Sensitivity Analysis of a Bayesian Network

Chenzhao Li
Department of Civil
and Environmental Engineering,
Vanderbilt University,
2301 Vanderbilt Place PMB 351826,
Nashville, TN 37235
e-mail: chenzhao.li@vandebilt.edu

Sankaran Mahadevan[1]
Department of Civil
and Environmental Engineering,
Vanderbilt University,
2301 Vanderbilt Place PMB 351826,
Nashville, TN 37235
e-mail: sankaran.mahadevan@vandebilt.edu

In a Bayesian network (BN), how a node of interest is affected by the observation at another node is a main concern, especially in backward inference. This challenge necessitates the proposed global sensitivity analysis (GSA) for BN, which calculates the Sobol' sensitivity index to quantify the contribution of an observation node toward the uncertainty of the node of interest. In backward inference, a low sensitivity index indicates that the observation cannot reduce the uncertainty of the node of interest, so that a more appropriate observation node providing higher sensitivity index should be measured. This GSA for BN confronts two challenges. First, the computation of the Sobol' index requires a deterministic function while the BN is a stochastic model. This paper uses an auxiliary variable method to convert the path between two nodes in the BN to a deterministic function, thus making the Sobol' index computation feasible. Second, the computation of the Sobol' index can be expensive, especially if the model inputs are correlated, which is common in a BN. This paper uses an efficient algorithm proposed by the authors to directly estimate the Sobol' index from input–output samples of the prior distribution of the BN, thus making the proposed GSA for BN computationally affordable. This paper also extends this algorithm so that the uncertainty reduction of the node of interest at given observation value can be estimated. This estimate purely uses the prior distribution samples, thus providing quantitative guidance for effective observation and updating.
[DOI: 10.1115/1.4037454]

Keywords: global sensitivity analysis, Bayesian network, Sobol' index, inference, uncertainty reduction

1 Introduction

During the past 30 years, the Bayesian network (BN) has become a key method for representation and reasoning under uncertainty in the fields of engineering [1,2], machine learning [3,4], artificial intelligence [5,6], etc. In a BN, random variables are denoted by nodes and their dependence relationships are denoted by directed arcs (arrows). An arc (arrow) between two nodes indicates that the distribution of the downstream node (child node) is conditioned on the value of the upstream node (parent node), i.e., the arc is associated with a conditional probability distribution (CPD). The entire BN represents the joint probability distribution of the random variables. As long as the BN has been established, two tasks can be pursued: (1) calculate the distribution of a downstream node based on the given observation of an upstream node, i.e., forward propagation and (2) estimate the posterior distribution of an upstream node at the given observation of a downstream node, i.e., backward inference. Usually, the purpose of backward inference is to reduce the uncertainty of the variable (node) of interest, as shown in Fig. 1.

We can see that both the forward propagation and the backward inference concern that whether fixing a node would affect the distribution of another node. And this concern actually requires a sensitivity analysis: in forward propagation, if the node of interest has a low sensitivity with respect to an input node (an upstreaming node which causes uncertainty in the node of interest), we can be simply fix this input node to achieve dimension reduction without affecting the resultant distribution of the node of interest; and in backward propagation, if the node of interest has a low sensitivity with respect to an observation node, then this observation node cannot effectively reduce the uncertainty of the node of interest (as shown in Fig. 1(*b*)); thus, we should switch to observe a more appropriate node leading to higher sensitivity. Note that this sensitivity analysis is desired before conducting the forward propagation and/or the backward inference.

Various sensitivity analysis methods have been developed in the literature [7], and this paper selected the prominent variance-based Sobol' index [8–13], one of the "global sensitivity analysis" (GSA) method that considers the entire distribution of the inputs. The Sobol' index includes first-order and higher-order indices. While the higher-order index requires uncorrelated model inputs, Saltelli and Tarantola [14] pointed out that the first-order index is still an informed choice to rank the importance of correlated model inputs. This paper focuses on calculating the first-order index S_i since the correlation among the nodes of a BN is very common. In a BN, the correlation between any two nodes can be established by the arcs (CPDs) between them. In other words, if the nodes are independent to each other, then they cannot even construct a BN. Consider a computational model in the form of $y = f(\boldsymbol{x})$ where $\boldsymbol{x} = \{x_1, \ldots, x_k\}$ is the vector of stochastic model inputs. The first-order Sobol' index $S_i (i = 1, 2, \ldots, k)$ is

$$S_i = \frac{V_{x_i}\left(E_{\boldsymbol{x}_{-i}}(y|x_i)\right)}{V(y)} = 1 - \frac{E_{x_i}\left(V_{\boldsymbol{x}_{-i}}(y|x_i)\right)}{V(y)} \tag{1}$$

where $\boldsymbol{x}_{-i}$ means all the model inputs other than x_i, and $V(\cdot)$ means the variance, and $E(\cdot)$ means the mean value. For the numerator of the middle term of Eq. (1), $V_{\Phi'}(y) = E_{\Phi'}(V_{\boldsymbol{x}_{-i}}(y|x_i)) + V_{\Phi'}(E_{\boldsymbol{x}_{-i}}(y|x_i))$ denotes the mean value of y across different values of $\boldsymbol{x}_{-i}$ at given value of x_i, and $V_{x_i}(\cdot)$ means the variance across different values of x_i. Similarly, for the numerator of the right term of Eq. (1), $V_{\boldsymbol{x}_{-i}}(y|x_i)$ denotes the variance of y across different values of $\boldsymbol{x}_{-i}$ at given value of x_i, and $E_{x_i}(\cdot)$ means the variance across different values of x_i. Furthermore, the middle term and the right term are equal due to the law of total variance $V(y) = V_{x_i}(E_{\boldsymbol{x}_{-i}}(y|x_i)) + E_{x_i}(V_{\boldsymbol{x}_{-i}}(y|x_i))$.

S_i quantifies the contribution of input x_i by itself to the uncertainty in output y. Specifically, the last term of Eq. (1) also indicates that S_i is the average ratio of the output variance reduction by fixing x_i. For example, $S_i = 0.3$ means that the variance of y will reduce by 30% on average after fixing x_i.

The proposed GSA for BN aims to calculate the first-order Sobol' index to quantity the contribution of node X_1 toward the

[1]Corresponding author.

Manuscript received August 31, 2016; final manuscript received January 27, 2017; published online September 7, 2017. Assoc. Editor: Yan Wang.

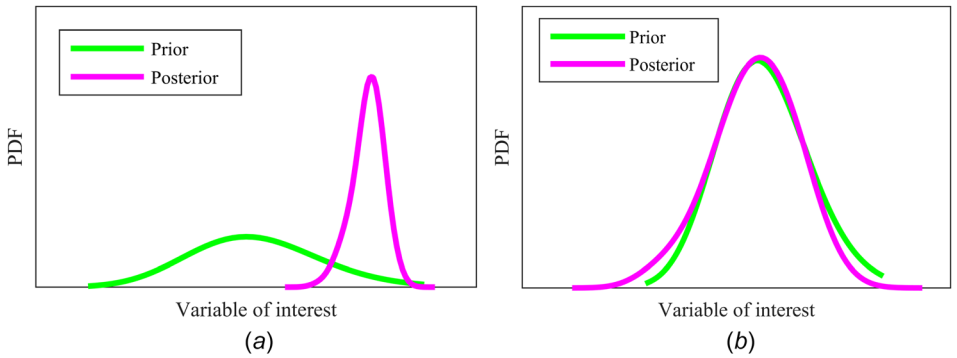


Fig. 1 Possible Bayesian inference results: (*a*) desired inference and (*b*) undesired inference

uncertainty of the node of interest X_N. (As mentioned earlier, X_1 denotes an input node in the case of forward propagation, and an observation node in the case backward inference). However, this GSA for a BN confronts two challenges of feasibility and affordability. First, the computation of the Sobol' index requires a deterministic function [15]. The term "deterministic function" means that if all the inputs are given specific values, the function outputs a single value. It does not matter whether the function is closed-form or implicit. But a unique input to the model gives a unique value of the output, and the variance-based Sobol' index [16] needs this assumption. In a previous paper, we have discussed the need for a deterministic function in the Sobol index computation [10]. Our paper emphasizes the term deterministic function to contrast from a "stochastic" prediction; in the latter case, the function output is probabilistic (i.e., it has many possible realizations) even if all inputs are fixed. A simple stochastic model is a Gaussian variable $X \sim N(\mu,\ \sigma^2)$ if we consider μ and σ as model inputs and X as model output. The BN is a stochastic model, i.e., the node of interest still has a distribution even at given values of other nodes. And the required deterministic function mapping X_1 (and some other variables) to the node of interest X_N is unestablished. Proof of the existence and the establishment of this deterministic function needs to be solved. Note that usually the Sobol' index is applied to deterministic function of continuous input/output, so that this paper only considers the continuous BN where all the nodes are continuous variables.

Second, the computation of the Sobol' index can be expensive even if the deterministic function is established. Based on Eq. (1), computing S_i analytically is nontrivial, since $E_{\boldsymbol{x}_{-i}}(\cdot)$ in Eq. (1) indicates a multidimensional integral. Computing S_i by Monte Carlo simulation (MCS) directly is also expensive. The numerator in Eq. (1) leads to a double-loop MCS [7]: the inner loop $E_{\boldsymbol{x}_{-i}}(y|x_i)$ computes the mean value of y using $\boldsymbol{X}_t \in \Re^m$ random samples of $\boldsymbol{x}_{-i}$; and the outer loop computes $V_{x_i}(E_{\boldsymbol{x}_{-i}}(y|x_i))$ by iterating the inner loop n_2 times at different values of x_i. Various algorithms have been proposed to reduce the cost in the computation of the first-order Sobol' index, categorized into analytical methods [17–19] and Monte Carlo methods [16,20–23]. However, all of these algorithms assume uncorrelated model inputs, while the nodes in a BN are usually correlated. Saltelli's paper [23] in 2002 mentioned that there is no alternative to the expensive double-loop MCS to compute S_i with correlated model inputs. This hurdle in computational cost is also to be solved.

The rest of the paper is organized as follows: Section 2 uses the auxiliary variable method to convert the path between node X_1 and node X_N to a deterministic function, thus making the Sobol' index computation feasible for a BN. Section 3.1 introduces an efficient algorithm to directly estimate the Sobol' index from Monte Carlo samples of the prior distribution of the BN, so that the proposed GSA for the BN is affordable. The resultant index is the averaged variance reduction ratio (VRR) of the node of interest across all possible values of the observation node. If a specific value of the observation node is given, the proposed method is extended to give a better estimate of variance reduction ratio at the given observation value. Section 4 illustrates the proposed method using two examples, including a time-independent static BN and a time-dependent dynamic BN.

Note: the sensitivity analysis in this paper is based on "current knowledge," which means the joint distribution of the Bayesian network. This paper uses the prior distribution samples to conduct the sensitivity analysis since that is our current knowledge before any observation and updating. Of course, after collecting an observation and subsequent calculation of the posterior distribution, we can also use the samples from the posterior distribution to calculate the posterior sensitivity index.

2 Feasibility of Global Sensitivity Analysis for a Bayesian Network

2.1 Auxiliary Variable Method. The auxiliary variable method was developed by Sankararaman and Mahadevan [24] to quantitatively distinguish the contributions of aleatory natural variability and epistemic distribution parameter uncertainty in a random variable x. The distribution of x is conditioned on the value of its distribution parameter $\boldsymbol{\theta}_x$. The uncertainty about the value of $\boldsymbol{\theta}_x$ is represented by a probability density $\pi(\boldsymbol{\theta}_x)$, and this uncertainty is epistemic. The aleatory uncertainty of x is denoted by a conditional distribution $\pi(x|\boldsymbol{\theta}_x)$, which changes as the value of $\boldsymbol{\theta}_x$ changes.

Based on the probability integral transform theorem [25], random sampling from the $\pi(x|\boldsymbol{\theta}_x)$ is realized in two steps: (1) define a variable U_x of standard uniform distribution $U(0,1)$ and generate its sample, which is taken as the cumulative density function (CDF) value of x, and (2) obtain a sample of x by inverting the conditional CDF $\mathcal{F}(x|\boldsymbol{\theta}_x)$, i.e.,

$$x = \mathcal{F}^{-1}(U_x|\boldsymbol{\theta}_x) \tag{2}$$

Equation (2) is a deterministic function $f: (U_x, \boldsymbol{\theta}_x) \rightarrow x$, since a sample of $\boldsymbol{\theta}_x$ and a sample of U_x lead to a single value of x. At a given value of $\boldsymbol{\theta}_x$, the sample of U_x and the sample of x have a one-to-one mapping, i.e., a single value of x is determined once the value of U_x is decided. Thus the natural variability in x is represented by U_x. This standard uniform random variable U_x, which is the CDF value of $\pi(x|\boldsymbol{\theta}_x)$, is named as the auxiliary variable.

Although the use of auxiliary variable is a standard procedure in sampling random variables, generally it is used implicitly and only the resultant samples of the random variables are recorded and utilized. However, if we use this auxiliary variable explicitly, it makes the GSA for BN possible. This will be explained in the remainder of this section.

2.2 Deterministic Function for a Directed Path in Bayesian Network. The auxiliary variable method have been extended to any variable whose distribution is conditioned on other variables

[10,26], i.e., to any CPD in the BN. Assume that the distribution of a random variable C depends on the value of two other random variables A and B by a CPD $\pi(C|A,B)$. Then the variability in $\pi(C|A,B)$ can be captured by a single auxiliary variable U_C, which is the CDF value of $\pi(C|A,B)$. Thus, the uncertainty in variable C is caused by two components: (1) the uncertainty due to the parent nodes A,B; and (2) the uncertainty expressed by the CPD at given values of A and B. The introduced auxiliary variable captures the later part. (With reference to Sec. 2.1, A and B are analogous to distribution parameters for the random variable C). As shown in Fig. 2 (A), (B), and (C) constitute a simple BN. The introduced auxiliary variable U_C converts C to be a deterministic node, which means the value of C is fixed once the value of its parent nodes $\{A,B,U_C\}$ is given. Finally, this auxiliary variable build a deterministic function $C=\mathcal{F}^{-1}(U_C|A,B)$, where $\mathcal{F}^{-1}(\cdot)$ is the inverse CDF of the CPD $\pi(C|A,B)$.

The auxiliary variable method can be further extended to a directed path $X_1 \rightarrow X_2 \ldots \rightarrow X_N$ in a BN. Here, X_N is the node of interest, and the objective is to compute the sensitivity index of X_1 to decide whether X_N is sensitive to it. An example of the directed path is $A \rightarrow C \rightarrow E \rightarrow G$ in Fig. 3. As shown in Fig. 4, by introducing auxiliary variables to each CPD in this directed path, a deterministic function mapping X_1 to X_N is established

$$\begin{cases} X_2 = \mathcal{F}^{-1}(U_{X_2}|Pa'_{X_2}, X_1) \\ X_3 = \mathcal{F}^{-1}(U_{X_3}|Pa'_{X_3}, X_2) \\ \cdots \\ X_N = \mathcal{F}^{-1}(U_{X_N}|Pa'_{X_N}, X_{N-1}) \end{cases} \tag{3}$$

where $\mathcal{F}^{-1}(U_{X_i}|Pa'_{X_i},\ X_{i-1})$ for $i=2$ to N is the inverse CDF of the CPD $\pi(X_i|Pa'_{X_i},X_{i-1})$, and U_{X_i} is the auxiliary variable introduced for this CPD, and Pa'_{X_i} represents the parent nodes of X_i that are not in this path (Note that another notation Pa_V is used later, which means all the parents node of V, i.e., $Pa_{X_i}=\{Pa'_{X_i},X_{i-1}\}$ in Fig. 4. The inputs of Eq. (3) are $\{X_1,P'_{X_i},U_{X_i}\}$ for $i=2$ to N, thus Eq. (3) can be also denoted as a deterministic function $f:\{X_1,P'_{X_i},U_{X_i}\}\rightarrow X_N$.

The deterministic function established in Eq. (3) can be illustrated by a simple BN in Fig. 3. For the directed path $A \rightarrow C \rightarrow E \rightarrow G$, thus the deterministic function mapping A to G is

$$\begin{cases} C = \mathcal{F}^{-1}(U_C|A,B) \\ E = \mathcal{F}^{-1}(U_E|C,D,F) \\ G = \mathcal{F}^{-1}(U_G|E,F) \end{cases} \tag{4}$$

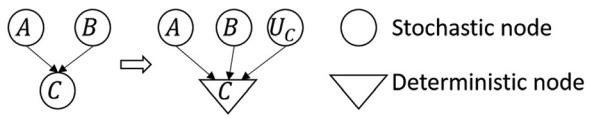


Fig. 2 Auxiliary variable for a CPD

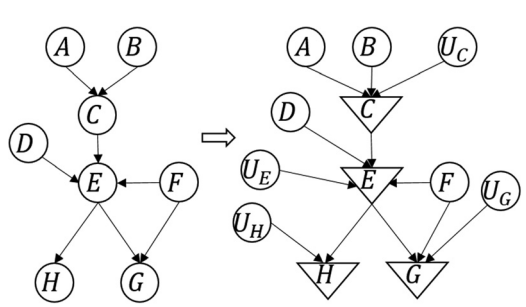


Fig. 3 Auxiliary variable for a BN

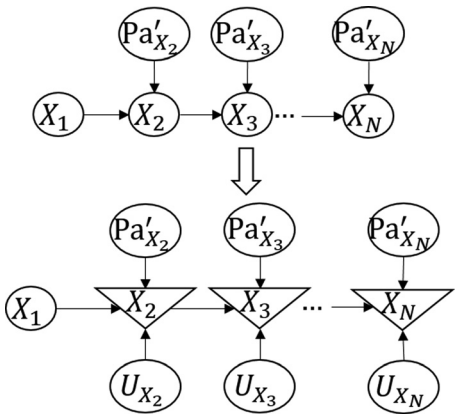


Fig. 4 Deterministic function for the path $X_1 \rightarrow X_N$

2.3 Deterministic Function for a Trail in Bayesian Network. The directed path from X_1 to X_N in Sec. 2.2 requires that all the arcs are directed toward X_N, which may NOT be satisfied. It is not usually that some arrows are toward X_1. In backward inference, the node of interest X_N is usually an upstream node (probably root node) while the observation node is usually a downstream node. Thus, the path between them turns to be $X_1 \leftarrow X_2 \leftarrow \ldots \leftarrow X_N$, where all the arrows are toward X_1. In this case, the function $f: X_1 \rightarrow X_N$ will be still missing.

To solve this problem, let consider a general case named "trail." A trail $X_1 - X_2 - \ldots - X_N$ (where the arc " $-$ " is still directed, either " $\rightarrow$ " or " $\leftarrow$ ") only requires all the adjacent nodes in the path are connected by arcs, regardless of the direction of the arcs. The deterministic function established in Eq. (3) for the directed path can be also extended to the trail based on the theorem of arc reversal [27].

THEOREM 1 ARC REVERSAL. *Given that there is an arc (V_1,V_2) from node V_1 to node V_2, but no other directed path from V_1 to V_2, arc (V_1,V_2) can be replaced by arc (V_2,V_1). Afterward, both nodes inherit each other's parent nodes.*

This theorem is illustrated in Fig. 5. Here, Pa_{V_1} indicates the parent nodes of V_1, and Pa_{V_2} indicates the parent nodes of V_2. In addition, Pa_{V_1} Pa_{V_2} are the nodes which are the parents of V_1 but not the parents of V_2, and correspondingly Pa_{V_2} Pa_{V_1} are the nodes which are the parents of V_2 but not the parents of V_1; and $Pa_{V_1} \cap Pa_{V_2}$ are the shared parents of V_1 and V_2. Figure 5 shows that after reversing the arc between V_1 and V_2, extra arcs $(Pa_{V_1}\ Pa_{V_2},V_2)$ and $(Pa_{V_2}\ Pa_{V_1},V_1)$ are also derived based on Ref. [27] and added the new BN to guarantee that the new BN after arc reversal is mathematically equivalent to the original BN. The CPDs also need to be redefined, and the derivation of the new CPDs can be also found in Ref. [27]. However, note that the

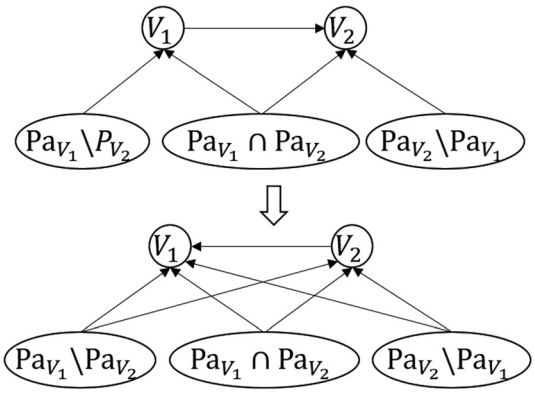


Fig. 5 Arc reversal

proposed method in this paper do NOT need to derive these new CPDs. The main focus of this section is to illustrate the possibility of arc reversal and prove the existence of the deterministic function mapping X_1 to X_N even if the path between them is undirected.

With respect to the trail between X_1 and X_N, Theorem 1 proves that the arc (X_i, X_j) between two adjacent nodes X_i and X_j ($i = j+1$ *or* $j-1$ so that they are adjacent) can be reversed, as long as there is no other directed path from X_i to X_j. If all the arcs toward X_1 can be reversed, this trail will be converted to a directed path from X_1 to X_N so that a deterministic function mapping X_1 to X_N exists based on Eq. (3). In Fig. 3, the trail $H \leftarrow E \rightarrow G$ can be converted to a directed path $H \rightarrow E \rightarrow G$ by reserving the arc $(E,\ H)$; then, a deterministic function mapping H to G can be constructed using auxiliary variables.

The arc reversal makes the sensitivity analysis regarding the backward Bayesian inference possible. In this case, we aim to compute the Sobol' index of observation node X_1 regarding an inference node of interest X_N so that we can quantify the variance reduction of X_N by fixing node X_1 at an observation value. The existing path in the Bayesian network is $X_1 \leftarrow X_2 \leftarrow \ldots \leftarrow X_N$, then the arc reversal enables us to reverse all the arcs to build the required deterministic function.

It is arc reversal that makes the sensitivity analysis regarding Bayesian inference possible. In this case, we aim to compute the sensitivity of root node X_N with respect to the observation node X_1 so that we can quantify the uncertainty reduction of X_N by fixing node X_1 at an observation value. We need a directed path from X_1 to X_N. However, the existing path in the Bayesian network is always directed from root node X_N toward the observation X_1, so that we must reverse all the arcs to build the required deterministic function.

2.4 Global Sensitivity for a Path in the Bayesian Network. For two nodes X_1 and X_N in the BN, Secs. 2.1–2.3 explained the possibility to build a deterministic function mapping X_1 to X_N via: (1) the directed path or trail between them, (2) the theorem of arc reversal, and (3) the introduction of auxiliary variables. Thus, we can conduct the GSA on Eq. (3) and compute the first-order Sobol' index S_{X_1} for X_1. As explained below in Eq. (1), S_{X_1} is the average ratio of the reduced variance of X_N by fixing X_1. In backward inference, a low value S_{X_1} indicates that measuring X_1 to reduce the uncertainty of X_N is just a waste of effort, and another node of higher sensitivity index would be more appropriate.

However, the computation of S_{X_1} is nontrivial. First, building the deterministic function explicitly can be complicated for both directed and trail. For the directed path $X_1 \rightarrow \ldots \rightarrow X_N$, the effort to build the deterministic function in Eq. (3) becomes intensive if the path is long so that many nodes and auxiliary variables will be involved. For the trail, more efforts to modify the structure of the BN and derive new CPDs are required. In backward inference, usually the observation node is downstream and the node of interest is the upstream; thus, the corresponding path is $X_1 \leftarrow \ldots \leftarrow X_N$. To build the required deterministic function mapping X_1 to X_N, all arcs in the path need to be reversed, which brings intensive computational efforts.

Second, even with the deterministic function established, calculating the sensitivity index also needs intensive effort. The inputs of the deterministic function include the nodes in the BN, so the correlation between them is very common. As mentioned in Sec. 1, since current efficient algorithms for Sobol' index usually require uncorrelated inputs, the expensive double-loop MCS is the only choice.

Actually, the purpose of this section is to mathematically prove the existence of a deterministic function between two nodes (regardless of the computational effort), so that the Sobol' index, which requires the deterministic function, is applicable to quantify the sensitivity for the Bayesian network. The Sobol' index computation will be realized by an algorithm proposed by the authors. Details of this algorithm can be found in Ref. [28], and a brief introduction is given in Sec. 3.1. This algorithm directly extracts the first-order Sobol' index from the input–output samples. Using this algorithm, the Sobol' index of the observation node quantifying its contribution toward the uncertainty of the node of interest can be computed purely using their samples generated from the joint prior distribution of the BN, where the computation of the inference is NOT needed. In detail, we generate samples of the root nodes from their prior distributions and then generate samples of nonroot nodes based on the samples of their parent nodes and the CPDs. Therefore, explicitly establishing the deterministic function is not necessary, and the expensive double-loop MCS method is also avoided. Section 3.2 extends this algorithm to a better variance reduction estimate at a specific observation value.

3 Sobol' Index Computation and Variance Reduction at Given Observation

3.1 Directly Estimate the First-Order Sobol' Index From Input–Output Samples. Consider a deterministic function of $y = f(\boldsymbol{x})$ where $\boldsymbol{x} = \{x_1, \ldots, x_k\}$. This algorithm is regarding first-order Sobol' index expression of $S_i = 1 - E_{x_i}(V_{\boldsymbol{x}_{-i}}(y|x_i))/V(y)$, whose numerator $E_{x_i}(V_{\boldsymbol{x}_{-i}}(y|x_i))$ implies an expensive double-loop Monte Carlo simulation.

First we illustrate stratified sampling, which generates samples in equal probability intervals to represent the distribution of a random variable x_i. Figure 6(*a*) shows one strategy [7] of stratified sampling: (1) divide the CDF of x_i into M intervals such that these intervals have the same length and (2) generate one sample u^l (the red dots in Fig. 6(*a*), and $l = 1$ to M) from each CDF interval and obtain samples of x_i (the green dots in Fig. 6) by CDF inversion $x_i^l = P^{-1}(u^l)$, where $P^{-1}(\cdot)$ is the inverse CDF of x_i. If we take the bounds of these intervals of the CDF as the inputs of $P^{-1}(\cdot)$, the sampling space of x_i is actually divided into M equally probable intervals $\Phi^l (l = 1$ to $M)$, as shown in Fig. 6, and x_i^l is actually a random sample generated within Φ^l. In this paper we also denote $\boldsymbol{\Phi} = \{\Phi^1, \Phi^2, \ldots, \Phi^M\}$.

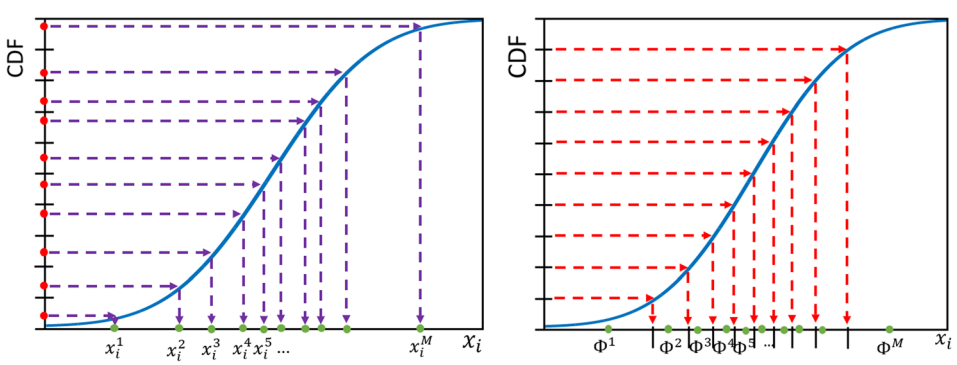


Fig. 6 Equally probable interval

$V_{\boldsymbol{x}_{-i}}(y|x_i)$ is a function of x_i. Based on the extreme value theorem, $V_{\boldsymbol{x}_{-i}}(y|x_i)$ must have a maximum value and a minimum value in Φ^l. The mean value of $V_{\boldsymbol{x}_{-i}}(y|x_i)$, i.e., $E_{\Phi^l}(V_{\boldsymbol{x}_{-i}}(y|x_i))$ for $x_i \in \Phi^l$, is between the maximum and minimum values

$$\min_{x_i \in \Phi^l}(V_{\boldsymbol{x}_{-i}}(y|x_i)) \le E_{\Phi^l}(V_{\boldsymbol{x}_{-i}}(y|x_i)) \le \max_{x_i \in \Phi^l}(V_{\boldsymbol{x}_{-i}}(y|x_i)) \tag{5}$$

Then the intermediate value theorem proves that

$$\exists x_i^{\#} \in \Phi^l \text{ s.t. } V_{\boldsymbol{x}_{-i}}\left(y|x_i^{\#}\right) = E_{\Phi^l}(V_{\boldsymbol{x}_{-i}}(y|x_i)) \tag{6}$$

With $\boldsymbol{X}_t \in \Re^m$, the law of total variance is

$$V_{\Phi^l}(y) = E_{\Phi^l}(V_{\boldsymbol{x}_{-i}}(y|x_i)) + V_{\Phi^l}(E_{\boldsymbol{x}_{-i}}(y|x_i)) \tag{7}$$

where the subscript Φ^l means all the terms are constrained to $x_i \in \Phi^l$. Substituting Eq. (7) into Eq. (6), we can have

$$\exists x_i^{\#} \in \Phi^l \text{ s.t.} V_{x_{-i}}\left(y|x_i^{\#}\right) = V_{\Phi^l}(y) - V_{\Phi^l}(E_{x_{-i}}(y|x_i)) \tag{8}$$

where $x_i^{\#}$ is an unknown but existing point in Φ^l. Note that now $V_{\Phi^l}(y)$ is the variance of y given $x_i \in \Phi^l$ and $V_{\Phi^l}(E_{\boldsymbol{x}_{-i}}(y|x_i))$ is the variance of $E_{\boldsymbol{x}_{-i}}(y|x_i)$ given $x_i \in \Phi^l$.

The last term in Eq. (8) can be rewritten as

$$V_{\Phi^l}(y) - V_{\Phi^l}\left(E_{\boldsymbol{x}_{-i}}(y|x_i)\right) = \left(1 - \frac{V_{\Phi^l}\left(E_{\boldsymbol{x}_{-i}}(y|x_i)\right)}{V_{\Phi^l}(y)}\right) V_{\Phi^l}(y) \tag{9}$$

In Eq. (9), the term $V_{\Phi^l}(E_{\boldsymbol{x}_{-i}}(y|x_i))/V_{\Phi^l}(y) = S_i^{\Phi^l}$ is nothing but the first-order Sobol' index of x_i as it is restricted to the interval Φ^l. We always have $S_i^{\Phi^l} < S_i$ since the uncertainty of x_i has been reduced significantly by restricting it in Φ^l such that its Sobol' index will be much lower. And the value of $S_i^{\Phi^l}$ approaches zero as M increases so that the Φ^l is narrowed. Then we have $V_{\Phi^l}(y) - V_{\Phi^l}(E_{\boldsymbol{x}_{-i}}(y|x_i)) \approx V_{\Phi^l}(y)$ so that Eq. (8) changes to

$$V_{\boldsymbol{x}_{-i}}\left(y|x_i^{\#}\right) \approx V_{\Phi^l}(y) \tag{10}$$

In the formula of $S_i = 1 - E_{x_i}(V_{\boldsymbol{x}_{-i}}(y|x_i))/V(y)$, the outer loop $E_{x_i}(\cdot)$ requires fixing x_i at different locations, and these selected locations are samples from the distribution of x_i. Based on stratified sampling, the set of these unknown but existing points $\boldsymbol{x}_i^{\#} = \{x_i^{1\#}, \ldots, x_i^{M\#}\}$ from the equally probable intervals $\boldsymbol{\Phi} = \{\Phi^1, \ldots, \Phi^M\}$ can represent the distribution of x_i. Based on Eq. (10), computation of S_i is expressed as

$$S_i = 1 - \frac{E_{\boldsymbol{\Phi}}\left(V_{\boldsymbol{x}_{-i}}\left(y|x_i^{\#}\right)\right)}{V(y)} \approx 1 - \frac{E_{\boldsymbol{\Phi}}\left(V_{\Phi^l}(y)\right)}{V(y)} \tag{11}$$

where $\Phi^l (l = 1 \text{ to } M)$ then Eq. (11) can be realized by the following steps:

(1) Generate n random samples of $\boldsymbol{x}$;
(2) Obtain corresponding values of y by evaluating $y = f(\boldsymbol{x})$, and estimate $V(y)$ using all samples of y;
(3) Divide the domain of x_i into M equally probable intervals, as shown in Fig. 6;
(4) Assign the samples of y into divided intervals based on one-to-one mapping between the samples of x_i and samples of y;
(5) Estimate $V_{\Phi^l}(y)$ as the sampling variance of y in each interval;
(6) Estimate $E_{\boldsymbol{\Phi}}(V_{\Phi^l}(y))$ as the sampling mean of t in step 5;
(7) The first-order index is $S_i = 1 - E_{\boldsymbol{\Phi}}(V_{\Phi^l}(y))/V(y)$.

The steps to realize the proposed method are also illustrated in Fig. 7, where samples in different equally probable intervals are represented in different colors. Step 1 in Fig. 7 shows that the proposed algorithm can directly estimate S_i from input–output samples without rerunning the underlying model. Moreover, steps 3 and 4 show that the samples of $\boldsymbol{x}_{-i}$ are not used in calculating the index S_i for x_i; therefore, the calculation of S_i purely depends on the samples of x_i and y, and can be achieved even if the samples of $\boldsymbol{x}_{-i}$ are missing. In addition, the derivation of Eq. (11) does not assume uncorrelated inputs; thus, this algorithm can handle both correlated and uncorrelated inputs. These three advantages make this algorithm suitable for the GSA of a BN using the samples generated from the joint prior distribution.

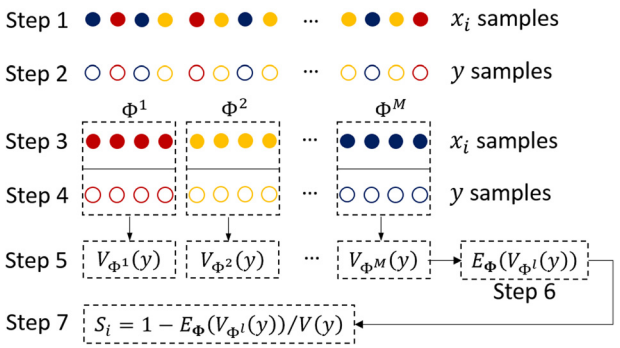


Fig. 7 Steps to realize the proposed algorithm

One question is how many samples are needed to reach desirable accuracy of the computed index. This question has been addressed in an earlier paper [28] by the authors. A heuristic rule is: (1) at least 50 intervals and (2) each interval Φ^l contains at least 50 samples. To guarantee this condition, we simply need to generate 2500 samples.

3.2 Variance Reduction Assessment by the Proposed Algorithm. As explained earlier, for a deterministic function $y = f(\boldsymbol{x})$ where $\boldsymbol{x} = \{x_1, \ldots, x_k\}$, the proposed algorithm only requires the samples of input x_i and output y to calculate the first-order Sobol' index S_i. And this calculation only needs to assign the samples to equally probable intervals of x_i and requires no more functional evaluation. For two nodes X_1 and X_N in the BN, since the existence of the deterministic function mapping X_1 to X_N has been proved in Sec. 2, the calculation of the first-order Sobol' index of node X_1 quantifying its contribution toward the uncertainty in node X_N only needs the samples of X_1 and X_N. Based on Eq. (11), the equation to compute the desired index S_{X_1} is

$$S_{X_1} = 1 - \frac{E_{\boldsymbol{\Phi}}\left(V_{\Phi^l}(X_N)\right)}{V(X_N)} \tag{12}$$

Specifically, the desired first-order index S_{X_1} can be calculated by considering the X_1 as "x_i" and X_N as "y" in Fig. 7.

The resultant index S_{X_1} is the average ratio of the variance reduction of X_N by fixing X_1 at its observation; and this is an average over all possible values of X_1. This is an informative estimate before the forward propagation or backward inference if the value of the observation is NOT known.

If the value of the observation of X_1 is known, this VRR estimate can be further improved by identifying the equally probably interval where the observation is located and computing the local variance. Denote $\hat{\Phi}$ as the equally probable interval that contains the observation $\hat{X}_1$, i.e., $\hat{X}_1 \in \hat{\Phi}$. Section 3.1 has proved that $V_{\hat{\Phi}}(X_N) = V\left(X_N|X_1 = X_1^{\#}\right)$ where $X_1^{\#} \in \hat{\Phi}$. Since $X_1^{\#}$ and $\hat{X}_1$ are in the same interval so that $\hat{X}_1 \approx X_1^{\#}$, then we have $V_{\hat{\Phi}}(X_N) \approx V(X_N|X_1 = \hat{X}_1)$. As the desired variance reduction ratio is $1 - V(X_N|X_1 = \hat{X}_1)/V(X_N)$, the improved estimate is

$$\text{VRR} \approx 1 - \frac{V_{\hat{\Phi}}(y)}{V(y)} \tag{13}$$

Compared to Eq. (12) which computes the average VRR of X_N over all possible values of X_1, Eq. (13) estimates the VRR of X_N at a specific value of X_1. The accuracy of Eq. (13) will be higher if: (1) $\hat{\Phi}$ is narrower so that $\hat{X}_1$ is closer to $X_1^{\#}$ and (2) more samples of X_N are assigned to $\hat{\Phi}$ so that $V\ \hat{\Phi}\ (X_N)$ is a better estimate of $V(X_N|X_1 = X_1^{\#})$.

3.3 Summary. Section 3.1 introduces a new algorithm to efficiently estimate the first-order Sobol' index from Monte Carlo samples. In the proposed algorithm, the calculation of the first-order Sobol' index of an input does not need the samples of other inputs. This advantage makes it an ideal algorithm for the GSA of a BN, where generating prior samples are much easier than building the underlying deterministic function in discussed in Sec. 2. The proposed algorithm can also handle correlated inputs, so that it is suitable for a BN where the nodes are generally correlated.

Section 3.2 described the implementation of the proposed algorithm to the BN. Equation (12) can be used to calculate the average ratio of the variance reduction of a node of interest X_N over all possible observations of another node X_1 ; Eq. (13) provides a better estimate of variance reduction ratio if the value of the observation is known.

In Sec. 4, two numerical examples are provided to illustrate the proposed GSA method for the BN. The first example is a time-independent static BN, and the second example is a time-dependent dynamic BN.

4 Numerical Examples

4.1 Structural Dynamics Problem. A structural dynamics problem provided by Sandia National Laboratories is used to illustrate the proposed method, and more details on this problem can be found in Refs. [29–32]. As shown in Fig. 8, the system of interest contains three mass–spring–damper components in series; and these components are mounted on a beam supported by a hinge at one end and a spring at the other end; and a sinusoidal force input $P = 3000\sin(350t)$ is applied on the beam.

This system has three model parameters of spring stiffnesses $\boldsymbol{k} = (k_1, k_2, k_3)$ and they are assumed to have unknown true values to be calibrated. The prior distribution of k_i is assumed to be Gaussian with a coefficient of variation of 10% and mean values of $\mu_{k_1} = 5000$, $\mu_{k_2} = 10,000$, and $\mu_{k_3} = 9000$.

The quantity to be measured for model calibration is the maximum acceleration A_3 in the third mass m_3. A computational model $A_3 = F(\boldsymbol{k})$ based on finite element analysis has been provided by Sandia National Laboratories [29].

To improve the computational efficiency, a Gaussian process (GP) [33,34] surrogate model $A_3 = \mathrm{GP}(\boldsymbol{k})$ is constructed to replace the expensive dynamics computational model. The prediction of the GP model is a Gaussian distribution $N(\mu(\boldsymbol{k}), \sigma^2(\boldsymbol{k}))$; thus, a CPD is given by the GP model.

The observation variable is denoted as D and we have $D = A_3 + \epsilon_m$ where ϵ_m is the measurement error with a zero-mean Gaussian distribution $\epsilon_m \sim N(0, \sigma_m^2)$. Thus, another CPD is given by the measurement error. In this example, σ_m is another parameter to be calibrated and we assign a noninformative uniform prior distribution $U(150, 250)$ to it.

A BN is established for this model calibration problem, as shown in Fig. 9. In this example, we are interested in: (1) calculating the first-order Sobol' index of the observation variable D quantifying its contribution toward the uncertainty in each calibration parameters of interest $\{k_1, k_2, k_3, \sigma_m\}$ and (2) predicting the VRR of the calibration parameters at a given observation.

As samples are generated from the joint prior distribution of this network, the first-order Sobol' indices of $\{k_1, k_2, k_3, \sigma_m\}$ are obtained by considering the calibration parameter as X_N and the observation variable D as X_1 in Eq. (12). The results are listed in Table 1. From this table, we conclude that the variance of k_1 will reduce by 50% on average due to calibration; the variance reduction of k_3 is 11% on average; while the variance of k_2 and σ_m will not be reduced significantly by calibration. This is a very valuable insight, which is also supported by Fig. 10. Thus, if we want to reduce the uncertainty in k_2, we need to observe another quantity. In the latter computation of VRR at specific observations of A_3, we focus on k_1 and k_3.

Table 1 shows the average VRR. Now assume that the specific observed value of A_3 is known (a synthetic data point). Based on Eq. (13), we predict the VRR of k_1 and k_3 for this specific observation, as shown in Table 2, where the "by inference" method mean we finish the Bayesian inference and compute the VRR based on the posterior distributions. In this example, the Bayesian inference is finished by the rejection sampling algorithm [35] which results in 2×10^4 samples from the posterior distributions of the calibration parameters. Figure 10 shows the probability density functions of these posterior distributions at data point of $D = 4500$. We recalculate the VRR by comparing the variances of the posterior samples and the prior samples. As shown in

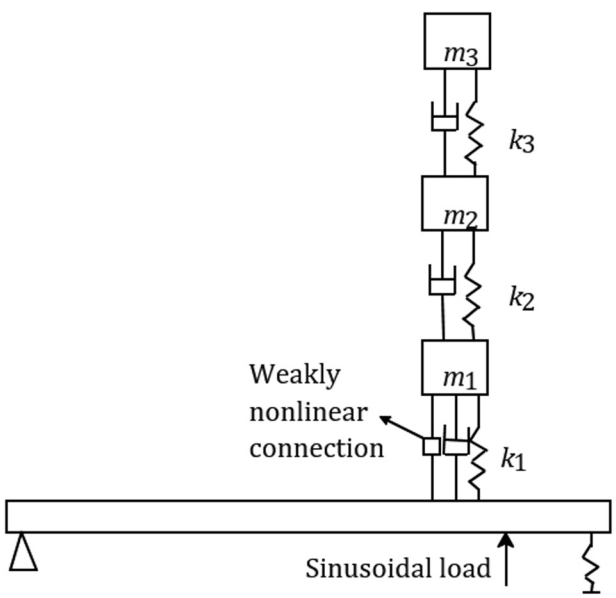


Fig. 8 Beam with mass–spring–damper

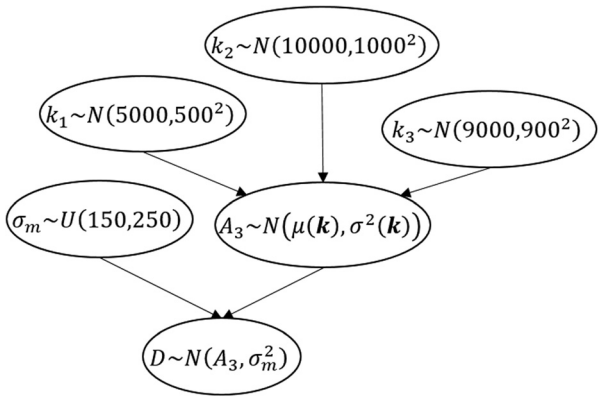


Fig. 9 BN of example 1

Table 1 First-order Sobol' index, example 1

Parameter of interest	k_1	k_2	k_3	σ_m
First-order Sobol' index of observation node D	0.50	0.02	0.11	0.00

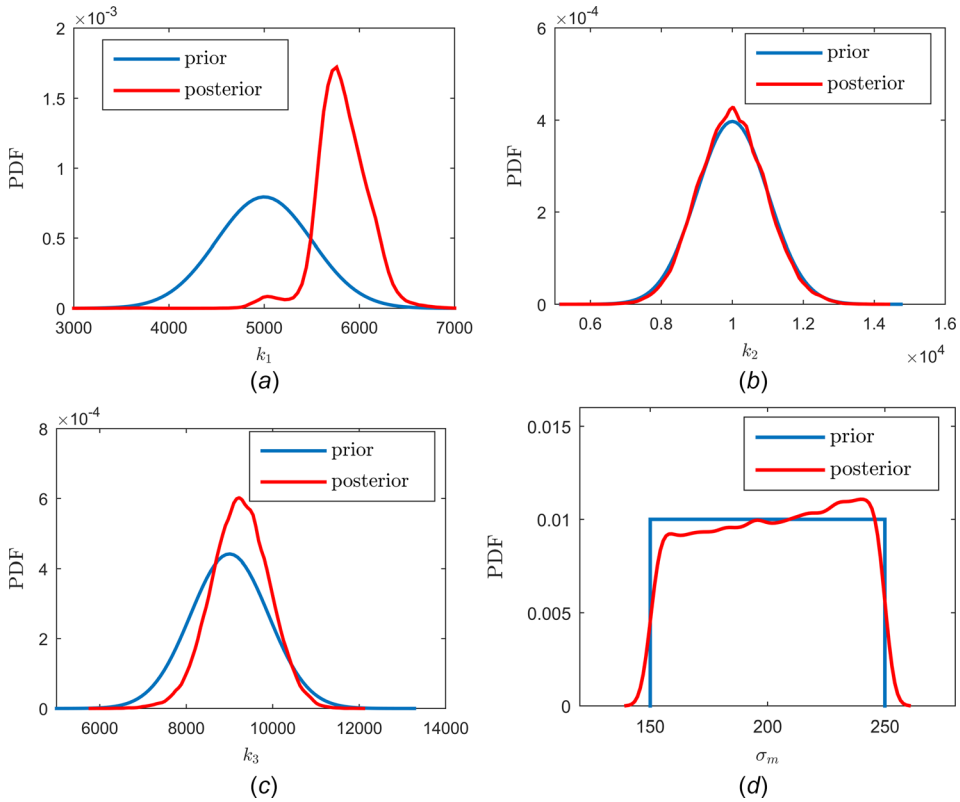


Fig. 10 Posterior distributions at observation value of $A_3 = 4500$

Table 2 Variance reduction ratio at specific observations of A_3

	k_1		k_3	
Data point	Proposed method (Bayesian inference NOT needed) (%)	By inference (Bayesian inference needed) (%)	Proposed method (%)	By inference (%)
3900	49.9	47.0	3.2	4.0
4000	45.2	44.2	13.6	15.4
4100	38.3	42.4	23.6	19.7
4200	43.5	44.8	20.7	25.5
4300	48.2	54.2	35.2	31.9
4400	60.5	63.2	41.7	38.9
4500	69.6	69.1	43.3	44.7

Table 2, our earlier predictions are close to the sample-based results. However, the proposed method only uses the samples from the prior distribution, and no actual Bayesian inference effort is required. In a personal computer of Intel Core i7-4710HQ CPU, the proposed method spends less than 1 s to calculate the sensitivity index at a data point, while the other method by inference spends about 20 s.

In summary, this example verified the effectiveness of the proposed method to predict the variance reduction ratio before conducting the Bayesian inference. Thus, the proposed method provides valuable guidance for selecting observation nodes; for example, in the subsequent updating, nodes such as k_2 and σ_m cannot be updated by observing A_3 data.

4.2 Example of a Dynamic Bayesian Network. This example applies the proposed method to a mathematical example of a dynamic BN, as shown in Fig. 11. The CPDs of this dynamic BN are as follows.

The root node C_0 has an unknown true value to be calibrated, so that C_0 is a static node and $C_0^t = C_0^{t+1}$. The prior distribution of C_0 is $N(2, 0.5^2)$. C_1 and C_2 are two dynamic state variables, and their states are to be tracked. At $t = 1$ the CPD of the child node C_1 is $C_1^1 \sim N(C_0^{12} + 10,\ 1^2)$; at $t > 1$ the CPD of C_1 is $C_1^t \sim N(C_0^{t\,2} + 0.9C_1^{t-1} + 1,\ 1^2)$, thus the distribution of C_1 depends on its previous value and the value of C_0. C_2 is the child

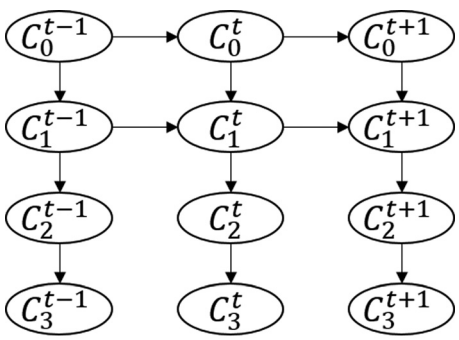


Fig. 11 Dynamic BN of example 2

node of C_1 and its CPD is $C_2^t \sim N(C_1^{t\,2}, 5^2)$. In this problem, the observation node is C_3 and its CPD is $C_3^t \sim N(C_2^t, (C_2^t/20)^2)$, i.e., the value of C_2^t plus a measurement error of zero mean Gaussian distribution. This example considers the first 30 steps of this dynamic BN. Assuming the true value of C_0 is 2.5, the synthetic data of the observation node C_3 is generated at each step, as shown in Fig. 12.

A widely used particle filter method named sequential importance resampling (SIR) algorithm [36] is applied in this example to track the state variables. Here a particle is a sample from the joint distribution of the state variables. This SIR algorithm propagates the particles of the posterior distribution at time step $t-1$ to time step t to obtain the particles of the prior distribution of time step t. The likelihoods of these particles are calculated and normalized as the weights for them. Then the particles are resampled based on the weight terms and the resultant new particles represent the posterior joint distribution of the state variables in time step t. Details of this SIR algorithm can be found in Ref. [36] and the Appendix.

The number of particles in this example is 50,000. The mean value and 95% bounds of the posterior distributions of the state variables are shown in Fig. 13. The uncertainty of C_0 reduces and its posterior distribution approximates to its true value 2.5, but this uncertainty reduction is not significant after step 20.

At each step, before the calculation of the likelihoods and the particle resampling, we apply the proposed method using the particles of the prior joint distribution of the state variables. The VRR of each state variable is predicted by the proposed method of Eq. (13) using the prior samples of the state variables. This VRR is also calculated by the prior/posterior samples at each step. Figure 14 shows that the results from these two methods are consistent so that the proposed method is verified. Note that the

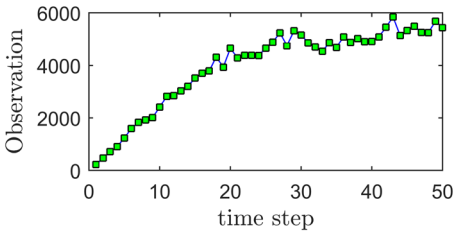
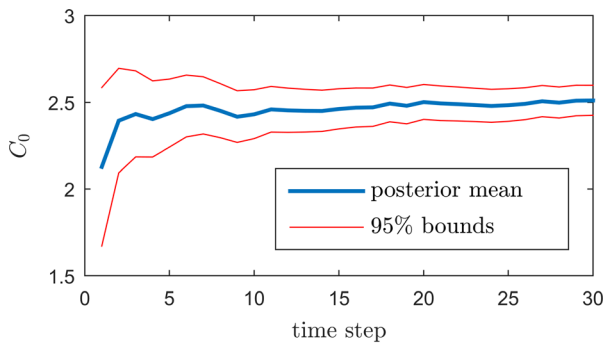
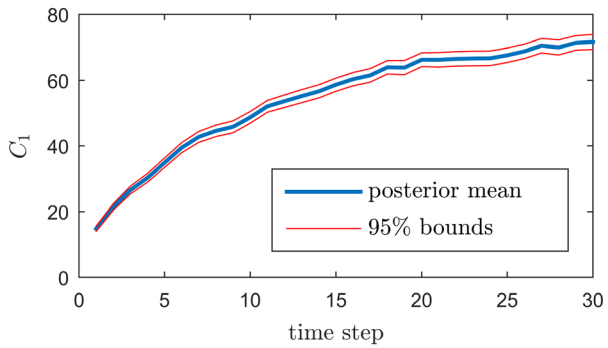


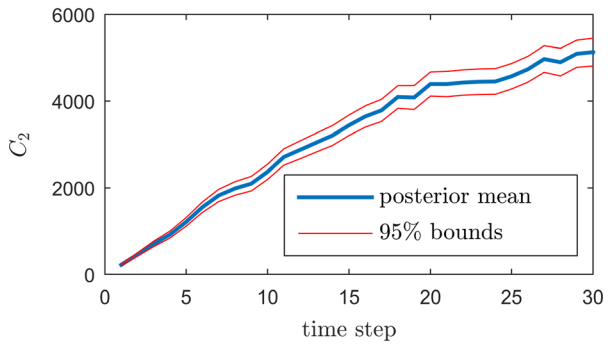
Fig. 12 Observations, example 2



(*a*)

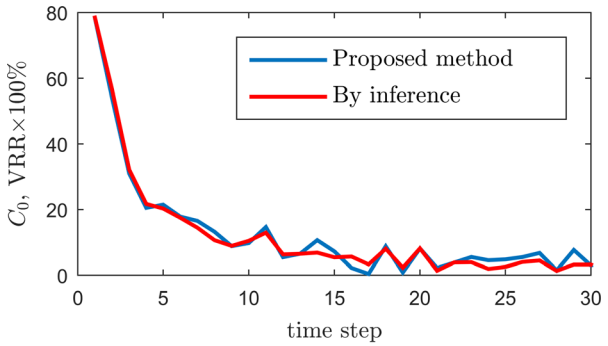


(*b*)

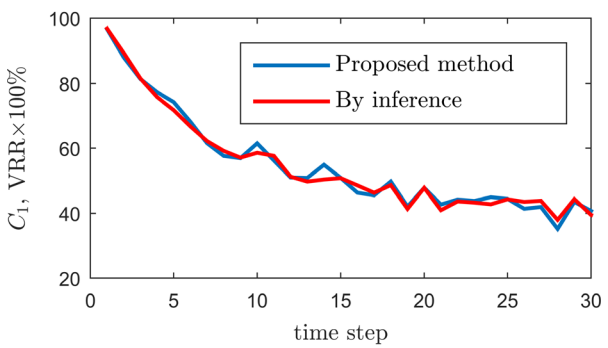


(*c*)

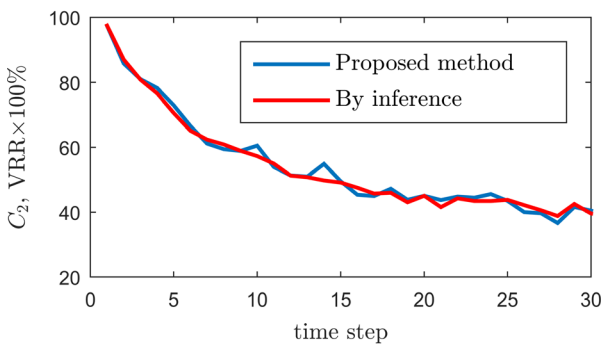
Fig. 13 Posterior distribution of state variables



(*a*)



(*b*)



(*c*)

Fig. 14 VRR of the state variables

proposed method uses the prior samples and the observation data, while the sample-based method needs both the prior and posterior sample. In other words, the proposed method can be applied before Bayesian inference, but the sample-based method happens after the Bayesian inference has been done.

In this example, the CPD of C_0 is $C_0^t = C_0^{t+1}$ so the uncertainty of C_0 will not be enlarged in the propagation from time step $t-1$ to time step t. However, its uncertainty is reduced by the updating in each time step. Figure 13 shows that this uncertainty reduction is significant for the first five times steps so that the VRR in Fig. 14 has a large value before time step 5; this uncertainty reduction is negligible after time step 20 so that the value of VRR is closer to zero after time step 20.

In comparison, Fig. 13 shows that the uncertainty in the posterior distributions of C_1 and C_2 are increasing, even if their VRR values in Fig. 14 are always significant. The reason is that the uncertainties of C_1 and C_2 are enlarged in the propagation from time step $t-1$ to time step t, so their prior distributions at t have more uncertainty than the posterior distribution at $t-1$. The uncertainty in the prior at t is reduced by the updating, but the posterior uncertainty at t may not be smaller than the posterior uncertainty at $t-1$ if the uncertainty reduction by the updating cannot outperform the uncertainty enlargement by the propagation. Note that the VRR in Fig. 14 is the variance reduction with respect to the prior/posterior distribution at the same time step, not the variance reduction for adjacent posterior distributions.

In summary, this example extended the proposed sensitivity analysis to a dynamic BN and verified its validity. The proposed method is capable of predicting the variance reduction of each state variable before updating.

5 Conclusion

In a BN, how a node of interest is affected by fixing another node at some value is of prominent interest. The proposed GSA for BN calculates the first-order Sobol' index to quantity the sensitivity of the node of interest X_N with respect to an input/observation node X_1. In forward propagation where X_1 is an upstream node, a low index indicates that X_N is not sensitive to the value of X_1 so that we can simply fix X_1 at a deterministic value. In backward inference where X_1 is a downstream node, a low sensitivity index indicates that the uncertainty of X_N cannot be reduced by observing X_1; thus, we should observe another node of higher Sobol' index in order to reduce its uncertainty.

The proposed GSA for BN is developed in two steps. First, an auxiliary variable method is used to convert the path between node X_1 and node X_N to a deterministic function thus making the Sobol' index computation feasible for a BN. If the path from X_1 to X_N is not a directed path form, the theorem of arc reversal is used to transform it to the desired directed path so that the auxiliary variable method can still be used to build the deterministic function. Second, this paper uses an efficient algorithm to directly estimate the Sobol' index from the samples of the prior distribution of the BN, so that the proposed GSA for the BN is computationally affordable. The resultant Sobol' index is the average variance reduction ratio across all possible observations of X_1. This paper also extend the algorithm to give a better estimate of the uncertainty reduction of the node of interest when the value of the observation is known, thus providing an informative guidance before the updating.

The limitation of the proposed method at present is that currently it only considers a single observation; thus, an extension to the case of multiple observations needs to be addressed in future work.

Acknowledgment

The authors thank Dr. Zhen Hu for valuable discussions.

Nomenclature

$\mathcal{F}^{-1}(\cdot)$ = inverse CDF
$\boldsymbol{k}$ = stiffnesses
P_{V_1}, P_{V_2} = parent nodes of V_1, V_2
S_i = first-order Sobol' index of x_i
U_x = auxiliary variable of x
V_1, V_2 = nodes in BN
x_i = a random variable
$\boldsymbol{x}_{-i}$ = all random variables other than x_i
X_j = a node in the BN
$\boldsymbol{\theta}_x$ = distribution parameter of random variable x
$\pi(\cdot)$ = probability density function
Φ^I = a equally probably interval

Funding Data

- The Air Force Office of Scientific Research (Grant No. FA9550-15-1-0018).

Appendix: Introduction to Particle Filter

Particle filter is a general algorithm to track the evolution of the state variables in a dynamic Bayesian network (DBN). In this section, capital letters denote random variables; lower-case letters denote particles, where the superscript i indicates that it is the ith particle. The subscripts of letters indicate the time step. Thus, the state variables at time step t are denoted as $\boldsymbol{X}_t$, as shown in the simple DBN of Fig. 15.

Assume that the state variables $\boldsymbol{X}_t \in \Re^m$ at time t evolves from the state variables $\boldsymbol{X}_{t-1} \in \Re^m$ according to the state function

$$\boldsymbol{X}_t = f(\boldsymbol{X}_{t-1}, \boldsymbol{v}_{t-1}) \tag{A1}$$

where $\boldsymbol{v}_{t-1} \in \Re^m$ is the vector of noise terms in the state function. The measurement $\boldsymbol{Z}_t \in \Re^n$ is obtained according to the measurement function

$$\boldsymbol{Z}_t = h(\boldsymbol{X}_t, \boldsymbol{n}_t) \tag{A2}$$

where $\boldsymbol{n}_t \in \Re^n$ is the vector of noise terms in the measurement function.

In case that the DBN represented by Eqs. (A1) and (A2) is not a linear Gaussian DBN, several particle filter algorithms have been developed to track the evolution of $\boldsymbol{X}_t$ and $\boldsymbol{Z}_t$. The most basic particle filter algorithm is sequential importance sampling [36]. The sequential importance sampling considers the full joint posterior distribution at time step t, $p(\boldsymbol{X}_{0:t}|\boldsymbol{Z}_{1:t})$. This distribution is approximated with a weighted set of particles $\{\boldsymbol{x}^i_{0:t}, \omega^i_t\}_{t=1}^N$. These particles approximate the joint posterior distribution $p(\boldsymbol{X}_{0:t}|\boldsymbol{Z}_{1:t})$ by

$$p(\boldsymbol{X}_{0:t}|\boldsymbol{Z}_{1:t}) \approx \sum_{i=1}^{N} \omega^i_t \delta_{\boldsymbol{x}^i_{0:t}} \tag{A3}$$

where $\delta_{\boldsymbol{x}^i_{0:t}}$ is a delta function at $\boldsymbol{x}^i_{0:t}$.

At time step t, the ith particle of $\boldsymbol{X}_t$ is denoted as $\boldsymbol{x}^i_t$, and it is sampled based on the current state $\boldsymbol{X}^i_{0:t-1}$ and the observation $\boldsymbol{Z}_{1:t}$ according to a proposal density

$$\boldsymbol{X}^i_t \sim q(\boldsymbol{X}_t|\boldsymbol{X}^i_{0:t-1}, \boldsymbol{Z}_{1:t}) \tag{A4}$$

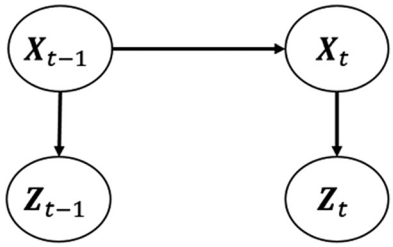


Fig. 15 A simple DBN

In other words, the new state $\boldsymbol{X}_t^i$ of the ith particle at time step t is sampled from a distribution which takes the current state $\boldsymbol{X}_{0:t-1}^i$ and the observation $\boldsymbol{Z}_{1:t}$ as parameters.

At time step t, the weight ω_t^i is updated from ω_{t-1}^i by

$$\omega_t^i \propto \omega_{t-1}^i \frac{p(\boldsymbol{Z}_t|\boldsymbol{X}_t^i)p(\boldsymbol{X}_t^i|\boldsymbol{X}_{t-1}^i)}{q(\boldsymbol{X}_t^i|\boldsymbol{X}_{t-1}^i, \boldsymbol{Z}_t)} \tag{A5}$$

In addition, the initial state $\boldsymbol{X}_0^i$ are sampled from the joint prior distribution of the state variables, and the initial weight ω_0^i for each particle is $1/N$.

In practice, iterations of Eqs. (A4) and (A5) over time step t may lead to particle degeneracy problem, i.e., only a few particles have significant weights but most particles have negligible weights. In that case, we are assigning most computational efforts to the particles of nonsignificant contribution to the posterior distribution. This degeneracy problem can be solved by resampling, i.e., generating a new set of N particles based on the particles of $\boldsymbol{X}_t$. The new particles represent the same posterior distribution as the former particles.

The simplest strategy of resampling is generating new resampled particles based on the discrete approximation shown in Eq. (A3), and the weight of each new particle is set as $1/N$ again. This resampling is bootstrapping process of N iterations, and each iteration selects one particle from current particles with replacement. In an iteration, the probability that a particle is selected is proportional to its weight.

The resampling strategy mentioned previously based on Eq. (A3) is adopted in a widely used algorithm, the SIR algorithm [36]. The SIR algorithm: (1) takes the state transition distribution $p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}^i)$ as the proposal density distribution $q(\boldsymbol{X}_t|\boldsymbol{X}_{0:t-1}^i, \boldsymbol{Z}_{1:t})$, and (2) conducts resampling at each iteration. Thus, Eqs. (A4) and (A5) reduce to

$$\boldsymbol{X}_t^i \sim p(\boldsymbol{X}_t|\boldsymbol{X}_{t-1}^i) \tag{A6}$$

$$\omega_t^i \propto p(\boldsymbol{Z}_t|\boldsymbol{X}_t^i) \tag{A7}$$

Note that resampling based on Eq. (A3) is after the calculation of Eqs. (A6) and (A7) at each time step, where new particles of $\boldsymbol{X}_t^i$ are generated and the weight of each new particle is set as $1/N$.

References

[1] Ling, Y., and Mahadevan, S., 2012, "Integration of Structural Health Monitoring and Fatigue Damage Prognosis," Mech. Syst. Signal Process., **28**, pp. 89–104.

[2] Sankararaman, S., Ling, Y., and Mahadevan, S., 2011, "Uncertainty Quantification and Model Validation of Fatigue Crack Growth Prediction," Eng. Fract. Mech., **78**(7), pp. 1487–1504.

[3] Helman, P., Veroff, R., Atlas, S. R., and Willman, C., 2004, "A Bayesian Network Classification Methodology for Gene Expression Data," J. Comput. Biol., **11**(4), pp. 581–615.

[4] Friedman, N., Geiger, D., and Goldszmidt, M., 1997, "Bayesian Network Classifiers," Mach. Learn., **29**(2–3), pp. 131–163.

[5] Korb, K. B., and Nicholson, A. E., 2010, *Bayesian Artificial Intelligence*, CRC Press, Boca Raton, FL.

[6] Poole, D., 1993, "Probabilistic Horn Abduction and Bayesian Networks," Artif. Intell., **64**(1), pp. 81–129.

[7] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola, S., 2008, *Global Sensitivity Analysis: The Primer*, Wiley, Chichester, UK.

[8] Hu, Z., and Du, X., 2015, "Mixed Efficient Global Optimization for Time-Dependent Reliability Analysis," ASME J. Mech. Des., **137**(5), p. 051401.

[9] Li, C., and Mahadevan, S., 2016, "Robust Test Resource Allocation Using Global Sensitivity Analysis," AIAA Paper No. 2016-0952.

[10] Li, C., and Mahadevan, S., 2016, "Relative Contributions of Aleatory and Epistemic Uncertainty Sources in Time Series Prediction," Int. J. Fatigue, **82**(Part 3), pp. 474–486.

[11] Nannapaneni, S., and Mahadevan, S., 2014, "Uncertainty Quantification in Performance Evaluation of Manufacturing Processes," IEEE International Conference on Big Data (Big Data), Washington, DC, Oct. 27–30, pp. 996–1005.

[12] Mullins, J., Li, C., Mahadevan, S., and Urbina, A., 2014, "Optimal Selection of Calibration and Validation Test Samples Under Uncertainty," *Model Validation and Uncertainty Quantification*, Vol. 3, H. S. Atamturktur, B. Moaveni, C. Papadimitriou, and T. Schoenherr, eds., Springer International Publishing, Berlin, pp. 391–401.

[13] Li, C., and Mahadevan, S., 2015, "Sensitivity Analysis for Test Resource Allocation," *Model Validation and Uncertainty Quantification*, Vol. 3, SE—14, H. S. Atamturktur, B. Moaveni, C. Papadimitriou, and T. Schoenherr, eds., Springer International Publishing, Berlin, pp. 143–150.

[14] Saltelli, A., and Tarantola, S., 2002, "On the Relative Importance of Input Factors in Mathematical Models," J. Am. Stat. Assoc., **97**(459), pp. 702–709.

[15] Marrel, A., Iooss, B., Laurent, B., and Roustant, O., 2009, "Calculations of Sobol Indices for the Gaussian Process Metamodel," Reliab. Eng. Syst. Saf., **94**(3), pp. 742–751.

[16] Sobol', I. M., 2001, "Global Sensitivity Indices for Nonlinear Mathematical Models and Their Monte Carlo Estimates," Math. Comput. Simul., **55**(1–3), pp. 271–280.

[17] Zhang, X., and Pandey, M. D., 2014, "An Effective Approximation for Variance-Based Global Sensitivity Analysis," Reliab. Eng. Syst. Saf., **121**, pp. 164–174.

[18] Sudret, B., 2008, "Global Sensitivity Analysis Using Polynomial Chaos Expansions," Reliab. Eng. Syst. Saf., **93**(7), pp. 964–979.

[19] Chen, W., Jin, R., and Sudjianto, A., 2005, "Analytical Variance-Based Global Sensitivity Analysis in Simulation-Based Design Under Uncertainty," ASME J. Mech. Des., **127**(5), pp. 875–884.

[20] Homma, T., and Saltelli, A., 1996, "Importance Measures in Global Sensitivity Analysis of Nonlinear Models," Reliab. Eng. Syst. Saf., **52**(1), pp. 1–17.

[21] Sobol', I. M., and Myshetskaya, E. E., 2008, "Monte Carlo Estimators for Small Sensitivity Indices," Monte Carlo Methods Appl., **13**(5–6), pp. 455–465.

[22] Owen, A., 2013, "Better Estimation of Small Sobol Sensitivity Indices," ACM Trans. Model. Comput. Simul., **23**(2), pp. 11–17.

[23] Saltelli, A., 2002, "Making Best Use of Model Evaluations to Compute Sensitivity Indices," Comput. Phys. Commun., **145**(2), pp. 280–297.

[24] Sankararaman, S., and Mahadevan, S., 2013, "Separating the Contributions of Variability and Parameter Uncertainty in Probability Distributions," Reliab. Eng. Syst. Saf., **112**, pp. 187–199.

[25] Angus, J. E., 1994, "The Probability Integral Transform and Related Results," SIAM Rev., **36**(4), pp. 652–654.

[26] Li, C., and Mahadevan, S., 2015, "Global Sensitivity Analysis for System Response Prediction Using Auxiliary Variable Method," AIAA Paper No. 2015-0661.

[27] Shachter, R. D., 1986, "Evaluating Influence Diagrams," Oper. Res., **34**(6), pp. 871–882.

[28] Li, C., and Mahadevan, S., 2016, "An Efficient Modularized Sample-Based Method to Estimate the First-Order Sobol' Index," Reliab. Eng. Syst. Saf., **153**, pp. 110–121.

[29] Red-Horse, J. R., and Paez, T. L., 2008, "Sandia National Laboratories Validation Workshop: Structural Dynamics Application," Comput. Methods Appl. Mech. Eng., **197**(29–32), pp. 2578–2584.

[30] Li, C., and Mahadevan, S., 2014, "Uncertainty Quantification and Output Prediction in Multi-Level Problems," AIAA Paper No. 2014-0124.

[31] Li, C., and Mahadevan, S., 2016, "Role of Calibration, Validation, and Relevance in Multi-Level Uncertainty Integration," Reliab. Eng. Syst. Saf., **148**, pp. 32–43.

[32] Mullins, J., Li, C., Sankararaman, S., Mahadevan, S., and Urbina, A., 2013, "Probabilistic Integration of Validation and Calibration Results for Prediction Level Uncertainty Quantification: Application to Structural Dynamics," AIAA Paper No. 2013-1872.

[33] Xu, P., Su, X., Mahadevan, S., Li, C., and Deng, Y., 2014, "A Non-Parametric Method to Determine Basic Probability Assignment for Classification Problems," Appl. Intell., **41**(3), pp. 681–693.

[34] Pan, F., Zhu, P., Chen, W., and Li, C., 2013, "Application of Conservative Surrogate to Reliability Based Vehicle Design for Crashworthiness," J. Shanghai Jiaotong Univ., **18**(2), pp. 159–165.

[35] Henrion, M., 1988, "Propagation of Uncertainty by Probabilistic Logic Sampling in Bayes' Networks," Uncertainty Artif. Intell., **5**, pp. 149–164.

[36] Arulampalam, M. S., Maskell, S., Gordon, N., and Clapp, T., 2002, "A Tutorial on Particle Filters for Online Nonlinear/Non-Gaussian Bayesian Tracking," IEEE Trans. Signal Process., **50**(2), pp. 174–188.