Bayesian network modeling of system performance

M.T. Bensi & A. Der Kiureghian
University of California, Berkeley, USA

D. Straub
Technical University of Munich, Germany

ABSTRACT: Bayesian networks (BNs) provide an excellent framework for modeling system performance, particularly in near-real time applications when it is necessary to update models in light of observations. However, BNs can be very demanding of computer memory and inference can become intractable if care is not taken to optimize their topology. In this paper, efficient BN formulations for modeling system performance are presented. First, formulations are developed for series and parallel systems. Then, results are extended to general systems for which the minimal link and/or cut sets are known. Finally, an optimization algorithm is developed to automate the generation of efficient BN formulations for modeling system performance.

1 INTRODUCTION

Civil infrastructure systems are backbones of modern societies, yet in many regions of the world they are vulnerable to seismic hazards. The authors are currently working toward the development of a probabilistic decision-support system for seismic infrastructure risk assessment and management using a Bayesian network (BN) formulation. Components of the BN framework for seismic risk assessment and management include: (1) a seismic demand model of ground motion intensity as a spatially distributed Gaussian random field accounting for finite fault rupture and directivity effects; (2) models of component performance; (3) models of system performance; and (4) the extension of the BN to include decision and utility nodes to aid pre- and post-earthquake decision-making. This paper focuses on the third component of the proposed framework, modeling of system performance, with particular emphasis on issues related to computational efficiency.

A BN is a probabilistic graphical model that represents a set of random variables and their probabilistic dependencies. The ease with which BNs facilitate information updating makes them ideally suited for the proposed application. However, like all computational methods, BNs have limitations. In particular, calculations in BNs can be highly demanding of computer memory. Space constraints do not permit the inclusion of an introduction on BNs here. Comprehensive coverage is available in textbooks (Jensen & Nielson 2007; Kjaerulff & Madsen 2008) and overviews are presented in many papers (e.g. Friis-Hansen 2004; Bayraktarli et al. 2006; D. Straub 2005; Bensi et al. 2010).

2 MODELING SYSTEM PERFORMANCE VIA BAYESIAN NETWORK

Consider a system of n components. Each component $i \in [1, n]$ has d_i discrete component states. Therefore, the number of distinct configurations of the system is $\prod_{i=1}^{n} d_i$. Bensi et al.

(2009) refer to the BN formulation that corresponds to the joint consideration of all combinations of component and system states as the *naïve BN formulation*. In this formulation, a single node S_{sys}, representing the state of the system, is defined as a child of all nodes representing the states of the components (a converging structure) as shown in Figure 1a. For a system with n binary components and binary system state ($\Pr(\text{survive}) = 1 - \Pr(\text{failure})$), node S_{sys} has a conditional probability table (CPT) of size 2^n. (The CPT provides the conditional probabilities of the states of a node for given states of its parent nodes.) For systems with a large number of components, the size of the CPT in the naïve formulation quickly causes the BN to become computationally intractable. However, the naïve formulation is useful in applications where the number of components is small, e.g. in the reliability assessment of simple structural systems. Mahadevan et al. (2001) utilize a formulation as in Figure 1a with components representing limit-state functions corresponding to failure modes of structural systems. The number of limit-state functions in their example is sufficiently small so that the size of the CPT associated with the system node is not prohibitively large. However, for realistic infrastructural systems such an approach to modeling is impractical.

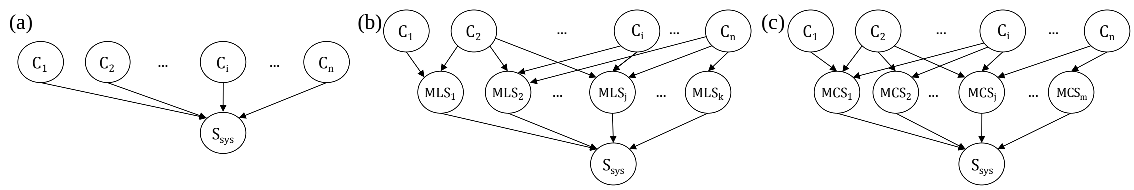

Figure 1: (a) Naïve formulation; (b) MLS formulation; and (c) MCS formulation

As an alternative to the naïve approach to modeling system performance, Bensi et al. (2009) offer four additional BN formulations. We first review two of the formulations presented in that paper, which are based on use of minimal link and cut sets. Then, we adapt these two formulations with the goal of minimizing computational demands. We focus first on series and parallel systems followed by consideration of general systems. Due to page limitations, we limit the scope of this paper to systems with binary states. However, it is noted that the formulations presented here can be extended to systems with multi-state components.

2.1 *Modeling performance of series and parallel systems with binary component states*

A common approach to making systems analysis more methodical is through the use of minimal link sets and minimal cut sets. A minimal link set (MLS) is a minimum set of components whose joint survival constitutes survival of the system. The *minimal link set BN formulation* introduces intermediate nodes between the component and system nodes which correspond to the MLSs. Torres-Toledano and Sucar (1998) use a MLS-based BN formulation for modeling system performance, though with less formality and generality than is described here. Figure 1b shows an example of the MLS BN formulation. The binary states of the MLS nodes are defined such that each MLS node is in the survival state only if all its constituent components have survived; otherwise it is in the failure state. The system node is in the survival state if any MLS node is in the survival state. With binary component and system states, the size of the CPT for each MLS is 2 to the power of the number of its constituent components, and the size of the system node CPT is 2 to the power of the number of MLSs. As a result, when the number of MLSs is large, the size of the CPT associated with the system node S_{sys} becomes large. A similar problem occurs for an MLS nodes when the number of its constituent components is large.

The dual of the MLS formulation is the *minimal cut set BN formulation*. A minimal cut set (MCS) is a minimum set of components whose joint failure constitutes failure of the system. In this formulation, the system node is a child of nodes representing MCSs, and each MCS node is a child of nodes representing the states of its constituent components. Figure 1c shows a conceptual BN employing the MCS formulation. The system node is a series system of all the MCS nodes (i.e., the system node is in the failure state if at least one MCS node is in the failure state), whereas each MCS is a parallel system of its parent nodes (i.e. all constituent components must be in the failure state for the MCS node to be in the failure state). As with the MLS formulation, the CPTs in this formulation become large as the number of MCSs increases and/or the number of components in an individual MCS becomes large.

In general, BN structures with nodes arranged in chains are significantly more efficient than the converging configurations characterizing the naïve and MLS/MCS formulations described above. Consider the two equivalent BN structures shown in Figure 2. The figure on the left shows a converging structure and the one on the right illustrates a chain structure. Both BNs model systems whose component states are dependent on a common demand D. A formal description of the construction of the BN in Figure 2b is presented later in this paper. Based on an investigation of growth in computational complexity for both structures, it is found that the converging structure is associated with exponentially increasing memory demand, while the complexity of the chain structure grows linearly. However, the converging structure is more efficient than the chain structure for systems with less than four components.

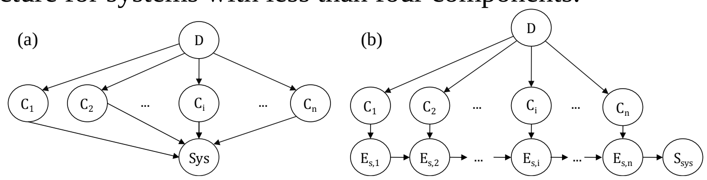

Figure 2: Illustration of BNs with (a) converging structure; and (b) chain structure

We next describe how system performance can be modeled with BNs having the chain topology. Define a *survival path sequence* (SPS) as a chain of events, corresponding to a MLS, in which the terminal event in the sequence indicates whether or not all the components in the MLS are in the survival state. Note that the term "sequence" does not have any time implications. A series system has one MLS and a parallel system has n MLSs. It follows that a series system has one SPS and a parallel system has n SPSs. A SPS is comprised of a chain of *survival path events* (SPEs), each of which describes the state of the sequence up to that event. SPEs are represented in the BN by nodes labeled $E_{s,i}$, the subscript i indicating that the particular SPE is associated with component i. The state of $E_{s,i}$ is defined as

$$\begin{aligned} E_{s,i} &= 1 \text{ if } E_{s,\mathrm{Pa}(i)} = 1 \ \cap \ C_i = 1 \\ &= 0 \text{ otherwise} \end{aligned} \tag{1}$$

where $E_{s,\mathrm{Pa}(i)}$ defines the state of the SPE node that is parent to $E_{s,i}$; $E_{s,i} = 1$ indicates that the node (or any other node considered in this paper) is in the survival state and $E_{s,i} = 0$ indicates its failure. C_i denotes the state of component i with $C_i = 1$ ($C_i = 0$) indicating the survival (failure) state. Thus, for a series system, the BN formulation takes the form shown in Figure 3a. The state of node $E_{s,1}$ is equal to the state of node C_1. $E_{s,2}$ is in survival state only if $E_{s,1}$ is in survival state and C_2 is in survival state. This pattern continues such that $E_{s,n}$ is in the survival state only if both $E_{s,n-1}$ and C_n are in the survival state. Consequently, the state of $E_{s,n}$ describes the state of the entire SPS (i.e. it indicates whether all components in the MLS have survived) and, therefore, that of the system.

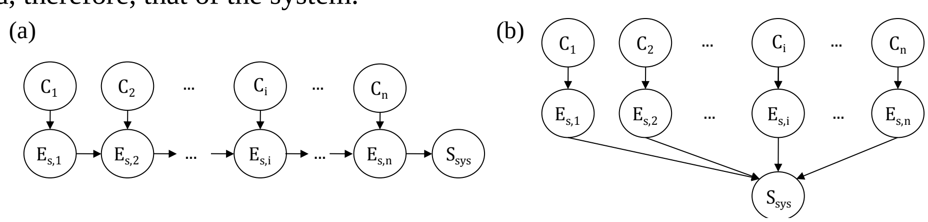

Figure 3: BN using SPEs to define performance of (a) a series system and (b) a parallel system

A parallel system has a SPS corresponding to each component. The resulting BN formulation is shown in Figure 3b. The system node indicates system survival if any node $E_{s,i}$ is in the survival state. Like the naïve formulation, the exponential growth in the size of the CPT associated with node S_{sys} renders the BN intractable when the number of components in the system is large.

Define a *failure path sequence* (FPS) as a chain of events, corresponding to a MCS, in which the terminal event in the sequence indicates whether or not all components in the MCS are in the failure state. For a parallel system, there is only one MCS and thus one FPS. For a series system with n components, there are n FPSs, one corresponding to each component. A FPS is comprised of a chain of *failure path events* (FPEs), each of which gives the state of the sequence up to that event. Let $E_{f,i}$ be the FPE associated with component i. The state of $E_{f,i}$ is expressed as

$$\begin{aligned} E_{f,i} &= 0 \text{ if } E_{f,\mathrm{Pa}(i)} = 0 \cap C_i = 0 \\ &= 1 \text{ otherwise} \end{aligned} \tag{2}$$

where $E_{f,\mathrm{Pa}(i)}$ defines the state of the FPE node that is parent to $E_{f,i}$. Thus, for a parallel system, the BN formulation takes the chain form shown in Figure 4a. The Boolean logic used to construct the CPTs in this BN is dual of that used for SPSs, i.e. $E_{f,i}$ is in the failure state only if the parent FPE is in the failure state and C_i is also in the failure state. For a series system, the BN formulation using FPSs is shown in Figure 4b. The size of the CPT associated with S_{sys} is 2^n and there is no computational advantage to this approach over the naïve formulation for series systems. These findings suggest that a combination of SPS and FPS formulations can be used to efficiently model general systems. This approach is described in the next section.

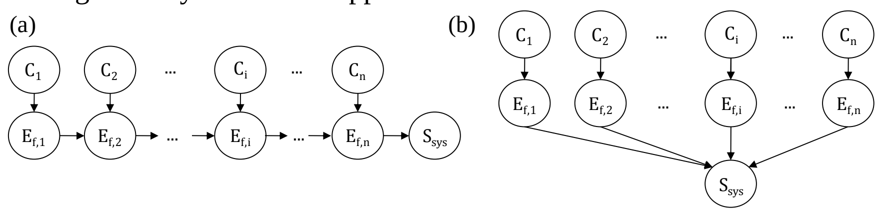

Figure 4: BN using FPEs to define (a) a parallel system and (b) a series system

2.2 *Modeling performance of general systems with binary component states*

A MLS is a series system of its constituent components. Therefore, based on the above discussion, one can construct a SPS to describe each MLS. Consider the example system in Figure 5a, which has four MLSs: MLS_1={1,7,8}, MLS_2 ={2,7,8}, MLS_3 ={3,7,8} and MLS_4 ={4,5,6,7,8}. In Figure 5b each MLS is modeled as an individual SPS. The SPEs, $E_{s,i}^j$, in each SPS are indexed by a subscript corresponding to the associated component i and a superscript corresponding to SPS/MLS number j. The dependence between the SPEs corresponding to the same component is modeled through a common parent node. The system node is in the survival state if the terminal node of any SPS is in the survival state. For reference, the BN formulation in which the MLSs are arranged in chain structures is named *efficient MLS BN formulation*. Similar logic leads to the creation of an *efficient MCS BN formulation*, whereby strings of FPSs are constructed corresponding to each MCS. The dependence between SPEs or FPEs sharing a component increases the computational demand when performing inference in the BN. By coalescing common SPEs/FPEs that appear in multiple SPSs/FPSs, the number of nodes and links in the BN, and hence the computational demand, are reduced. In the example system, components 7 and 8 appear in all SPSs. We take advantage of this observation and introduce only one "instance" of the SPEs associated with these components. The resulting BN is shown in Figure 6a. The states of SPE nodes having multiple SPEs as parents (e.g. node $E_{s,7}$ in Figure 6a) are specified using the Boolean relation

$$\begin{aligned} E_{s,i} &= 1 \text{ if } \left[\cup \{E_{s,\mathrm{Pa}(i)} = 1\}\right] \cap C_i = 1 \\ &= 0 \text{ otherwise} \end{aligned} \tag{3}$$

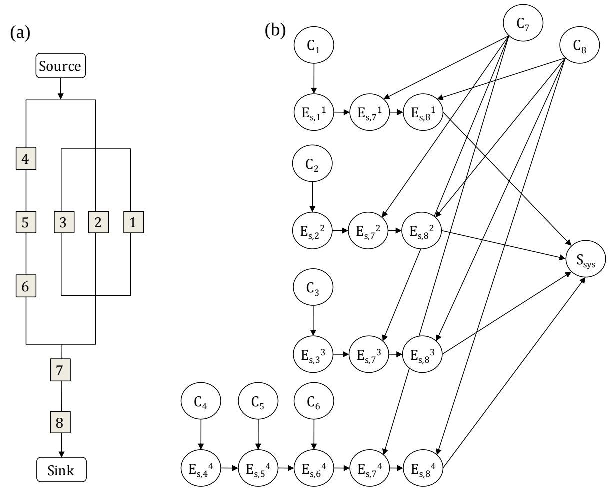

Figure 5: (a) Example system; (b) Efficient MLS formulations for system in (a) with distinct SPSs

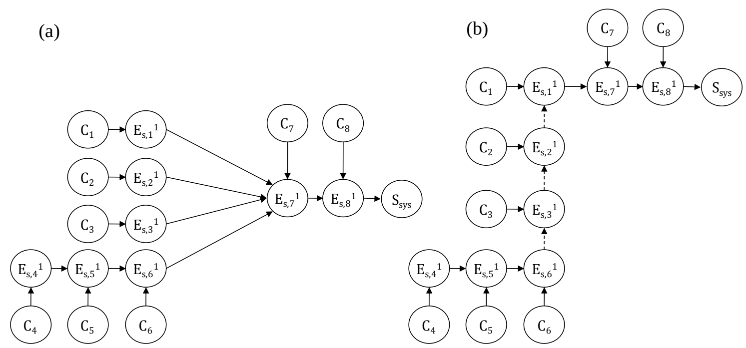

Figure 6: Efficient MLS formulations for the example system with coalesced SPEs associated with components 7 and 8 using (a) a converging structure (b) a chain structure

A notational change has been introduced in Figure 6a: the superscript associated with each SPE node, which previously indicated the MLS number, now represents the instance of the SPE, i.e. if multiple SPEs are associated with the same component, then they are recognized as different instances of the SPE and are distinguished through the superscript. Because, for this system, each component is associated with only one SPE, all superscripts in Figure 6a are 1.

It is noted that node $E_{s,7}$ in Figure 6a has more than 3 parents. Earlier, it was indicated that chain structures are more efficient than converging structures when the number of parents is greater than 3. Thus, the BN in Figure 6a is further modified by replacing the parallel SPE nodes associated with components 1, 2, 3, and 6 with nodes arranged in a chain, resulting in the BN in Figure 6b, with CPTs defined using the relation

$$\begin{aligned} E_{s,i} &= 1 \text{ if } \left[(C_i = 1) \cap \left(\cup \{ E_{s,Pa(i)'} = 1 \} \right) \right] \cup \{ E_{s,Pa(i)''} = 1 \} \\ &= 0 \quad \text{otherwise} \end{aligned} \tag{4}$$

where $E_{s,Pa(i)'}$ are the SPE nodes that are parent to $E_{s,i}$ before the addition of the chain modification and which remain parents after it; $E_{s,Pa(i)''}$ are the SPE nodes that become parents to $E_{s,i}$ after the chain structure is added (identified by dashed links in Figure 6b).

Thus far, the SPEs in a SPS (FPEs in FPSs) corresponding to a particular MLS (MCS) have been arranged in an arbitrary order. However, for complex systems, the arrangement of the SPEs in the SPSs may strongly influence our ability to coalesce SPEs in multiple SPSs (and analogously for FPEs in FPSs). The order in which SPEs appear can be optimized such that SPEs in as many SPSs as possible are coalesced. As mentioned earlier, this reduces the number of nodes and links in the BN. This optimization problem is described next. For brevity, only the formulation employing SPSs is presented; a dual formulation applies to FPSs and, in fact, the example at the end of this paper will use MCSs/FPSs.

Let $L(i^m, j^n) = 1$ indicate the existence of a directed link from $E_{s,i}^m$ to $E_{s,j}^n$ in the efficient MLS BN formulation and $L(i^m, j^n) = 0$ indicate its absence, where i and j are component indices and m and n are indices denoting the instances of these SPE nodes in the BN. Similarly, let $C_i^m = 1$ indicate a directed link between the node representing component i and node $E_{s,i}^m$ and $S_i^m = 1$ indicate a directed link between $E_{s,i}^m$ and the system node (with $C_i^m = 0$ and $S_i^m = 0$ respectively denoting their absences). $L(i^m, j^n)$, C_i^m and S_i^m are the decision variables in the optimization problem. Formulation of the optimization problem assumes the use of only SPE nodes and a converging structure at the system node. To further increase computational efficiency of the resulting BN, the converging structure at any node with more than 3 SPE nodes as parents is replaced by a chain structure in the manner described in Figure 6b.

The objective of the optimization problem is to minimize the number of links in the BN, i.e.

$$\min\left[\sum_{i=1}^{N_c}\sum_{j=1}^{N_c}\sum_{m=1}^{N_I}\sum_{n=1}^{N_I} L(i^m, j^n) + \sum_{i=1}^{N_c}\sum_{m=1}^{N_I} C_i^m + \sum_{i=1}^{N_c}\sum_{m=1}^{N_I} S_i^m\right] \tag{5}$$

where N_C is the number of components in the system and N_I is the maximum number of instances of any SPE. It is desirable that N_I be as small as possible, but its value is not known prior to solving the optimization problem. Thus, an iterative procedure is used to find the smallest N_I value under which the optimization problem is feasible. The existence of links between the component and SPE nodes as well as between the SPE nodes and the system node are controlled by the arrangement of SPE nodes in the BN. Specifically, $C_i^m = 1$ if node $E_{s,i}^m$ exists in the BN, which occurs if the decision variables indicate a link going into or out of node $E_{s,i}^m$. (A node without links going into or out of it can be removed from the BN.) Mathematically, this constraint is written as

$$\left[\sum_{j=1}^{N_C}\sum_{n=1}^{N_I} L(i^m, j^n) + L(j^n, i^m) \geq 1\right] \Rightarrow C_i^m = 1 \tag{6}$$

Techniques are available for modeling "if-then" and "k-out-of-n" (which are needed later) constraints in numerical optimization (Sarker & Newton 2008). The decision variable $S_i^m = 1$ if node $E_{s,i}^m$ is a terminal node in a SPS, i.e. $E_{s,i}^m$ exists and has no other SPE node as a child. Mathematically, this is written as

$$\left[\sum_{j=1}^{N_C}\sum_{n=1}^{N_I} L(j^n, i^m) \geq 1\right] \cap \left[\sum_{j=1}^{N_C}\sum_{n=1}^{N_I} L(i^m, j^n) = 0\right] \Rightarrow S_i^m = 1 \tag{7}$$

There are two constraints governing the arrangement of the SPE nodes in the BN: (1) each MLS must be represented by a SPS; and (2) no SPS may exist that is not strictly a MLS. If the first constraint is violated, then one or more MLSs are excluded resulting in overestimation of the system failure probability. If the second constraint is violated, then the BN will include one or more fictitious MLSs and thus underestimate the system failure probability.

The first constraint requires that each MLS be represented as a SPS, i.e. at least one *permutation* of the SPEs associated with the components in each MLS must be connected as a chain. Define MLS_i to be the set of components contained in the i^{th} MLS and let $N_{MLS,i}$ be the number of components in MLS_i. For the system in Figure 5a: $N_{MCS,1} = N_{MCS,2} = N_{MCS,3} = 3$ and $N_{MCS,4} = 5$. Let P_i be the set of permutations, without replacement, of the components in MLS_i and define $p_i^\alpha = \{p_{i,1}^\alpha, p_{i,2}^\alpha, \ldots, p_{i,N_{MCS,i}}^\alpha\}$ as the α^{th} permutation contained in the set P_i. As an example, for the system in Figure 5a, $P_1 = [p_1^1 = \{8,7,1\}, p_1^2 = \{8,1,7\}, p_1^3 = \{7,8,1\}, p_1^4 = \{7,1,8\}, p_1^5 = \{1,7,8\}, p_1^6 = \{1,8,7\}]$.

Next, let Q_i be the set of permutations with replacement of $N_{MCS,i}$ draws from the index set $\{1, \ldots, N_I\}$. Define $q_i^\beta = \{q_{i,1}^\beta, q_{i,2}^\beta, \ldots, q_{i,N_{MLS,i}}^\beta\}$ as the set of instance indices ordered according to the β^{th} member of Q_i. Using the same example and assuming $N_I = 2$, we have $Q_1 = \{q_1^1 = (1,1,1),\ q_1^2 = (1,1,2),\ q_1^3 = (1,2,1),\ q_1^4 = (1,2,2),\ q_1^5 = (2,1,1),\ q_1^6 = (2,1,2),\ q_1^7 = (2,2,1),\ q_1^8 = (2,2,2)\}$. Note that P_i has $N_{MLS,i}!$ members, while Q_i has $N_I^{N_{MLS,i}}$ members.

Define a set $r_i^{\alpha,\beta} = [r_{i,1}^{(\alpha,\beta)}, r_{i,2}^{(\alpha,\beta)}, \ldots, r_{i,N_{MLS,i}}^{(\alpha,\beta)}]$ which combines the elements of p_i^α and q_i^β. Specifically, $r_i^{\alpha,\beta}$ includes the set p_i^α with superscripts given by the set q_i^β. For the example system, $r_1^{1,1} = \{8^1, 7^1, 1^1\}$, $r_1^{1,2} = \{8^1, 7^1, 1^2\}$, $r_1^{2,4} = \{8^1, 1^2, 7^2\}$, etc. Overall, for this specific MLS, there are $3! * 2^3 = 48$ possible ways to arrange the component numbers given by p_i^α and the subscripts given by q_i^β.

For convenience, define the sum $X_i^{(\alpha,\beta)} = \sum_{l=1}^{N_{MLS,i}-1} L[r_{i,l}^{(\alpha,\beta)}, r_{i,l+1}^{(\alpha,\beta)}]$, where $r_{i,l}^{(\alpha,\beta)}$ is the l^{th} element of $r_i^{(\alpha,\beta)}$. $X_i^{(\alpha,\beta)} = N_{MLS,i} - 1$ only if the SPEs corresponding to the components in

MLS_i form a SPS in the order specified by p_i^{α} and with instance indices assigned by q_i^{β}. For a required SPS to exist in the BN, $X_i^{(\alpha,\beta)} = N_{MLS,i} - 1$ for *at least one* component/instance index ordering from the set $r_i^{(\alpha,\beta)}$. The constraint is written as

$$\max_{\alpha,\beta} X_i^{(\alpha,\beta)} \geq N_{MLS,i} - 1 \qquad \forall i, \qquad \alpha = 1, \dots, N_{MLS,i}!, \qquad \beta = 1, \dots, N_I^{N_{MCS,i}} \tag{8}$$

The second constraint requires that no SPS exist in the BN which does not correspond to a MLS. Consider the BN shown in Figure 7a. Let the shaded nodes ($E_{s,1}^1 \rightarrow E_{s,2}^1 \rightarrow E_{s,3}^1 \rightarrow E_{s,4}^1$) represent a particular permutation of components/instance indices $r_i^{(\alpha,\beta)} = \{1^1, 2^1, 3^1, 4^1\}$ resulting in a valid SPS. Constraint (2) must prohibit a SPE $E_{s,j}^n$, for any n, from "branching-off" the SPS at any point (i.e. being a child of any node in the chain) unless the component j exists in a MLS with all preceding components in the sequence. For example, in Figure 7a, $E_{s,j}^n$ cannot exist as a child of $E_{s,3}^1$ unless components 1,2,3, and j all exist together in a MLS. If components 1,2,3, and j do not exist in a MLS, then the false survival path shown by nodes with dashed edges is introduced into the BN. The associated constraint is written as

$$\begin{gathered} \left[X_i^{(\alpha,\beta)} = N_{MLS,i} - 1, \forall i, \alpha, \beta\right] \Rightarrow \\ L\left[r_{i,l}^{(\alpha,\beta)}, j^n\right] = 0, \qquad \forall j\text{: } \left\{p_{i,1}^{(\alpha)}, \dots, p_{i,l}^{(\alpha)}, j\right\} \notin MLS_m, \forall n, \forall m, \forall l \end{gathered} \tag{9}$$

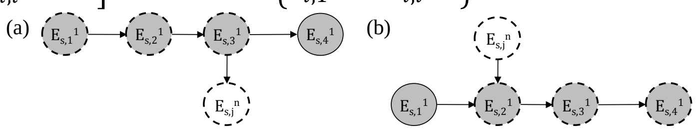

Figure 7: BN used to illustrate constraint (2)

Furthermore, the constraint must prohibit SPE $E_{s,j}^n$, for any n, from being a parent to any node in a valid SPS, unless component i exists in a MLS with all subsequent components in the sequence. For example, in Figure 7b, $E_{s,j}^n$ cannot be a parent of $E_{2,1}$ unless components 2,3,4, and j all exist together in a MLS. The second constraint takes the form

$$\begin{gathered} \left[X_i^{(\alpha,\beta)} = N_{MLS,i} - 1, \forall i, \alpha, \beta\right] \Rightarrow \\ L\left[j^n, r_{i,l}^{(\alpha,\beta)}\right] = 0 \qquad \forall j\text{: } \left\{j, p_{i,l}^{(\alpha)}, \dots, p_{i,N_{MLS,i}}^{(\alpha)}\right\} \notin MLS_m, \forall n, \forall m, \forall l \end{gathered} \tag{10}$$

The combination of these two requirements along with the objective function (the minimization of which ensures links that are not necessary for constructing the required SPSs are not in the BN) prohibits invalid SPSs in the BN. Combining (9) and (10) results in the constraint

$$\begin{gathered} \left[X_i^{(\alpha,\beta)} = N_{MLS,i} - 1, \forall i, \alpha, \beta\right] \Rightarrow \\ \sum_{m=1}^{N_{MLS}} \sum_{n=1}^{N_I} \sum_{l=1}^{N_{MLS,i}} \sum_{\forall j: \left\{p_{i,1}^{(\alpha)}, \dots, p_{i,l}^{(\alpha)}, j\right\} \notin MLS_m} L\left[r_{i,l}^{(\alpha,\beta)}, j^n\right] + \\ \sum_{m=1}^{N_{MLS}} \sum_{n=1}^{N_I} \sum_{l=1}^{N_{MLS,i}} \sum_{\forall j: \left\{j, p_{i,l}^{(\alpha)}, \dots, p_{i,N_{MLS,i}}^{(\alpha)}\right\} \notin MLS_m} L\left[j^n, r_{i,l}^{(\alpha,\beta)}\right] = 0 \end{gathered} \tag{11}$$

The integer optimization problem described above requires consideration of permutations of components. Consequently, it becomes difficult to solve this problem in practice for large systems. To overcome this problem, several heuristics have been developed to reduce the size of the optimization problem that must be considered. Specifically, groups of components are considered as single "super-components," thus reducing the number of components, and measures are taken to eliminate unnecessary permutations of component indices. These heuristics will be described in an extended version of this paper.

To briefly illustrate the computational advantages of using the proposed efficient BN formulation, consider the structural system in Figure 8a consisting of 10 labeled components which can fail. The system has 11 MCSs: {1,2}, {3,4}, {1,3,10}, {1,4,10}, {2,3,10}, {2,4,10}, {5}, {6}, {7}, {8}, and {9}. The BN obtained using the optimization algorithm (including the use of heuristics) is shown in Figure 8b. An example of a heuristic adaption is observed when the single component MCSs are arranged as a series system using SPE nodes on the left of Figure 8b. Because this BN uses a mixture of FPE and SPE nodes, the relations between nodes require modifications of the equations given above. The total clique table size associated with this BN obtained using the optimization scheme is 164. The total clique table size of the MLS BN formulation with the converging structure is 5,140. Thus, the optimized BN is over an order of magnitude computationally more efficient.

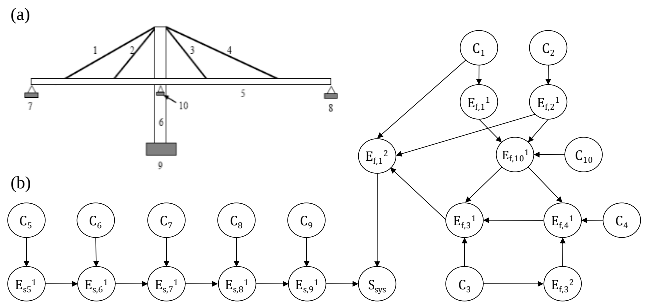

Figure 8: (a) Example structural system; (b) system performance BN

3 CONCLUSIONS

This paper develops efficient BN formulations for modeling the performance of general systems for which the MLSs or MCSs are known. We show that BNs with nodes in chain structures are more efficient than when nodes are arranged in converging topologies. We demonstrate how BNs with nodes arranged in chain structures are constructed for series and parallel systems. This idea is then extended to develop similar topologies for general systems. Finally, a binary integer optimization program is developed with the goal of automatically constructing the BN such that the number of links is minimized. An example highlights the advantage gained from the efficient BN formulation.

REFERENCES

Bayraktarli, Y.Y., Yazgan, U., Dazio, A. & Faber, M., 2006. Capabilities of the Bayesian probabilistic networks approach for earthquake risk management. In *Proceedings of First European Conference on Earthquake Engineering and Seismology*. Geneva, Switzerland.

Bensi, M., Der Kiureghian, A. & Straub, D., 2010. Bayesian network for seismic infrastructure risk assessment: challenges and opportunities. In *Joint Conference Proceedings:7th International Conference on Urban Earthquake Engineering and 5th International Conference on Earthquake Engineering*. Tokyo Institute of Technology, Tokyo, Japan.

Bensi, M., Straub, D., Friis-Hansen, P. & Der Kiureghian, A., 2009. Modeling infrastructure system performance using BN. In *Proceedings of the 10th International Conference on Structural Safety and Reliability*. Osaka, Japan.

Friis-Hansen, P., 2004. Structuring of complex systems using Bayesian Networks (with additional material from the unpublished extension to the paper, 2008). In O. Ditlevsen & P. Friis-Hansen, eds. *Proceedings of Two Part Workshop at DTU*. Technical University of Denmark.

Jensen, F.V. & Nielson, T.D., 2007. *Bayesian Networks and Decision Graphs* second., New York: Springer Science Business Media|Springer.

Kjaerulff, U.B. & Madsen, A.L., 2008. *Bayesian Networks and Influence Diagrams: A Guide to Construction and Analysis*, New York: Springer Science+Business Media, LLC.

Mahadevan, S., Zhang, R. & Smith, N., 2001. Bayesian networks for system reliability reassessment. *Structural Safety*, 23, 231-251.

Sarker, R.A. & Newton, C.S., 2008. *Optimization Modelling: a Practical Approach*, Boca Raton, Florida: CRC Press, Taylor and Francis Group, LLC.

Straub, D., 2005. Natural hazards risk assessment using Bayesian networks. In *Proceedings of 9th International Conference on Structural Safety and Reliability (ICOSSAR 05), Rome, Italy*.

Torres-Toledano, J. & Sucar, L., 1998. Bayesian Networks for Reliability Analysis of Complex Systems. In *Proceedings of the 6th Ibero-American conference on AI (IBERAMIA 98).Lecture notes in artificial intelligence, vol. 1484*. Springer-Verlag Berlin Heidelberg, pp. 195-206.