{e}$. The results well approximate the exact result of $P(x_9^1|x_1^1) = 0.9008$, which demonstrates that using the MBN, the BN methodologies of modeling and inference can be extended to the approximate inference. Such capability of general probabilistic inference demonstrates the utility of BN as well, especially compared to standard analysis methodologies such as fault tree analysis [26].

5. Numerical examples

5.1. MBN application to connectivity of random graphs

To demonstrate the efficiency of the MBN, the connectivity of random graphs [34] is modeled and investigated using CPMs. The size of these graphs is increased from 11 nodes and 22 uni-directional links to 21 nodes and 55 links by adding the nodes and links such that roughly 10% of possible links are randomly selected. Fig. 9 depicts the final graph after the random additions of nodes and links.

It is assumed that the links are the only components that can fail, i.e. the nodes do not fail. The system is considered to have survived if there is at least one path between the source node 1 and the terminal node 11, and failed otherwise. Since the connectivity is determined by the states of the links, the graphical structure of the BN is modeled by a converging structure as in Fig. 5 with the number of parent nodes equal to that of links, denoted by N. Therefore, this naïve formulation requires 2^{N+1} parameters to describe the PMF of the child node – the r.v. that represents the connectivity of the graph – if a CPT is used for such BN.

On the contrary, the MBN can be modeled by identifying the disjoint rules employing the RDA [15,16], which systematically identifies the disjoint subsets of an event for network connectivity problems. The methodology follows an equivalent principle with the branch and bound framework illustrated in Section 3.2. The order of r.v.'s according to which the event is decomposed is determined by sequentially searching the shortest paths from the source to the terminal nodes. Then, the CPM for $P(X_{12}|\boldsymbol{X}_c)$ can be constructed based on these identified paths.

For example, let us suppose that among the 11 links $\boldsymbol{X}_c = \{X_1, X_2, \cdots, X_{11}\}$, the RDA identifies that failure of X_1, X_5, and X_7 results in the disconnection of the system (X_{12}). This cut-set corresponds to the rule $\mu = \langle \mathbf{c} \langle \{X_{12}\} \cup \boldsymbol{X}_c \rangle; \mathbf{p} \rangle$ with

$$\begin{aligned} \mathbf{c} \ &= \ [0 \ \ 0 \ \ -1 \ \ -1 \ \ -1 \ \ 0 \ \ -1 \ \ 0 \ \ -1 \ \ -1 \ \ -1 \ \ -1] \text{ and} \\ \mathbf{p} \ &= \ [1] \end{aligned} \tag{46}$$

where the value 0 denotes failure of the corresponding link or the system, and the elements of **c** sequentially represent the assignments over the r.v.'s x_{12} and x_i, $i = 1, \cdots, 11$. Fig. 10 shows the number of disjoint subsets identified by the RDA until the entire event space is covered. It is noted that each identified subset produces one rule, and

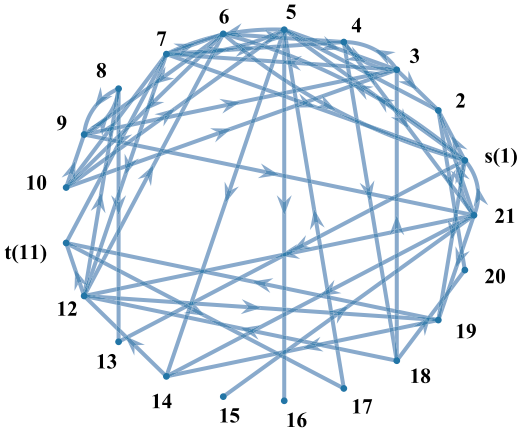


Fig. 9. The example random graph with 21 nodes and 55 links.

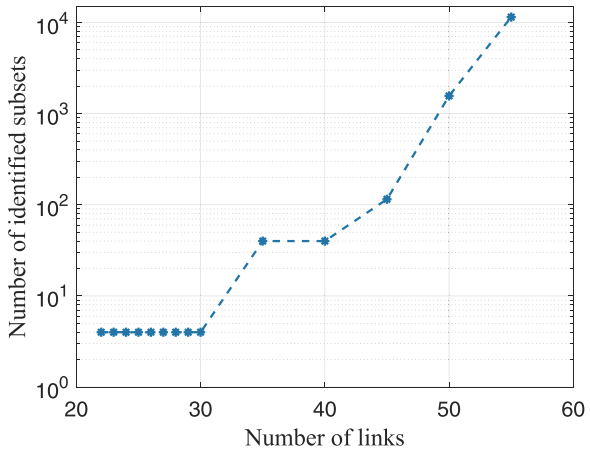


Fig. 10. The number of disjoint subsets identified by the RDA for the random graphs.

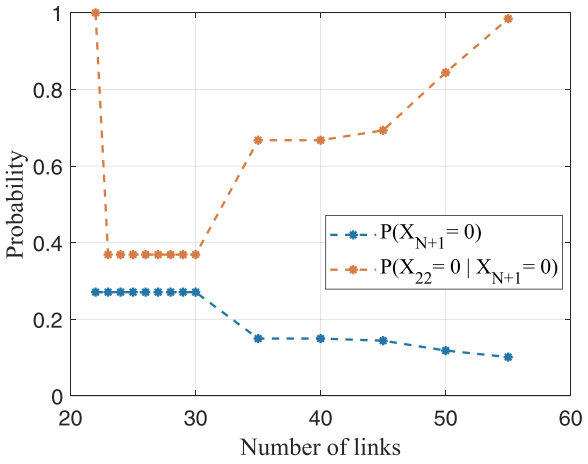


Fig. 12. The probabilities $\boldsymbol{P}(\boldsymbol{x}^0_{N+1})$ and $\boldsymbol{P}(\boldsymbol{x}^0_{22}|\boldsymbol{x}^0_{N+1})$ in the random graphs.

the comparison between the number of parameters required for the MBN – the elements in the CPMs – and the naïve formulation is provided in Fig. 11. Furthermore, the probability of disconnection $P(x^0_{N+1})$ and the conditional probability $P(x^0_{22}|x^0_{N+1})$ are evaluated assuming that the r.v.'s X_i, $i = 1, 2, \cdots, N$ have CPM $\langle \mathbf{C}; \mathbf{p} \rangle$ with

$$\mathbf{C} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \text{ and } \mathbf{p} = \begin{bmatrix} 0.1 \\ 0.9 \end{bmatrix} \tag{47}$$

where X_{22} corresponds to the link that connects nodes 7 and 11. Fig. 12 shows the inference results of the two probabilistic quantities for each graph.

The results confirm that the MBN can model the BN much more efficiently than the naïve formulation, and facilitate straightforward exploitation of the given information on an event. This suggests that the MBN can extend the applicability of BNs to larger systems.

5.2. MBN application to connectivity of Sioux Falls benchmark network

In order to demonstrate MBN-based approximate inference of a large-size network, we investigate the connectivity between the source and terminal nodes of the Sioux Falls network [25], illustrated in Fig. 13. The network consists of 24 nodes and 76 uni-directional links, wherein the state of each link is determined by a reinforced concrete (RC) bridge, and the links are the only components that can fail. Fig. 14 describes the graphical structure of the BN for the network where the r.v.'s M and L respectively are the magnitude and location of an earthquake, and X_{77} represents the connectivity of the system – taking the value 1 for connection and 0 for disconnection between the source and terminal nodes. In addition, D_i represents the deterioration state (1 for being deteriorated and 0 otherwise), I_i the result of inspection (1 for being observed as deteriorated and 0 otherwise), and X_i the bridge operation state (1 for being in operation and 0 for failure) of the i-th link, $i = 1, \cdots, 76$.

Due to the large number of links, exact inference by BN is infeasible, and thus the CPM for $P(X_{77}|\boldsymbol{X}_c)$ remains non-exhaustive. The probability queries of interest in this example, are $P(x^0_{77})$ and $P(x^0_{38}|x^0_{77}, i^1_{38})$. We apply deterministic approximate inference, wherein we identify the rules in the CPM for $P(X_{77}|\boldsymbol{X}_c)$ by the RDA, and stochastic inference.

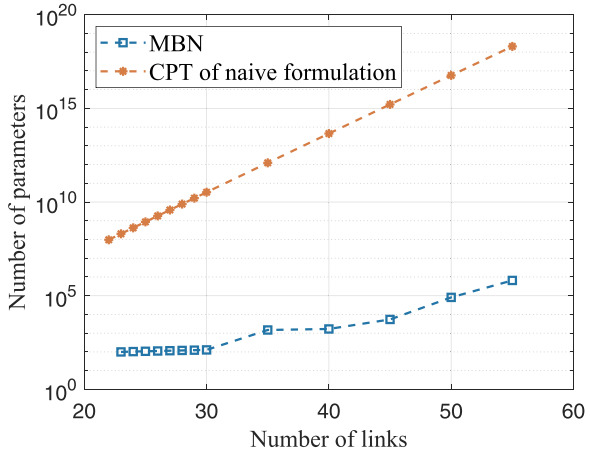


Fig. 11. The number of parameters required for the MBN and the CPT of naïve formulation.

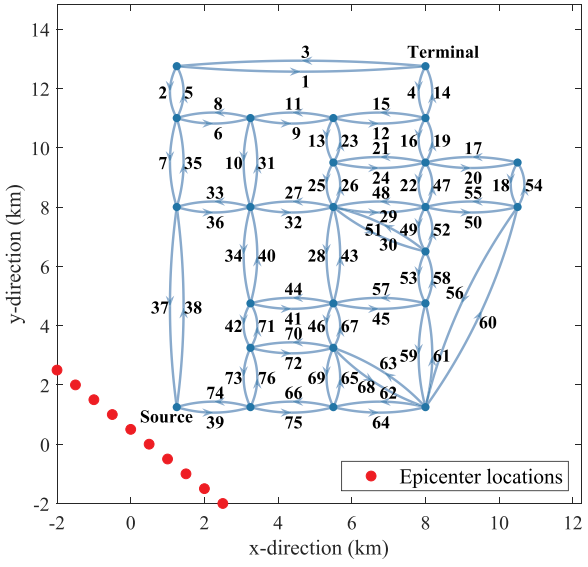


Fig. 13. The Sioux Falls network and the hypothetical epicenter locations.

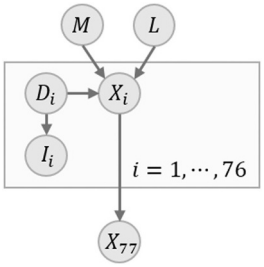


Fig. 14. The BN for the Sioux Falls network.

5.2.1. Conditional probability distribution of each random variable

The magnitude M is assumed to follow the truncated exponential distribution [35] as

$$f_M(m) = \begin{cases} \frac{\beta \exp[-\beta(m-m_0)]}{1-\exp[-\beta(m-m_0)]}, & \text{for } m_0 \le m \le m_p \\ 0, & \text{elsewhere} \end{cases} \quad (48)$$

where the parameter β, and the minimum m_0 and the maximum m_p are respectively set to 0.76, 6.0, and 8.5. M is discretized into five states with intervals $(-\infty, 6.5]$, (6.5, 7], (7, 7.5], (7.5, 8], and $(8, \infty)$. For L, ten discrete locations of epicenter are considered, each having probability 0.1. The locations range from -2.0 km to 2.5 km in x-coordinate and 2.5 km to -2.0 km in y-coordinate as illustrated in Fig. 13.

$P(D_i)$ for $i = 1, 2, \cdots, 76$ are determined by simplifying the RC column deterioration model proposed by Choe et al. [36]. The formula for time T_{corr} when the chloride concentration at the cover depth of the reinforcement reaches the critical chloride concentration is simplified to

$$T_{\text{corr}} = C\frac{X_I}{k_e} \quad (49)$$

where X_I, k_e, and C respectively refer to the model uncertainty coefficient to account for the idealization of Fick's second law, the environmental factor, and the coefficient that summarizes the other terms in the original formulation. In this example, k_e and C are assumed to be constant while X_I is the only r.v., following a lognormal distribution with mean 1 and the standard deviation 0.05. C is assumed to be 100, while k_e is 0.924 for bridges under tidal condition (Links 37, 38, 39, 42, 62, 64, 65, 66, 69, 70, 71, 72, 73, 74, 75, and 76), 0.265 for bridges under splash condition (Links 7, 33, 34, 35, 36, 40, 41, 44, 45, 46, 57, 59, 61, 63, 67, and 68), and 0.676 for bridg under atmospheric condition (the other links). These values are the mean values proposed by Choe et al. [36]. The probability of deterioration taking place after time $t = 20$ years is computed as

$$\begin{aligned} &P(d_i^1) = P(T_{\text{corr}} \le t) \text{ and} \\ &P(d_i^0) = 1 - P(d_i^1), \quad i = 1, 2, \cdots, 76 \end{aligned} \quad (50)$$

On the other hand, $P(I_i|D_i)$ for $i = 1, 2, \cdots, 76$ are given as

$$P(i_i^0|d_i^0) = 0.9, \quad P(i_i^1|d_i^0) = 0.1, \quad P(i_i^0|d_i^1) = 0.2, \quad \text{and } P(i_i^1|d_i^1) = 0.8 \quad (51)$$

The construction of $P(X_i|D_i, M, L)$ for $i = 1, 2, \cdots, 76$, follows the procedure illustrated in the numerical example of Lee et al. [25]. The failure probability of the links is computed based on the approximate bivariate seismic fragility model proposed by Huang et al. [37] along with the geometry and material properties of [37]. Using the fragility model, the probability that the drift capacity level of 4% is attained or exceeded, is computed as the failure probability of each link, given the values of two earthquake intensity measures, the normalized pseudo-spectral acceleration (PSA) PSA/g, and the normalized peak ground velocity (PGV) $PGV \cdot T_1/H_c$; therein g is the acceleration of the gravity, $H_c = 6.7056$ m is the height of the bridge column, and T_1 is the first mode period of the bridges, estimated respectively as 0.9616 and 1.0106 s for d_i^0 and d_i^1. Furthermore, the assignment of d_i^1 is assumed to result in 10% reduction in the reinforcement diameter while the full reinforcement area is considered for d_i^0. The intensity measures PSA and PGV that each bridge experiences given each assignment of $\{M, L\}$, are computed following [25].

5.2.2. Approximate inferences of probabilistic queries

For the deterministic construction of the CPM of $P(X_{77}|\boldsymbol{X}_c)$, 5,000 disjoint rules are searched by RDA. Based on this CPM and the ones illustrated in Section 5.2.1, all r.v.'s but X_{77} are marginalized for the query $P(x_{77}^0)$, and all r.v.'s other than X_{77}, X_{38}, and I_{38} for $P(x_{38}^0|x_{77}^0, i_{38}^1)$. Then, the bounds are computed as

$$\tilde{P}(x_{77}^0) \le P(x_{77}^0) \le 1 - \tilde{P}(x_{77}^1) \quad (52)$$

and

$$\frac{\tilde{P}(x_{38}^0, x_{77}^0, i_{38}^1)}{1 - \tilde{P}(\{(x_{77}^0, i_{38}^1)\}^c)} \le P(x_{38}^0|x_{77}^0, i_{38}^1) \le \frac{1 - \tilde{P}(\{(x_{38}^0, x_{77}^0, i_{38}^1)\}^c)}{\tilde{P}(x_{77}^0, i_{38}^1)} \quad (53)$$

For the evaluation of the joint probabilities in Eq. (53), we use the VE algorithm of Algorithm 3. The inference results – upper and lower bounds – of the two quantities are summarized in Table 2.

For stochastic construction of the CPM of $P(X_{77}|\boldsymbol{X}_c)$, 100,000 samples of $\boldsymbol{X}_c$ are generated by sampling X_i for $i = 1, 2, \cdots, 76$ independently from the binomial distribution with probability 0.95 of taking value 1. x_{77}^0 is observed in 832 of those samples. This IS density is selected for illustrative purposes and is not optimized; we note that the selection of an appropriate IS density should be further investigated in the future.

Table 1 tabularizes the input variables to Algorithm 4 for estimating the queries other than $\boldsymbol{P}$ and $\boldsymbol{M}$, where $\boldsymbol{I} = \{I_1, I_2, \cdots, I_{76}\}$ and $\boldsymbol{D} = \{D_1, D_2, \cdots, D_{76}\}$. Thereby, as $\boldsymbol{E}$ is an empty set when evaluating $P(x_{77}^0)$, its estimation corresponds to the unnormalized IS, while $P(x_{38}^0|x_{77}^0, i_{38}^1)$ is estimated as the normalized IS. Computation of the normalizing $P(x_{77}^0, i_{38}^1)$ requires the CPMs of $P(X_i|D_i, M, L)$ for $i = 1, 2, \cdots, 76$, and $P(X_{77}|\boldsymbol{X}_c)$.

The inference results – the mean value and coefficient of variation (c.o.v.) of the estimates – are summarized in Table 2, and they agree with the results evaluated both by the deterministic approach and by direct Monte Carlo Simulation (MCS) with 1,000,000 samples. The latter performs better in the unconditional case (which is to be expected given the higher number of samples), but worse in the conditional case where MCS is not efficient as many samples are rejected.

Table 1
Input variables of Algorithm 4 for the Sioux Falls network example.

Query	f	$\boldsymbol{E} = \boldsymbol{e}$	$\prec$ on $\boldsymbol{Z}$
$P(x_{77}^0)$	$\mathrm{II}[x_{77}^0]$	$\emptyset$	$\boldsymbol{I} \prec \boldsymbol{D} \prec M \prec L \prec \boldsymbol{X}_c$
$P(c_{38}^0|x_{77}^0, i_{38}^1)$	$\mathrm{II}[(c_{38}^0, x_{77}^0, i_{38}^1)]$	$\{X_{77}, I_{38}\} = (x_{77}^0, i_{38}^1)$	$\boldsymbol{I} - \{I_{38}\} \prec \boldsymbol{D} \prec M \prec L \prec \boldsymbol{X}_c - \{X_{38}\}$

Table 2
Inference results with the non-exhaustive CPM of $\boldsymbol{P}(\boldsymbol{X}_{77}|\boldsymbol{X}_c)$.

Methods to build CPM $P(X_{77}\|\boldsymbol{X}_c)$	$P(x_{77}^0)$	$P(x_{38}^0\|x_{77}^0, i_{38}^1)$
RDA: [lower, upper bounds]	[1.609, 1.612] $\times$ 10^{-3}	[0.6909, 0.7086]
Sampling from binomial distribution: mean (c.o.v.)	1.808 $\times$ 10^{-3} (0.08882)	0.7052 (1.390 $\times$ 10^{-3})
Monte Carlo Simulation: mean (c.o.v.)	1.593 $\times$ 10^{-3} (0.02504)	0.6889 (0.04090)

It is noted that 5,000, 100,000, and 1,000,000 rules are utilized to quantify the CPM of $P(X_{77}|\boldsymbol{X}_c)$ respectively for deterministic inference, IS, and MCS, while other CPMs remain identical. Since the number of rules is the parameter that controls the computational cost of inference as discussed in Section 4.1, the results suggest that the deterministic approach is more efficient in this example. It is expected that this is the case whenever there exists an efficient way to extract context-specific independence of component events in describing a system event, i.e. when the CPM of a system event can be constructed using the "−1" state. The RDA can identify such a construction in a connectivity problem, as demonstrated in this example. However, for general problems, e.g. delayed time due to congestion in transportation networks, this construction might not be available, in which case sampling methods are the only alternative.

6. Concluding remarks

In this paper, a matrix-based Bayesian Network (MBN) has been developed for efficient memory storage and flexible inference in modeling and evaluation of discrete Bayesian Networks (BNs). The MBN unifies previous research efforts to achieve efficiency in BN applications and is compatible with standard BN techniques. In MBN, the concept of conditional probability matrix (CPM) is introduced to quantify the probability mass functions (PMFs.) In CPMs, the assignments and probabilities are separately stored as matrices. CPMs can provide more compact and straightforward representation of PMFs by: (1) excluding the outcomes that have no effect on inference results from the storage, and (2) using "−1" state to denote the context-specific independence. In addition, the matrix form of the data structure in the MBN enables efficient and general implementations. With the operations and algorithms proposed to facilitate MBN-based inference, the MBN can replace the role of conditional probability tables in existing BN methodologies. Another distinct feature of the MBN is that CPMs do not require identifying the exhaustive set of events by either deterministic or stochastic algorithms. Thereby, the MBN enables general applications of BNs to approximate inferences especially as multiple non-exhaustive sets of events can be systematically taken into account. A simple reliability block diagram has been investigated to illustrate the procedure of modeling and inference using the MBN, and to compare the MBN with existing methodologies. Furthermore, we apply the MBN to two examples of large-size systems, which demonstrate its applicability and efficiency to such systems.

With its efficient modeling and flexible application, the MBN can model and analyze more complex and larger real-world systems than existing BN approaches. Since the BN methodology in general can achieve intuitive models of complex systems through its graphical representation and provide a systematic procedure for inference, such extension is expected to expand the BN-based probabilistic modeling and analysis to a broader range of civil systems. The MBN can be expanded to applications with system performance functions other than network connectivity, e.g. structural systems and flow of networks, or can be combined with advanced methodology for identifying optimal IS density. The expansion to other types of inferences, e.g. the evaluation of the maximum a posteriori estimator, will facilitate a wider range of queries on real-world civil systems to support decision-making, e.g. on strengthening bridges against seismic risk. However, the BN size is still a bottleneck in quantifying and inferencing BN, which is implied in Fig. 10 by the increasing number of required subsets as the number of links increases. The inference methodology can be further developed to address such limitation and handle even larger systems.

Acknowledgement

This research was supported by a grant (19SCIP-B146946-02) from Smart Civil Infrastructure Research Program funded by Ministry of Land, Infrastructure and Transport (MOLIT) of Korea government and the National Research Foundation of Korea (NRF) - Deutsche Forschungsgemeinschaft (DFG) Collaborative Research Program through grant STR 1140/5-1. The fourth author is supported by the Institute of Construction and Environmental Engineering at Seoul National University. These supports are gratefully acknowledged.

References

[1] Bruneau M, Chang SE, Eguchi RT, Lee GC, O'Rourke TE, Reinhorn AM, Shinozuka M, Tierney K, Wallace WA, von Winterfeldt D. A framework to quantitatively assess and enhance the seismic resilience of communities. Earthquake Spectra 2003;19(4):733–52.
[2] Bensi M, Der Kiureghian A, Straub D. Framework for post-earthquake risk assessment and decision making for infrastructure systems. ASCE-ASME J Risk Uncertainty Eng Syst Part A 2014;1(1):04014003.
[3] Gehl P, Cavalieri F, Franchin P. Approximate Bayesian network formulation for the rapid loss assessment of real-world infrastructure systems. Reliab Eng Syst Saf 2018;177(1):80–93.
[4] Fenton N, Neil M. Risk assessment and decision analysis with Bayesian Networks. Croydon: CRC Press; 2013.
[5] Bensi M, Der Kiureghian A, Straub D. Efficient Bayesian network modeling of systems. Reliab Eng Syst Saf 2013;112(1):200–13.
[6] Boutilier C, Friedman N, Goldszmidt M, Koller D. Context-specific independence in Bayesian networks. 12th international conference on Uncertainty in artificial intelligence. 1996. p. 115–23.
[7] Koller D, Friedman N. Probabilistic graphical models: principles and techniques. Cambridge: The MIT Press; 2009.
[8] Darwiche A. A differential approach to inference in Bayesian networks. J ACM 2003;50(3):280–305.
[9] Barber D. Bayesian reasoning and machine learning. Cambridge: Cambridge University Press; 2012.
[10] Tien I, Der Kiureghian A. Algorithms for Bayesian network modeling and reliability assessment of infrastructure systems. Reliab Eng Syst Saf 2016;156(1):134–47.
[11] Heckerman D, Breese JS. Causal independence for probability assessment and inference using Bayesian networks. IEEE Trans Syst Man Cybern-Part A 1996;26(6):826–31.
[12] Cavalieri F, Franchin P, Gehl P, D'Ayala D. Bayesian networks and infrastructure systems: computational and methodological challenges. In: Gardoni P, editor. Risk and reliability Analysis: theory and applications. Berlin: Springer; 2017. p. 385–415.
[13] Jane C-C, Laih Y-W. A dynamic bounding algorithm for approximating multi-state two-terminal reliability. Eur J Oper Res 2010;205(3):625–37.
[14] Lee Y-J, Song J. Risk analysis of fatigue-induced sequential failures by branch-and-bound method employing system reliability bounds. J Eng Mech 2011;137(12):807–21.
[15] Li J, He J. A recursive decomposition algorithm for network seismic reliability evaluation. Earthquake Eng Struct Dyn 2002;31(8):1525–39.
[16] Lim H-W, Song J. Efficient risk assessment of lifeline networks under spatially correlated ground motions using selective recursive decomposition algorithm. Earthquake Eng Struct Dyn 2012;41(13):1862–82.
[17] Au S-K, Beck JL. Estimation of small failure probabilities in high dimensions by subset simulation. Probab Eng Mech 2001;16(4):263–77.
[18] Kurtz N, Song J. Cross-entropy-based adaptive importance sampling using Gaussian mixture. Struct Saf 2013;42(1):35–44.
[19] Neal RM. Probabilistic inference using Markov chain Monte Carlo methods. Toronto: University of Toronto; 1993. Technical Report CRG-TR-93-1.
[20] Straub D, Papaioannou I. Bayesian updating with structural reliability methods. J Eng Mech 2014;141(3):04014134.
[21] Gehl P, Cavalieri F, Franchin P, Negulescu C. Robustness of a hybrid simulation-based/Bayesian approach for the risk assessment of a real-world road network. Proc. of the 12th International Conference on Structural Safety and Reliability. 2017.
[22] Zwirglmaier K, Song J, Papaioannou I, Straub D. Application of hybrid Bayesian networks to large infrastructure problems. J Pap Manuscript 2018.
[23] Kang W-H, Song J, Gardoni P. Matrix-based system reliability method and applications to bridge networks. Reliab Eng Syst Saf 2008;93(11):1584–93.
[24] Song J, Kang W-H. System reliability and sensitivity under statistical dependence by matrix-based system reliability method. Struct Saf 2009;31(2):148–56.
[25] Lee YJ, Song J, Gardoni P. Post-hazard flow capacity of bridge transportation

network considering structural deterioration of bridges. Struct Infrastruct Eng 2011;7(7):509–21.
[26] Vesely WE, Goldberg FF, Roberts NH, Haasl DF. Fault tree handbook. Washington DC: Nuclear Regulatory Commission; 1981.
[27] Gondran M. System reliability: evaluation and prediction in engineering. Berlin: Springer; 1986.
[28] Wolsey LA. Integer programming. Newsery: Wiley; 1998.
[29] Jane CC, Laih YW. A practical algorithm for computing multi-state two-terminal reliability. IEEE Trans Reliab 2008;57(2):295–302.
[30] Gelfand AE, Smith AFM. Sampling-based approaches to calculating marginal densities. J Am Stat Assoc 1990;85(410):398–409.
[31] Pearl J. Evidential reasoning using stochastic simulation of causal models. Artif Intell 1987;32(2):245–57.
[32] Zhang NL, Poole D. Exploiting causal independence in Bayesian network inference. J Artif Intell Res 1996;5(1):301–28.
[33] Poole D. Probabilistic conflicts in a search algorithm for estimating posterior probabilities in Bayesian networks. Artif Intell 1996;88(1):69–100.
[34] Albert-László B. The physics of the Web. Phys World 2001;14(7):33.
[35] Cosentino P, Ficarra V, Luzio D. Truncated exponential frequency-magnitude relationship in earthquake statistics. Bull Seismol Soc Am 1977;67(6):1615–23.
[36] Choe D-E, Gardoni P, Rosowsky D, Haukaas T. Probabilistic capacity models and seismic fragility estimates for RC columns subject to corrosion. Reliab Eng Syst Saf 2008;93(3):383–93.
[37] Huang Q, Gardoni P, Hurlebaus S. Probabilistic seismic demand models and fragility estimates for reinforced concrete highway bridges with one single-column bent. J Eng Mech 2010;136(11):1340–53.