Bayesian Networks in Reliability

Author: Helge Langseth (Helge.Langseth@sintef.no)
Author: Luigi Portinale (portinal@di.unipmn.it)

The University of Piemonte Orientale Department of Computer Science Research Technical Reports are available via WWW at URL `http://www.di.mfn.unipmn.it/`.
Plain-text abstracts organized by year are available in the directory

Recent Titles from the TR-INF-UNIPMN Technical Report Series

2004-08 *Modelling a Secure Agent with Team Automata*, Egidi, L., Petrocchi, M., July 2004.

2004-07 *Making CORBA fault-tolerant*, Codetta Raiteri D., April 2004.

2004-06 *Orthogonal operators for user-defined symbolic periodicities*, Egidi, L., Terenziani, P., April 2004.

2004-05 *RHENE: A Case Retrieval System for Hemodialysis Cases with Dynamically Monitored Parameters*, Montani, S., Portinale, L., Bellazzi, R., Leonardi, G., March 2004.

2004-04 *Dynamic Bayesian Networks for Modeling Advanced Fault Tree Features in Dependability Analysis*, Montani, S., Portinale, L., Bobbio, A., March 2004.

2004-03 *Two space saving tricks for linear time LCP computation*, Manzini, G., February 2004.

2004-01 *Grid Scheduling and Economic Models*, Canonico, M., January 2004.

2003-08 *Multi-modal Diagnosis Combining Case-Based and Model Based Reasoning: a Formal and Experimental Analisys*, Portinale, L., Torasso, P., Magro, D., December 2003.

2003-07 *Fault Tolerance in Grid Environment*, Canonico, M., December 2003.

2003-06 *Development of a Dynamic Fault Tree Solver based on Coloured Petri Nets and graphically interfaced with DrawNET*, Codetta Raiteri, D., October 2003.

2003-05 *Interactive Video Streaming Applications over IP Networks: An Adaptive Approach*, Furini, M., Roccetti, M., July 2003.

2003-04 *Audio-Text Synchronization inside mp3 file: A new approach and its implementation*, Furini, M., Alboresi, L., July 2003.

2003-03 *A simple and fast DNA compressor*, Manzini, G., Rastero, M., April 2003.

2003-02 Engineering a Lightweight Suffix Array Construction Algorithm, Manzini, G., Ferragina, P., February 2003.

2003-01 *Ad Hoc Networks: A Protocol for Supporting QoS Applications*, Donatiello, L., Furini, M., January 2003.

2002-06 *Stochastic modeling, analysis techniques and tools for dependable reactive systems*, Codetta Raiteri, D., Bobbio, A., October 2002.

Bayesian Networks in Reliability

Helge Langseth
Department of Mathematical Sciences,
Norwegian University of Science and Technology,
N-7491 Trondheim, Norway.

Luigi Portinale
Department of Computer Science,
University of Eastern Piedmont "Amedeo Avogadro",
15100 Alessandria, Italy

Abstract

Over the last decade, Bayesian Networks (BNs) have become a popular tool for modelling many kinds of statistical problems. We have also seen a growing interest for using BNs in the reliability analysis community. In this paper we will discuss the properties of the modelling framework that make BNs particularly well suited for reliability applications, and point to ongoing research that is relevant for practitioners in reliability.

Keywords: Bayesian networks, reliability analysis, modelling, causality.

1 Introduction

A typical task for the reliability analyst is to give input to a decision problem. An example can be to examine the effect that environmental conditions have on a component's time to failure, and give this as input to a maintenance optimization problem. As the quantities in such studies are uncertain or due to random fluctuations, the end result should be a statistical model describing a set of random variables. This model must be mathematically sound, and at the same time easy to understand for the decision maker. Furthermore, such models require a set of parameters to be fully specified, and either statistical data or expert judgement must be used to estimate them. Since both these sources of information can have low quality, as well as come with a cost, one would like the formalism to minimize the number of parameters required by the model. Finally, the model must be represented such that the quantities we are interested in can be calculated efficiently.[1] In a statistical setting, the numbers we would like to find are either conditional probabilities (e.g., the probability that a component will survive for more than one year in a given environment), or deduced from these numbers (for instance the expected life-length of the component).

All of these requirements have lead to reduced focus on traditional frameworks like fault trees, and more flexible modelling frameworks have received increased attention. One such framework which has gained popularity over the last decade is the set of Bayesian Network (BN) models. The history of BNs in reliability can (at least) be traced back to Barlow [1] and Almond [2]. More recently, BNs have found applications in, e.g., software reliability [3, 4], fault finding systems [5, 6, 7, 8, 9, 10], maintenance modelling [11, 12], and general reliability modelling [13, 14, 15, 16, 17, 18, 19, 20, 21, 22]. In particular, common aims and goals

---

[1]This efficiency is of utter importance. One of the current trends in reliability analysis is that the complexity of the technical systems analyzed is growing, and that one at the same time is increasingly interested in including the systems' surroundings in terms of, e.g., organizational factors into the overall reliability assessments. The result is complex models, and often prohibitively time-consuming calculations.

are currently being recognized by researchers in classical reliability theory and researchers in the Bayesian Network community, and examples of fields of fruitful cooperation include probabilistic inference for fault detection or identification, monitoring, maintenance, and prediction (see for instance the recent *Bayesian Applications Modeling Workshops* in 2003 and 2004 [23, 24] held in connection with the conferences on *Uncertainty in Artificial Intelligence*).

We see a partiality to discrete variables in the BN community, mainly due to the technicalities of the calculation scheme (see Section 5). We note that the BNs' applicability in reliability analysis would be enormously limited if one would only consider discrete variables, and we will therefore not limit our attention in this way. We will rather embrace models containing both continuous as well as discrete variables, and from that point of view we will pinpoint some interesting research directions for the BN theorists.

In this paper we will consider BNs in some detail, mention the most prominent reasons for the increasing popularity of BN models, and briefly scratch the surface when it comes to current trends in BN research. We will focus on research directions that will have a direct influence on the applicability of BNs in reliability analysis. Much of the discussion will be related to simple examples.[2] The paper is organized as follows: We start by giving the basics of the BN framework in Section 2, and consider BN modelling in Section 3. BNs are often given a causal interpretation, and this is covered in Section 4. An important feature of the BN framework is the calculation algorithms, which we consider in Section 5. Finally, we give some examples related to a real-life model in Section 6, and offer some conclusions in Section 7.

2 Bayesian Networks

A Bayesian Network [25, 26, 27] is a compact representation of a multivariate statistical distribution function. A BN encodes the probability density function governing a set of n random variables $\boldsymbol{X} = (X_1, \ldots, X_n)$ by specifying a set of conditional independence statements together with a set of conditional distribution functions (CDFs). More specifically, a BN consists of a qualitative part, a *directed acyclic graph* where the nodes mirror the random variables, and a quantitative part, the set of CDFs. An example of a BN over the variables $\boldsymbol{X} = (X_1, \ldots, X_5)$ is shown in Figure 1, only the qualitative part is given.

The driving force when making BN models is the set of *conditional independence* statements the model encodes. We will use the notation $\mathcal{X} \perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$ to denote that the random variables in the two sets $\mathcal{X}$ and $\mathcal{Y}$ are conditionally independent given the variables in $\mathcal{Z}$. If $\mathcal{Z}$ is the empty set, we simply write $\mathcal{X} \perp\!\!\!\perp \mathcal{Y}$ to denote that the sets $\mathcal{X}$ and $\mathcal{Y}$ are marginally independent. Note that we may sometimes use the term *conditional* independence even when the conditioning set is empty. We use $\mathcal{X} \not\!\perp\!\!\!\perp \mathcal{Y} | \mathcal{Z}$ to make explicit that $\mathcal{X}$ and $\mathcal{Y}$ are conditionally *dependent* given $\mathcal{Z}$.

The qualitative part of the BN is used to encode the conditional independence statements, but before we present the mathematical properties of the BN structure we need some notation: We call the nodes with outgoing edges pointing into a specific node the *parents* of that node, and say that X_j is a *descendant* of X_i if and only if there exists a directed path from X_i to X_j in the graph. In Figure 1, X_1 and X_2 are the parents of X_3, written $\mathrm{pa}\,(X_3) = \{X_1, X_2\}$ for short. Furthermore, $\mathrm{pa}\,(X_4) = \{X_3\}$ and since there are no directed path from X_4 to any of the other nodes, the descendants of X_4 are given by the empty set and, accordingly, its non-descendants are $\{X_1, X_2, X_3, X_5\}$.

The edges of the graph represents the assertion that a variable is conditionally independent of its non-descendants in the graph given its parents in the same graph. The graph in Figure 1 does for instance assert that for all distributions compatible with it, we have that X_4 is conditionally independent of $\{X_1, X_2, X_5\}$ when conditioned on $\{X_3\}$, $X_4 \perp\!\!\!\perp \{X_1, X_2, X_5\} | X_3$. Another example is obtained by looking at X_1: $\mathrm{pa}\,(X_1) = \emptyset$, and the descendants of X_1 are $\{X_3, X_4, X_5\}$, so its only non-descendant is $\{X_2\}$. This gives us that $X_1 \perp\!\!\!\perp X_2$ in this model.

---

[2]We want to emphasize that these examples are just simple illustrations, and not considered full-blown reliability models. We will, for instance, take the liberty to use quite loosely defined parameters (like "Quality of an item"), and may not always put effort into operationalizing them.
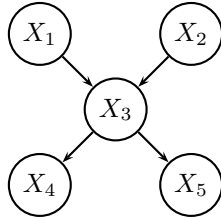
Figure 1: An example BN over the nodes $\{X_1, \ldots, X_5\}$. Only the qualitative part of the BN is shown.

All conditional independence statements can be read off a BN structure by using the rules of *d-separation* [25]. The general analysis of d-separation centers around the three categories of network fragments shown in Figure 2. The serial connection (part (a)) encodes that $X_1 \perp\!\!\!\perp X_3 \mid X_2$, but $X_1 \not\perp\!\!\!\perp X_3$ marginally. For an example, let X_1 denote the *planned* preventive maintenance (PM) program for a given component, let X_2 be the *implemented* PM, and X_3 the life-length of the component. The conditional independence statements encoded in this model tells us that if we do not know the implemented PM program, then the planned PM can tell us something about the life-length of the component ($X_1 \not\perp\!\!\!\perp X_3$). However, as soon as we learn about the implemented PM program, the *plans* are irrelevant for the life-length: $X_1 \perp\!\!\!\perp X_3 \mid X_2$.

Part (b), a diverging connection, dictates similar properties: $Y_1 \perp\!\!\!\perp Y_3 \mid Y_2$, but $Y_1 \not\perp\!\!\!\perp Y_3$ marginally. This BN can for instance model the quality of the production from an assembly line. Let Y_1 be the quality of the first item that was produced at this line, and Y_3 be the quality of the second item. Finally, let Y_2 be a measure of how well the assembly line operates overall. Now, if Y_2 is unknown to us, information about Y_1 being good (bad) would make us infer that the quality of the line as such (Y_2) was good (bad) as well, and finally that Y_3 therefore would be good (bad) too. Thus, $Y_1 \not\perp\!\!\!\perp Y_3$. On the other hand, if the quality of the production line is known, the quality of each produced item may be seen as independent of each other ($Y_1 \perp\!\!\!\perp Y_3 \mid Y_2$).

Finally, the converging connection in part (c) encodes that Z_1 and Z_3 are marginally independent, but $Z_1 \not\perp\!\!\!\perp Z_3 \mid Z_2$. To understand this part, one can think of Z_1 as the quality of an assembly line, Z_3 as the environmental conditions of the production (temperature, humidity, etc.), and Z_2 as the quality of a product coming from the assembly line. The quality of the assembly line is a priori independent of the environmental conditions ($Z_1 \perp\!\!\!\perp Z_3$), however, as soon as we observe the quality of the product, we can make inference regarding the quality of the line from what is known about the environmental conditions ($Z_1 \not\perp\!\!\!\perp Z_3 \mid Z_2$).
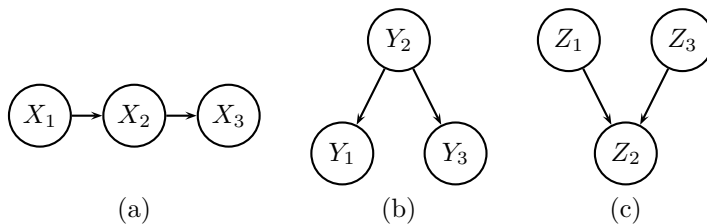

Figure 2: Three small network fragments describing different structural situations: Serial (a), diverging (b) and converging(c).

When it comes to the quantitative part, we will use $f(\boldsymbol{x} \mid \boldsymbol{y})$ to denote the CDF of $\boldsymbol{x}$ given $\boldsymbol{y}$. The same notation is used whether $\boldsymbol{x}$ is a vector of discrete or continuous (or mixed) variables. We will sometimes call $f(\boldsymbol{x} \mid \boldsymbol{y})$ a CDF even if $\boldsymbol{y}$ is empty, but will use $f(\boldsymbol{x})$ as a shortcut for $f(\boldsymbol{x} \mid \emptyset)$.

Now, each variable is described by the conditional probability function of that variable *given its parents* in the graph, i.e., the collection of conditional probability functions $\left\{ f(x_i \mid \mathrm{pa}\,(x_i)) \right\}_{i=1}^{n}$. The underlying

assumptions of conditional independence encoded in the graph allow us to calculate the joint probability function as

$$f(x_1,, \ldots, x_n) = \prod_{i=1}^{n} f(x_i \,|\, \mathrm{pa}\,(x_i)), \tag{1}$$

and this is in fact the main point when working with BNs: Assume that a distribution function $f(x_1,, \ldots, x_n)$ factorizes according to Eq. 1. This defines the parent set of each X_i, which in turn defines the graph, and from the graph we can read off the conditional independence statements encoded in the model. Hence, the graphical representation is the bridging of the gap between the (high level) conditional independence statements we want to encode in the model and the (low level) constraints this enforces on the conditional distribution function.

To fully specify the set of CDFs, we must (i) select parametric families for each $f(x_i \,|\, \mathrm{pa}\,(x_i))$, and ($ii$) determine values for all parameters of each CDF. To do the last part, we must use either statistical data or expert judgement, but since both of these sources of information can have low quality, as well as come with a cost, one would like the BN-formalism to minimize the number of parameters required. This is exactly what it attempts to do; it represents the multi-dimensional distribution in a cost-efficient manner (through its factorized representation in Eq. 1). Let us consider the domain given by $\boldsymbol{X} = (X_1, \ldots, X_n)$, and let the nodes be labelled such that $\mathrm{pa}\,(X_i) \subseteq \{X_1, \ldots, X_{i-1}\}$ for $i = 1, \ldots, n$ (this can always be obtained through relabelling). Although it is not essential for the following argument, we will for now assume that all variables take on values in $\{0, 1\}$ (these values could for instance signify a component being either failed or operating). According to Eq. 1 we will require conditional probability distributions of the type $f(x_i \,|\, \mathrm{pa}\,(x_i))$; these are in this case fully determined by the probability $P(X_i = 1|\mathrm{pa}\,(x_i))$. The number of parameters required to specify $f(x_i \,|\, \mathrm{pa}\,(x_i))$ for a given i is thus equal to $2^{|\mathrm{pa}(X_i)|}$, and the total number of parameters is given by $\sum_{i=1}^{n} 2^{|\mathrm{pa}(X_i)|}$. This number is by construction not higher than $2^{i} - 1$, that is the number of parameters required to define the full joint distribution (without using the conditional independence statements in the BN structure). Hence, when we use a BN, we are never worse off than if the full joint distribution is defined directly, and we are bound to improve as long as at least one conditional independence assumption can be made. As an example, if all variables of the domain in Figure 1 are binary, then the full joint distribution will require 31 parameters to be specified. Alternatively, the BN representation uses only 10 parameters.

3 Building BN models

When we want to build a BN, we rely on two sources of information: Input from domain experts and statistical data. In this section we will briefly describe the basics when building BN models.

3.1 Expert elicitation

3.1.1 The main process

Building BNs from expert input can be a difficult and time consuming task. This is typically an assignment given to a group of specialists. A BN expert guides the model building, asks relevant questions, and explains the assumptions that are encoded in the model to the rest of the group. The domain experts, on the other hand, supply their knowledge to the BN expert in a structured fashion. In our experience, it will pay off to start the model building by familiarization. The BN expert should learn a bit about the domain under study, and the reliability analysts would benefit from having a (limited) knowledge about BNs. As soon as this is established, model building will proceed through a number of phases:

Step 1 − **Defining variables:** Select the important variables in the model. The *range* of the continuous variables and the *states* of the discrete variables are also determined at this point.

Step 2 − **The qualitative part:** The next task is to define the graphical structure that connects the variables. In this phase it can be beneficial to consider the edges in the graph as *causal*, but the trained

domain expert may also be confident about dependencies/independencies to include in the model. Domain experts can often be very eager to incorporate unpractically many links in the structure in an attempt to "get it right". The BN expert's task is in this setting to balance the model complexity with the modelling assumptions the domain experts are willing to accept. Often, a postprocessing of the structure may reveal void edges (e.g., those creating triangles in the graph [28]).

Step 3 – The quantitative part: To define the quantitative part, one must select distributional families for all variables, and fix parameters to specify the distributions. If the BN structure has not been carefully elicited (and pruned) this may be a formidable task. Luckily, the consistency problems common when eliciting large probability distribution functions are tackled by a "divide-and-conquer" strategy here: If each conditional distribution function $f(x_i \mid \mathrm{pa}\,(x_i))$ is defined consistently, then this implies global consistency as well [29]. To elicit the quantitative part from experts, one must acquire all conditional distribution functions $\{f(x_i \mid \mathrm{pa}\,(x_i))\}_{i=1}^{n}$ in Eq. 1, and once again the causal interpretation can come in as a handy tool. Alternatively, the expert can supply a mix of both marginal and conditional distributions, which can then be glued together by the IPFP algorithm [30, 31].

Step 4 – Verification: Verification should be performed both through sensitivity analysis as well as by testing how the model behaves when analyzing well-known scenarios. Typically, this step gives need for refinement/redefinition of the model, and this is repeated until the work put into improving the model does not lead to substantial benefits. As pointed out by, e.g., Druzdzel and van Gaag [32], the sensitivity wrt. BN structure is relatively large, and the graph is thus the most vital part. Sensitivity wrt. the parameters that we put in is in large dependent on the application.

Lately, some tools that are aimed at guiding the model-building have emerged. These tools attempt to enable a domain expert to build a BN without interacting with a BN expert. For instance, Skaanning [33] describes a system that can be used to build troubleshooter systems efficiently.

3.1.2 Building large models

One conceptually nice feature when building BNs, is that a BN structure can be seen as built up by smaller pieces following a "Lego brick" perspective. An example is given in Figure 3, where a finite mixture model is given in part (a). We follow the convention that random variable that cannot be observed are represented by *shaded* nodes in the graph. The (unobserved, discrete) M determines the distribution over the random vector $\boldsymbol{Y}$. Part (b) gives the BN structure of a classical factor analysis model. Note that it is simple to read the underlying model assumptions off this graph: For instance, we have that Y_i is a non-descendant of Y_j, hence $Y_i \perp\!\!\!\perp Y_j \mid \boldsymbol{X}$. This implies one of the classical assumptions of factor analysis: $\mathrm{Cov}(\boldsymbol{Y} \mid \boldsymbol{X})$ must be a diagonal matrix. Finally, we can piece these two fragments together to obtain the "mixture of factor analyzers" (see, e.g., [34]) in part (c). This is found as a combination of the finite mixture model and the standard factor analysis model, and the structural properties of this model can be seen as a combination of the properties of its building blocks.

This idea of using small and "easy-to-read" pieces as building blocks to create a complex model is an often applied technique when large BNs models are constructed. In the reliability field, this practice can be indirectly applied when considering the use of standard combinatorial tools like Fault Trees (FT). *Fault tree analysis* is very popular among reliability engineers for the analysis of large safety-critical systems [35, 36]. Its goal is to represent the (deterministic) combination of elementary causes, called the *primary events*, that lead to the occurrence of an undesired catastrophic event, the so called *top event*. The construction of an FT eventually reveals the combination of a set of logical gates describing the faulty behavior of the modelled system. It is quite straightforward to map a given FT into an equivalent BN with binary nodes, where FT gates (with their input and output events) are mapped into small fragments of the BN whose combination produce the whole BN corresponding to the given FT [15]. In other words, the modular construction of an
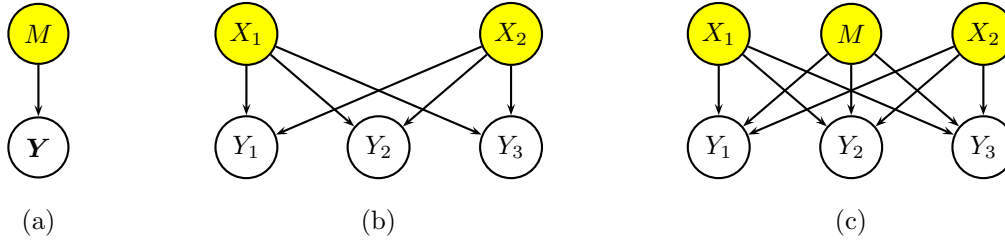
Figure 3: Fragments of BN structures can be used as building blocks to obtain larger structures. Here the finite mixture model in part (a) is combined with the factor analysis model in part (b) to obtain the mixture of factor analyzers in part (c).

FT can be mapped into a modular construction of an equivalent BN.[3] The advantages of a BN over an FT are both at the modelling as well as the analysis side, as we will describe in Section 6 (see [14, 15] for details).

In domains that can approximately be described using an object oriented language [37] we typically find repetitive substructures or substructures that can naturally be ordered in a superclass/subclass hierarchy. For such domains, the domain expert is usually able to provide information about these properties. The basic building blocks available from domain experts examining such domains are information about random variables that are grouped into substructures with high internal coupling and low external coupling. These substructures naturally correspond to instantiations in an object-oriented BN (OOBN) [38, 39]. For instance, an instantiation may correspond to a physical object or it may describe a set of entities that occur at the same instant of time (a dynamic Bayesian network [40] is a special case of an OOBN). Moreover, analogously to the grouping of similar substructures into categories, instantiations of the same type are grouped into classes. As an example, several variables describing a specific pump may be said to make up an instantiation. All instantiations describing the same type of pump are said to be instantiations of the same class. OOBNs offer an easy way of defining BNs in such object-oriented domains s.t. the object-oriented properties of the domain are taken advantage of during model building, and also explicitly encoded in the model.

Parametric extensions of basic FTs called *Parametric Fault Tree* (PFT) have been introduced recently [41]. The kind of problems addressed by PFTs (i.e., modelling redundancy, classes of equivalent components, etc.) appears to be naturally handled by the OOBN framework, as well as other higher order probabilistic formalisms like *probabilistic Horn abduction* [42].

3.1.3 Distributional assumptions

We have already mentioned that BNs can be seen as a way of reducing the burden to fully specify the distribution functions over the variables $(X_1, \ldots, X_n)$. However, the parametrization in a BN is not defined to minimize the number of parameters required for its specification. The BN is merely specified to be sufficient to encode *any* distribution compatible with the conditional independence statements encoded in the graph. This is particularly evident whenever we work with the conditional distribution function of discrete variables having only discrete parents. An example is given in the left pane of Figure 4, where Y has m parents, $\text{pa}(Y) = \{Z_1, \ldots, Z_m\}$. If we assume that each parent has k possible states, and Y has ℓ states, the conditional probability distribution for Y can be seen as a table with ℓ rows and k^m columns. Thus, we have one column in the table per configuration of the parents, and one row per state Y can take. The numbers in each column add up to 1, so it gives us a total of $(\ell - 1) \cdot k^m$ parameters to elicit. Obviously, if k, ℓ or m are "large", the expert elicitation process will become tedious (and possibly quite expensive), and the quality of the elicited numbers may be reduced, e.g., due to expert exhaustion. So, even though BNs may

---

[3]This does not mean that reliability engineers should get rid of FT construction tools, only that they in principle are able to construct BNs out of small building blocks in the same way as they build FTs.

significantly reduce the number of probabilities required to specify the domain, the number of parameters required can still be a serious bottleneck. Many researchers (see, e.g., [43, 44, 45]), have therefore explored even more cost-efficient representations. Common for these approaches is the realization that *all parameters are required if we do not make additional assumptions.* However, if the domain experts are able to identify, e.g., functional relations, then this should be taken into consideration.

In our example, if ℓ is large, but k^m is reasonably small, it can be beneficial to try to define the conditional distribution of Y given its parents as following some distribution family (e.g., Poisson with parameter λ_j, $j = 1, \ldots, k^m$). On the other hand, if ℓ is small, and k^m is large, then one may look in another direction. Recall that the converging structure (part (c) of Figure 2) encodes that the parents are dependent given Y, but makes no assumption regarding what this correlation looks like. If the analyst is able to make some assumptions, then this can be exploited during model building. Of particular importance to the reliability analyst is the *Noisy-OR* relation [46, 47] and its counterpart, the *Noisy-AND*. For the case of this discussion, let $k = \ell = 2$. A Noisy-OR is as its name indicates an extension of the deterministic OR relation. Refer again to the left pane of Figure 4, and assume that we consider to model Y as true if and only if at least one of its children is true. Now, to make deterministic statements like that is often somewhat bold, in particular if we are modelling systems that are not fully shielded from external influences. A more realistic assumption may be to say that whenever Z_i is true it will, with probability $1 - q_i$, force Y to be true as well ($i = 1, \ldots, m$). If Z_i is false it does not contribute.[4] This can be modelled as in the right pane of Figure 4. We introduce new variables Z_i' with the notion that $P(Z_i' = 1 | Z_i = 1) = 1 - q_i$ and $P(Z_i' = 1 | Z_i = 0) = 0$. Next, $Y = 1$ if and only if at least one $Z_i' = 1$. Thus, we can model this (special) conditional distribution $f(y | z_1, \ldots, z_m)$ by using m parameters (q_i, $i = 1, \ldots, m$) instead of the 2^m parameters we would need in general. Notice that the possibility of modelling such non-deterministic interactions among events is one of the advantages of using BNs instead of FTs. Indeed, by translating FT gates into corresponding BN fragments, we are now able to model probabilistic gates (instead of deterministic ones). Note also that this is just one example of how to reduce the burden of eliciting probability distributions from domain experts. Flexible representation like probability trees have also be used with success in many situations [48].
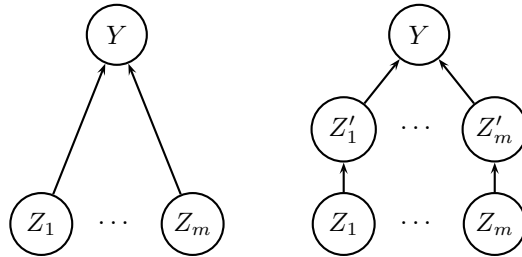

Figure 4: Y has m parents $Z_1, \ldots, Z_m$. If all variables are binary, the full specification of the conditional probability distribution will require the elicitation of 2^m parameters. If additional assumptions can be made, then more efficient structures – like Noisy OR – can be used.

3.2 Using empirical data to obtain a BN model

The Bayesian formalism offers an intuitive way to estimate models based on the combination of statistical data and expert judgement.

---

[4]Assuming that each Z_i does not contribute to the effect Y in case of falsity might not be realistic in situations where some potential causes are missing (i.e., are not modelled for different reasons). In such cases a *leak* or *background probability* can be introduced to summarize the contribution of these missing causes.

For a given graphical structure, estimation of the CDFs was considered in [49]. It was shown how the full posterior distribution over the parameter-space can be obtained in closed form by efficient computations. The *EM-algorithm* [50] is also particularly intuitive in this setting [51].

Structural learning, i.e., to estimate the qualitative structure of a BN (the edges of the graph), can also be done efficiently. Cooper and Herskovits [52] showed how a posterior distribution over the space of directed acyclic graphs can be obtained through efficient computations (see also [53, 54]).

The algorithms mentioned above have been developed for *discrete* BNs, but researchers are continuously working to define efficient algorithms for more general models. Sampling (see in particular [55]) seems to be the most widely applied method for parameter estimation, whereas the state-of-the-art for structural learning still appears to be immature (see, however, [56, 57, 58]).

4 Causal interpretation

We have advocated a causal interpretation of the BN model when it is built based on expert judgement. This gives the domain experts (who sometimes lack thorough statistical training) a language to communicate their knowledge to the BN expert. When the BN model building is finished and the BN is employed for decision support, it would be valuable to maintain this interpretation of the model. We would then move from conclusions like *"In this dataset we find that components that are maintained once a year fail twice as often as those maintained twice a year"* to the much more potent statement *"If we change the PM interval of Component XX-Y (say) from* 12 *to* 6 *months we can expect an increase in its remaining life by a factor of about* 2*"*. Obviously, the latter statement is of more use than the former for, e.g., PM optimization. The recent book by Pearl [59] gives a clear exposition of BNs as causal models, and although statisticians have traditionally been reluctant to the use of causal models,[5] a statistical treatment of causal mechanisms and causal inference in association with BNs is starting to dawn [61, 62]. The main argument against making causal statements is that statistics lack the language to model causal effects. Causal models require an operator that signify *intervention*; that one actively enters into a domain to change one or more of the variables there. Such interventions are in the context of BNs often denoted by the *do-operator*. Consider a BN over the variables $(\boldsymbol{X}, \boldsymbol{Y})$. Equipped with the do-operator, we want to calculate the distribution over the variables $\boldsymbol{X}$ given that we enforce the variables $\boldsymbol{Y}$ to take the value $\boldsymbol{y}^*$, written $f(\boldsymbol{x} \,|\, \mathtt{do}\,(\boldsymbol{Y} \leftarrow \boldsymbol{y}^*))$. This should not be confused with the standard CDF $f(\boldsymbol{x} \,|\, \boldsymbol{Y} = \boldsymbol{y}^*)$, and we use the models in Figure 5 to highlight the crucial difference between the two:
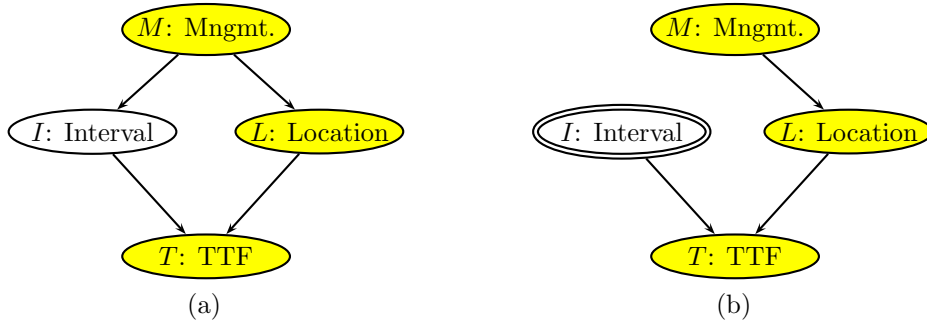

Figure 5: (a) The original BN structure for the life-length model, and (b) the mutilated graph after intervention. The *I*-node is double-lined to signify that this node is subject to intervention.

Assume we want to model the life-length distribution (T) of a given component. The component is

---

[5]As an example, Speed [60] wrote that "considerations of causality should be treated as they have always been treated in statistics: preferably not at all (but if necessary, then with very great care)".

maintained corresponding to a maintenance interval (I) and it is placed at location L. This location may either be outdoors exposed to external environment or inside a cabinet (which in this example would shield the component from environmental stress and therefore increase its expected life-length). Finally, the safety alertness of the management (M) affects the location of the component as well as its maintenance program. In our simple example we presume that a safety-alert management may be more willing to both shield the component and maintain it frequently, although both these actions come with a (potentially high) cost. For the case of this argument, we shall assume that we will not observe L or M. We now get to know that the PM interval is 24 months (rather high for this type of components, say). From this we would infer (Figure 5 a) that the management does not put in much effort to increase the life-length of this component, and from this we would again infer that the component is likely to be exposed to the external environment (as management seems eager to save money, we conclude that it is likely that they also reduce costs with respect to location). This inference will influence the life-length distribution we calculate, as the distribution of T is determined based on both I *and* L.

Next, consider the situation where we go into the plant and decide to set the maintenance interval such that it is maintained every 24 months. We do not ask for permission from the management to do this, and we do not observe any aspect of the domain before we intervene. The chain of inference outlined above is no longer valid, as our intervention into the system does not tell us anything about the preferences of the management. The relation between the variable we intervene with (I) and its parents (here, $\mathrm{pa}\,(I) = \{M\}$) has been overruled. On the other hand, it seems appropriate to assume that I will influence its descendants in the model just as before; $f_T(t|I = i^*, L = l)$ does not change just because I was *forced* to take the value i^*. In our graphical formulation we hence assume that *intervention* in the original BN (Figure 5 a) is identical to *observation* in the *mutilated graph* [59] (Figure 5 b). We use double-line to depict I in the mutilated model; this signifies that I has been set (as opposed to simply observed). The only difference between these two graphs is that all edges pointing into I have been removed in part (b). We shall use $\tilde{f}(\cdot\,|\,\cdot)$ to signify that a CDF is related to the mutilated model, and we note that our assumptions indicate that, e.g., $\tilde{f}(t|I = i, L = l) = f(t|I = i, L = l)$ and $\tilde{f}(m) = f(m)$, but that $\tilde{f}(i)$ is not determined by the above.[6] Calculating $f(t|\mathrm{do}\,(I \leftarrow i^*))$ in the original model is assumed identical to calculating the posterior distribution $\tilde{f}(t|I = i^*)$ in the mutilated model. This way of thinking can be generalized: Consider a BN over the variables $(\boldsymbol{X}, \boldsymbol{Y})$ where we want to calculate the general expression $f(\boldsymbol{x}\,|\,\mathrm{do}\,(\boldsymbol{Y} \leftarrow \boldsymbol{y}^*))$. In this case we mutilate the graph by removing all edges pointing into variables in $\boldsymbol{Y}$, and calculate $\tilde{f}(\boldsymbol{x}\,|\,\boldsymbol{Y} = \boldsymbol{y}^*)$.

More mathematically formulated, we want to establish the set of assumptions required so that $f(\boldsymbol{x}|\mathrm{do}\,(\boldsymbol{Y} \leftarrow \boldsymbol{y}^*))$ can be calculated from the ordinary rules of probability calculus using the mutilated graph. Before we give the general result, we consider the example in Figure 5, where we want to calculate $f(t, l, m\,|\,\mathrm{do}\,(I \leftarrow i^*))$:

$$
\begin{aligned}
f(t, l, m\,|\,\mathrm{do}\,(I \leftarrow i^*)) &= \tilde{f}(t, l, m\,|\,I = i^*) \\
&= \frac{\tilde{f}(t, l, m, i^*)}{\tilde{f}(i^*)} \\
&= \frac{\tilde{f}(t|i^*, l) \cdot \tilde{f}(l|m) \cdot \tilde{f}(i^*) \cdot \tilde{f}(m)}{\tilde{f}(i^*)} \\
&= f(t|i^*, l) \cdot f(l|m) \cdot f(m).
\end{aligned}
\tag{2}
$$

The first equality above is the assumption that intervention in the original model can be treated as observation in the mutilated graph; the second is the definition of conditional probability; the third follows from Eq. 1; the final equality follows from the assumption that most CDFs are equal in the original and the mutilated model. Thus, the assumptions made with respect to modelling the effect of intervention are identical to assuming that the chain of equalities leading to Eq. 2 holds.

---

[6]Due to a technicality we will need that $\tilde{f}(i^*) > 0$ for the value i^* that I is forced to take, but it can otherwise be defined arbitrarily.

We generalize this by stating that $f(\boldsymbol{x} \,|\, \text{do}\,(\boldsymbol{Y} \leftarrow \boldsymbol{y}^*))$ can be calculated in this way if we assume that

$$f(x_1,\ldots,x_n|\text{do}\,(\boldsymbol{Y} \leftarrow \boldsymbol{y}^*)) \;\; = \;\; \prod_{i=1}^{n} f\left(x_i \,|\, \text{pa}\,(x_i)\right) \Bigg|_{\boldsymbol{y}=\boldsymbol{y}^*}. \tag{3}$$

The conditioning on $\boldsymbol{y} = \boldsymbol{y}^*$ simply means that whenever some variable X_i has any of the variables in $\boldsymbol{Y}$ as a parent, then that parent should be given the corresponding value from $\boldsymbol{y}^*$. This was for example the case above, where $f(t|i,l)\big|_{i=i^*} = f(t|i^*,l)$ was required to calculate $f(t,l,m \,|\, \text{do}\,(I \leftarrow i^*))$.

The correctness of the assumption in Eq. 3 is unfortunately not statistically falsifiable from empirical data, and this is presumably the major obstacle for causal analysis to be widely embraced by the statistical community.

Even more problems occur if we have estimated the quantitative part of the BN from data, but still want to use the causal interpretation of BNs. One thing is that we must decide how to model interventions (e.g., based on the assumption of Eq. 3), the other is that we need to be able to assess the strength of the causal relations. As we saw above, the causal strength can be calculated from the set of CDFs in the original BN (as soon as Eq. 3 is assumed true), but this requires that these are known – or can be estimated. This can be justified when the CDFs are obtained through expert elicitation, but difficulties may occur as soon as they are estimated from data. This has been put under detailed investigation by Judea Pearl [63, 59], and we refer the interested reader to those publications. Here, we will restrict ourselves to only scratch the surface of this interesting body of work, and we do that by way of an example from our own experience.

Figure 6 (a) is a fragment of a network that was estimated from a database containing environmental conditions and failure times for a number of mechanical components [64]. The CDFs were estimated from empirical data. Inspired by a causal interpretation of this model, one may believe that the TTF-distribution can be affected simply by planning a different maintenance regime, and that the estimated CDFs can be used to calculate this effect. This is indeed the case as long as the depicted model structure is correct. The mutilated graph is identical to the original one (P does not have any parents), and since the domain contains no unobserved nodes, we can estimate the required probability distribution $f(t|\text{do}\,(P \leftarrow p^*)) = f(t|P = p^*)$ from the dataset.[7] However, if there are doubts regarding the model, we must be careful. From the data we can only find the strength of the *correlation* between the planned maintenance and the TTF, but this could, e.g., be explained by ways of the model in part (b): In this model the safety alertness at management level (M) influences both the maintenance plans as well as the TTF, and therefore introduce a correlation between the two. M is not observed, and we see that P and T are (marginally) dependent even if we remove the edge from P to T (compare with Figure 2 b). From empirical data we cannot separate between the model with the edge $P \rightarrow T$ and the one without it, and since the causal interpretation of these two models are quite different, it is evident that the causal effect is not identifiable. On the other hand, we would have been able to calculate the causal strengths if the data used to estimate the CDFs had been *experimental*.

Finally, if one can agree that the model in part (c) is correct, the story gets another twist: The causal effect the planned maintenance has on the actually performed maintenance (A) can be identified in this model, as can the causal effect that A has on T. Notice that we must assume that A is observed to get this to work. The overall causal effect can then be calculated as

$$f\left(t \,|\, \text{do}\,(P \leftarrow p^*)\right) = \int_a f(a|p^*) \left(\int_p f(t|p,a)f(p)\,dp \right)\, da,$$

see [63] for details.

5 Inference

A pivotal property of any modelling framework is the ability to calculate quantities meaningful for decision support. In our setting these numbers are defined as arbitrary CDFs ($f(x_i, x_j, x_k)$ and $f(x_i, x_j \,|\, x_k, x_\ell)$, say).

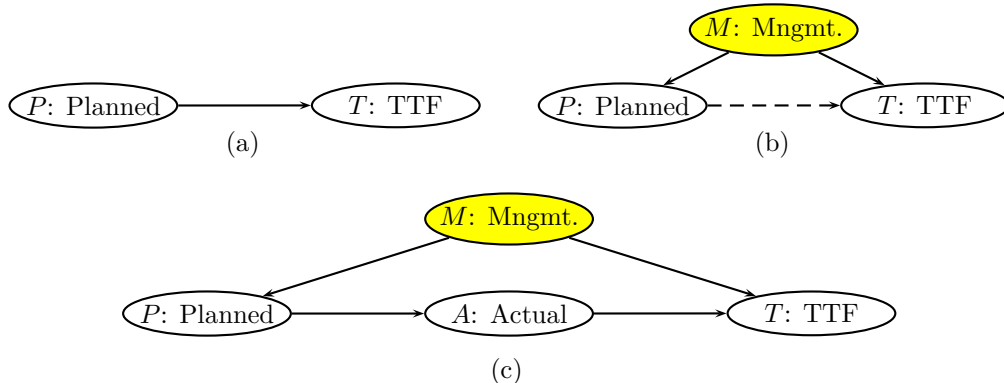[7]A technical detail makes us require that $f(p^*) > 0$.

Figure 6: Part (a) shows a fraction of the BN learned in [64]. There is a dependency between *Planned maintenance program* (P) and *Time to failure* (T). A causal interpretation would imply that changing the plans (and not the implemented maintenance program) would change the life-length distributions. Part (b): We introduce *Management* (M) to obtain an alternative model. The causal strength between P and T is not identifiable from empirical data in this example. Part (c) gives yet another modelling option where the causal effect from P to T is identifiable (using the *front-door criterion* [63]).

This can, for instance, be the survival time distribution for a component given environmental conditions. Fortunately, the conditional independence statements encoded in the BN can be exploited for making these calculations efficiently. Over the last decade we have seen a number of commercially available tools for making calculations with BNs. The practitioner will therefore not need a formal understanding of the algorithms employed, and a detailed description of the inference algorithms is thus beyond the scope of this paper (the interested reader is referred to [27, 65]). We shall, however, look briefly at how one can determine the computational complexity of the inference. This is important, as cleverly designed BN models containing thousands of random variables can be handled efficiently [66], whereas significantly smaller models may be intractable if less thought has been put into the design of the BN.

The first point to be made is that the conditional independence statements are encoded in the *qualitative part* of the BN. As these statements are the main fuel for the inference algorithm, it first works on the graphical representation of the BN. The quantitative part comes into play at a later stage, but we will refrain from covering those details here. It is the graphical structure which (to a great extent) determine the computational complexity. We will tie the discussion to the example in Figure 7.

We start off by considering the BN structure (see the left-most part of Figure 7 where an example network over $\boldsymbol{X} = (X_1, \ldots, X_5)$ is given). The first step is the so-called *moralization* of the structure: For each node in the graph we check if that node has more than one parent. If so, we verify that all the parents are connected, or introduce new undirected edges to connect those that are not. This is done for all nodes, and we thereafter drop the direction of the edges to obtain the *moral graph*. Only X_5 has more than one parent in Figure 7 (a), and as its parents are not directly connected ($X_3 \notin \mathrm{pa}\,(X_4)$ and $X_4 \notin \mathrm{pa}\,(X_3)$) we introduce a new edge between the two. Thereafter the directions of all edges are dropped, and we obtain the structure in part (b).

The next step is called *triangulation*. We look for cycles in the moral graph with length more than 3 that are without a cord. If such a cycle is found, we break it by introducing a cord. In Figure 7 (b) the cordless cycle $X_1 — X_2 — X_4 — X_3 — X_1$ has length 4. An edge between X_1 and X_4 is introduced to break the cycle. Often, one can triangulate a graph in more ways than one, and clever heuristics exist for efficient triangulation [67]. The triangulated graph is shown in Figure 7 (c).

Finally, we find the cliques in the triangulated graph. A clique is a maximal subset of nodes where all nodes in the subset are connected. Hence, $\{X_1, X_2, X_4\}$ is a clique in Figure 7 (d), but for instance $\{X_1, X_2, X_3, X_4\}$
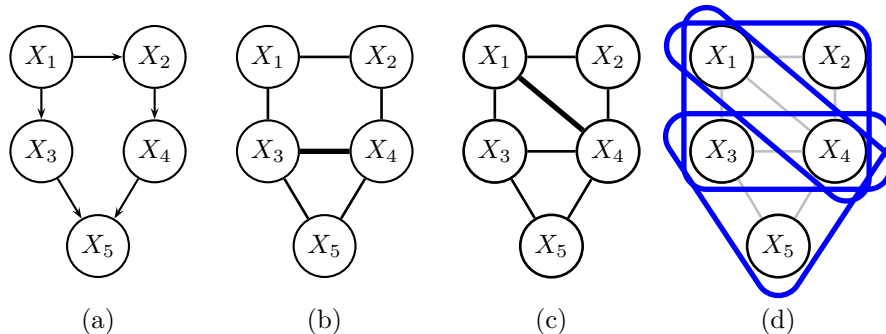
11

Figure 7: The four-step process to create the inference structure: The initial BN (a), the moral graph (b), the triangulated graph (c), and the triangulated graph overlayed the cliques (d).

is not a clique as X_2 and X_3 are not neighbours in the triangulated graph. Actually, there are three cliques in our example: $\{X_1, X_2, X_4\}$, $\{X_1, X_3, X_4\}$ and $\{X_3, X_4, X_5\}$. And now we are at the end of this bumpy road as we can state that the complexity of exact inference. Inference is performed by maintaining functions over the cliques. The cost of maintaining these functions depends on the quantitative part of the BN (more precisely it depends on the distributional families we use) and the cost is for instance exponential in the number of nodes in each clique for discrete variables.[8] This means that it is of great importance to able to keep the cliques as small as possible. Note that for each node, we have that the node together with all its parents will be a subset of at least one clique. Hence, to control the size of the cliques one should remember to control the maximum number of parents nodes in the domain have. There are a number of modelling tricks to obtain this, see, e.g., [27].

Whenever exact inference is infeasible, we may use approximate methods instead. One of the most exciting developments in statistics over the last years is the rise of sampling techniques, in particular Markov chain Monte Carlo methods (see, e.g., [69] for an overview). We propose to use the excellent tool called `BUGS` [70] whenever sampling from a BN. This is a general purpose modelling language, which takes as its input a BN model, and generates arbitrary CDFs as its output.

The complexity of the sampling scheme is determined by the size of what is called the *Markov blanket* for each variable. If we create the moral graph from a BN structure and pick out a node X_i, the Markov blanket of X_i are the set of nodes directly connected to X_i. In Figure 7 (b) we see that, for instance, the Markov blanket of X_3 is $\{X_1, X_4, X_5\}$. When `BUGS` does its sampling, it will for each X_i generate samples from the CDF of X_i given its Markov blanket. The cost of generating each of these samples depends on the families of distributions we have put into the BN, for example is the cost exponential in the size of the Markov blanket for discrete variables. We note that the Markov blanket for X_i will at least include all of X_i's parents, so it is beneficial to keep track of the number of parents for all variables also when sampling is used.

6 FT-like analysis with Bayesian networks

Fault Tree Analysis (FTA) is one of the most popular techniques used by reliability engineers. The aim of this section is to give some ideas regarding how BNs can be used to improve modelling of systems considered by FTA. In particular, BNs may include features like:

---

[8]BNs as we have described them in this paper can represent any distribution function, but they have traditionally been mostly used to handle discrete distributions. The reason for this is that the algorithm for exact inference utilizes some properties of the discrete distributions that are not found in general continuous distributions. Recently, however, Moral et al. [68] showed that also the *mixtures of truncated exponential* (MTE) distributions possess the required properties, and they proved that *any* distribution can be approximated arbitrarily well by MTEs. This is important, as it relieves the BN modeler from having to work with only discrete (or discretized) variables. Now she can rather model the variables in the domain as she thinks best.

- Uncertainty on local dependencies (i.e., probabilistic gates; Section 6.4);

- Multi-state variables (i.e., multiple behavioral modes; Section 6.5);

- Uncertainty on model parameters (Section 6.6);

- Dependence between components (e.g., introduced by a common environment; Section 6.7).

Given a standard FT, it is quite straightforward to translate it into an equivalent BN with discrete nodes, and any analysis that can be performed using the FT formulation can also be performed by means of standard BN inference [14, 15]. However, BNs offer more than this. We will discuss these issues by way of a real-world example: The dependability analysis of the HSDE gas turbine controller [71].

6.1 System description

We start by giving a brief description of the HSDE controller. With respect to this analysis, the structure of the system can be summarized as follows (see Figure 8):

- The actual controller is composed by two subsystems:
  - The main controller provides control and shutdown functions. In this analysis we restrict our attention to shutdown related to critical over-temperature or over-speed of the turbine;
  - The back-up unit provides protection function related to both of these critical situations. It has a CPU independent from the main controller and uses a separate power supply circuit (operating from the same supply inlet).

- The two subsystems share the two thermocouple signals and have the same speed probe.

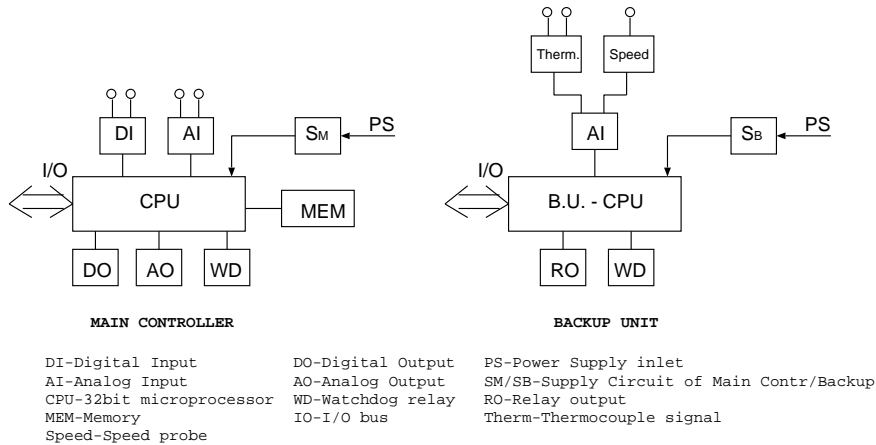- One watchdog is associated to each hardware circuit board.

Figure 8: Hardware structure of the HSDE controller.

The hardware structure of the main controller and the back-up unit are depicted in the left and right pane of Figure 8, respectively. The elementary blocks of the HSDE Controller are assumed to have constant failure rates as reported in Table 1.

In this analysis we are interested in the event that the controller fails to provide the correct control function, and that the backup unit fails to provide its protective function at the same time. This was also used as the top event of the FT representation of the problem given in [71].

13

IObus:	λ_{IO}	=	2.0	Therm.:	λ_{Th}	=	2.0
Speed:	λ_{Sp}	=	2.0	Memory:	λ_{M}	=	50
DO:	λ_{DO}	=	250	AO:	λ_{AO}	=	250
RO:	λ_{RO}	=	250	DI:	λ_{DI}	=	300
AI:	λ_{AI}	=	300	PS:	λ_{PS}	=	300
S_{MC}:	λ_{Sm}	=	300	S_{BU}:	λ_{Sb}	=	300
CPU:	λ_{CPU}	=	500	WD:	λ_{WD}	=	250

Table 1: Failure rates (per 10^9 hours) for the elementary blocks of the HSDE controller.

6.2 The Bayesian network model

According to the translation algorithm presented in [15], the Bayesian network derived form the FT is reported in Figure 9 (see [71] for details). To make Figure 9 more self-consistent, we have labelled the internal nodes with the corresponding boolean function (either "AND" or "OR").

The BN is multiply connected, because both the main controller and the backup unit rely on the power system (PS) and the sensors (the sub-system denoted Trans Sig in Figure 9). All variables are binary, with values that represents the state of the associated component or subsystem (either working or failed). At this stage in the modelling, the only probabilistic nodes of the BN are the root nodes (corresponding to the basic events in the FT). These must be assigned CDFs, and based on the failure rates in Table 1, the probability for a component failure at a given mission time t is calculated as follows: Consider a generic component C with failure rate λ_C. At a specific mission time t we have that $P(C = \mathsf{F}) = 1 - e^{-\lambda_C t}$; we use "F" as a shorthand for "Failed"; we will also use "W" for "Working".

6.3 Typical FTA results

We can now evaluate the system's unreliability by computing the probability of TE in our BN. The calculated unreliability is plotted with a solid line in fig. 11. Next, we consider how to analyze the criticality of the system components with respect to system failure. To this end, we should consider system failure as evidence provided to the BN. There are two main computations that can be performed:

1. The posterior probability of each single component being failed given that the system has failed;

2. The most probable configuration over the set of components

The first analysis allows one to obtain information about the Vesely/Fussell criticality of each single component. It is calculated by entering evidence that top event has occurred, and the inference algorithm (Section 5) is employed to compute the probability of each component being failed. Table 2 reports these numbers computed at mission time $t = 5 \cdot 10^5$ hours.[9] We notice that the two watchdogs WD_M and WD_B have a criticality 1.0. This is to be expected, as they both have to fail for the system to fail. We also remark that the importance of the CPU is 31% for the main controller and 37% for the backup unit. The different roles the two components play in the overall system explains why the importance values are different, even though their failure rates are identical.

The second kind of analysis is more sophisticated, and approaches the criticality problem over a set of components. It closely resembles the computation of minimal cutsets in FTs. One can think of the calculations taking place as finding the posterior joint probability of all the components, given the fact that the system has failed (although more clever algorithms are employed in practice [74]). Notice that, while in general the first and second kind of analysis are performed using different algorithms, the algorithm used in the second point

---

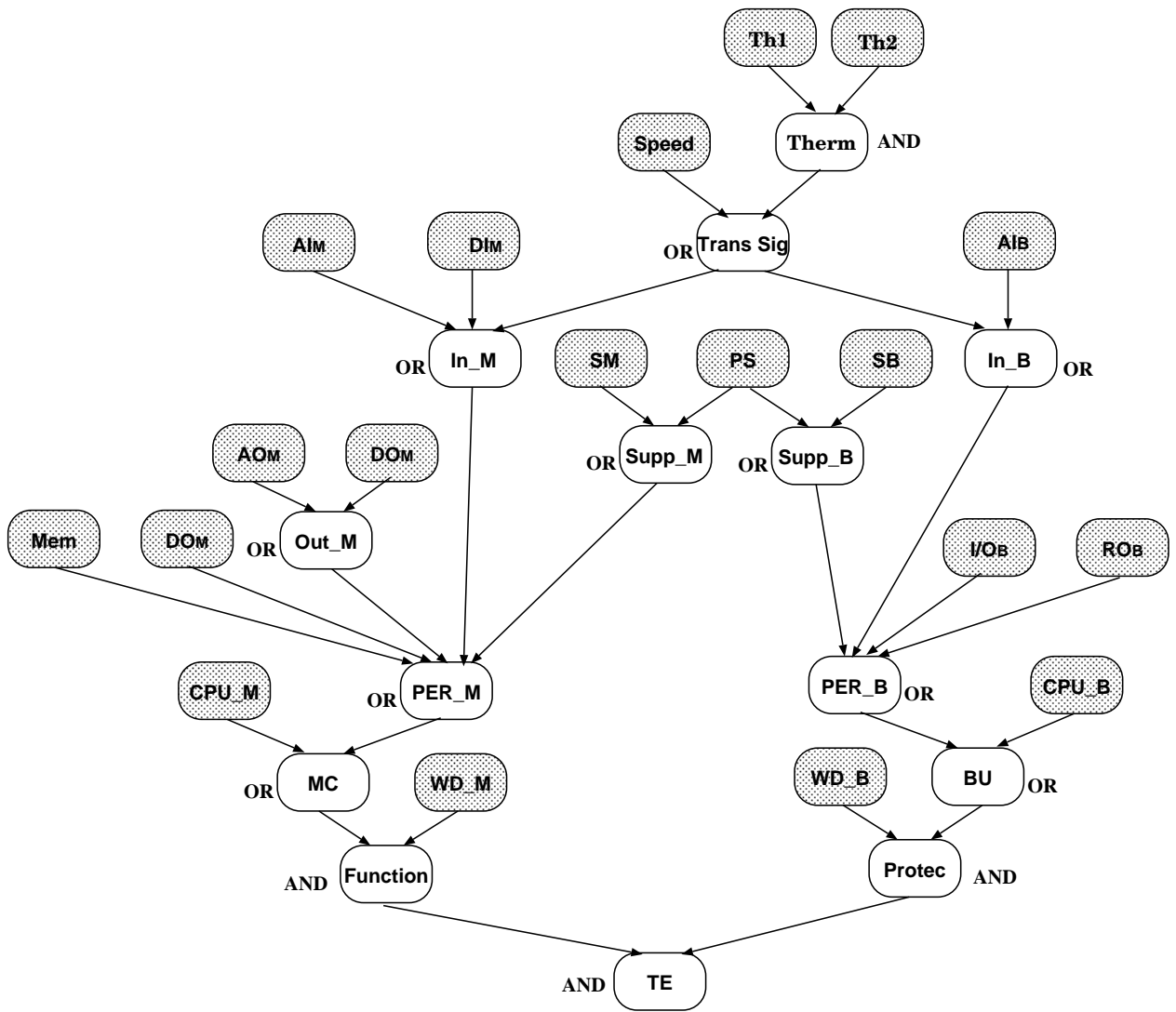[9]These values were calculated by the SPI tool [72] and the results were verified using JavaBayes [73].

Figure 9: A Bayesian network model describing this system (translated from a FT, see [71]).

Component	Posterior	Component	Posterior
WD_B	1.0	WD_M	1.0
CPU_B	0.37	PS	0.35
CPU_M	0.31	AIB	0.23
SB	0.23	ROB	0.20
AIM	0.19	SM	0.19
DIM	0.19	AOM	0.16
DOM	0.16	Mem	0.034
Speed	0.0025	I/OB	0.0017
I/OM	0.0014	Th1	0.0010
Th2	0.0010		

Table 2: Vesely/Fussell's importance measure.

above may compute any arbitrary joint CDF over a set of variables. This means that one can concentrate on any particular part of the system and to analyze the criticality of that particular subsystem.

Table 3 reports the top 9 configurations of the components having the highest posterior probabilities. Table 3 should be read such that the mentioned components are faulty, whereas all the others are operating (e.g., the most probable configuration of the components given system failure is that the power supply inlet and the two watchdogs are faulty whereas all other components work properly; the probability for this situation is 6.6%).

Faulty Components	Posterior
PS WDb WDm	0.0660
CPUb CPUm WDb WDm	0.0329
DIm CPUb WDb WDm	0.0187
SM CPUb WDb WDm	0.0187
AIm CPUb WDb WDm	0.0187
SB CPUm WDb WDm	0.0187
PS CPUm WDb WDm	0.0187
PS CPUb WDb WDm	0.0187
AIb CPUm WDb WDm	0.0187

Table 3: Most Probable Posterior Configurations

6.4 Coverage factor in BN model (probabilistic AND)

An important modelling improvement in redundant systems is to consider *coverage* factors. The coverage factor is defined as the probability that a single failure in a redundant system entails a complete system failure. This accounts for the fact that the recovery mechanism can be inaccurate, and that the redundancy therefore becomes inoperative even when only one component has failed. Coverage factors may be modelled in FTs [75]; however, it finds an even more natural application in BNs. Here, we resort to defining probabilistic gates (as described, e.g., in the discussion leading to Figure 4).

Figure 10 reports an excerpt of Figure 9 related to the gate labelled Function. It shows a probabilistic AND-gate and the CDF, which models the situation. Events are binary, and its possible values are denoted by W (working) or F (failed). When using the deterministic AND-gate, Function is failed (with probability 1) when both inputs are down, and working (with probability 1) otherwise. This should compared with the

probabilistic case, where Function may be down with probability $1 - c$ (c is the coverage factor) even when only one input is down. Notice that the coverage factor model modifies only the quantitative part of the model (as in [76]).
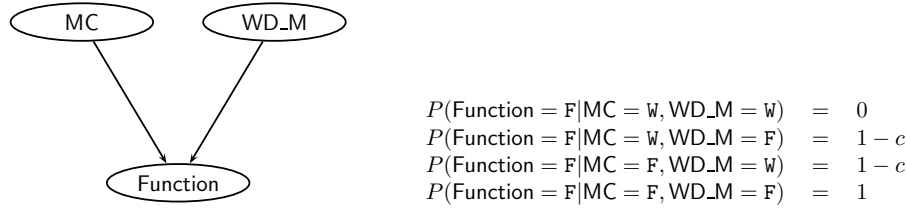

$$
\begin{aligned}
P(\mathsf{Function} = \mathtt{F} | \mathsf{MC} = \mathtt{W}, \mathsf{WD\_M} = \mathtt{W}) &= 0 \\
P(\mathsf{Function} = \mathtt{F} | \mathsf{MC} = \mathtt{W}, \mathsf{WD\_M} = \mathtt{F}) &= 1 - c \\
P(\mathsf{Function} = \mathtt{F} | \mathsf{MC} = \mathtt{F}, \mathsf{WD\_M} = \mathtt{W}) &= 1 - c \\
P(\mathsf{Function} = \mathtt{F} | \mathsf{MC} = \mathtt{F}, \mathsf{WD\_M} = \mathtt{F}) &= 1
\end{aligned}
$$

Figure 10: An AND gate with coverage and the corresponding CPT.

To show the effect of the coverage factor on the availability of the system, we introduced a coverage factor c to the gates Function, Protec and TE. We calculated the unavailability of the system for $c = 0.9$, $c = 0.95$, and $c = 0.99$, and give the results in fig 11. For the sake of comparison, also the deterministic case (coverage factor $c = 1$) is reported.

6.5   Multi-state nodes

Next, we will look at events whose behavior is best described by multi-state variables. Let us consider a case where the power supply (PS) can be found in three different conditions (states): *working*, *failed*, and *abnormal*. When PS is working or failed, the behavior of the overall system is the same as for the model in Figure 9. However, we assume that when PS is abnormal, it induces an anomalous behavior in the supply equipment of the main controller (SM) and the back-up unit (SB) as well. Figure 12 shows the BN, which models this situation (again, only the relevant part of Figure 9 is considered). Edges introduced to connect PS with SM and SB indicate an influence of PS on SM and SB. This influence is quantified in the CDFs (not shown). The abnormal status of the power supply now has both a direct effect as well as an indirect effect on the system dependability; the latter originates from the power supply's negative influence on other components in the system.

Note that we have introduced this extra aspect to our model without problems; in fact only a small part of the model is changed, and this is seamlessly integrated into the overall BN. The calculation algorithms do not have to be changed. We could not have done this as easily in the FT framework.

6.6   Parameter uncertainty and sensitivity analysis

Our next example takes a closer look at parameter uncertainty. In classical FTA we can treat the uncertainty in the model parameters (e.g., components' failure rates) by repeatedly computing the system's unreliability using different parameter values and thereby obtain upper and lower bounds for the unreliability. For example, assume that all failure rates in Table 1 are subject to an uncertainty of $\pm 10\%$. The results are comparatively reported in Figure 13.[10] The distance between the two curves lambda_max (+10%) and lambda_min (-10%) represents the uncertainty in the calculated unreliability due to parameter uncertainty.

In the Bayesian setting, parameters are considered as random variables, and their uncertainty captured by probability distributions. Accordingly, the system's unreliability is calculated as a weighted average over the parameters, and do not reflect a single deterministic value.

To illustrate this point, we can carry out the following experiment, assuming again PS as the exemplificative node. The parameter we require is in this case the failure rate of the power supply, λ_{PS}. In the Bayesian setting, we assume that Λ_{PS} is a random variable, for instance by using the Gamma-distribution:

---

[10]The system's unreliability was computed by the SHARPE tool [77].
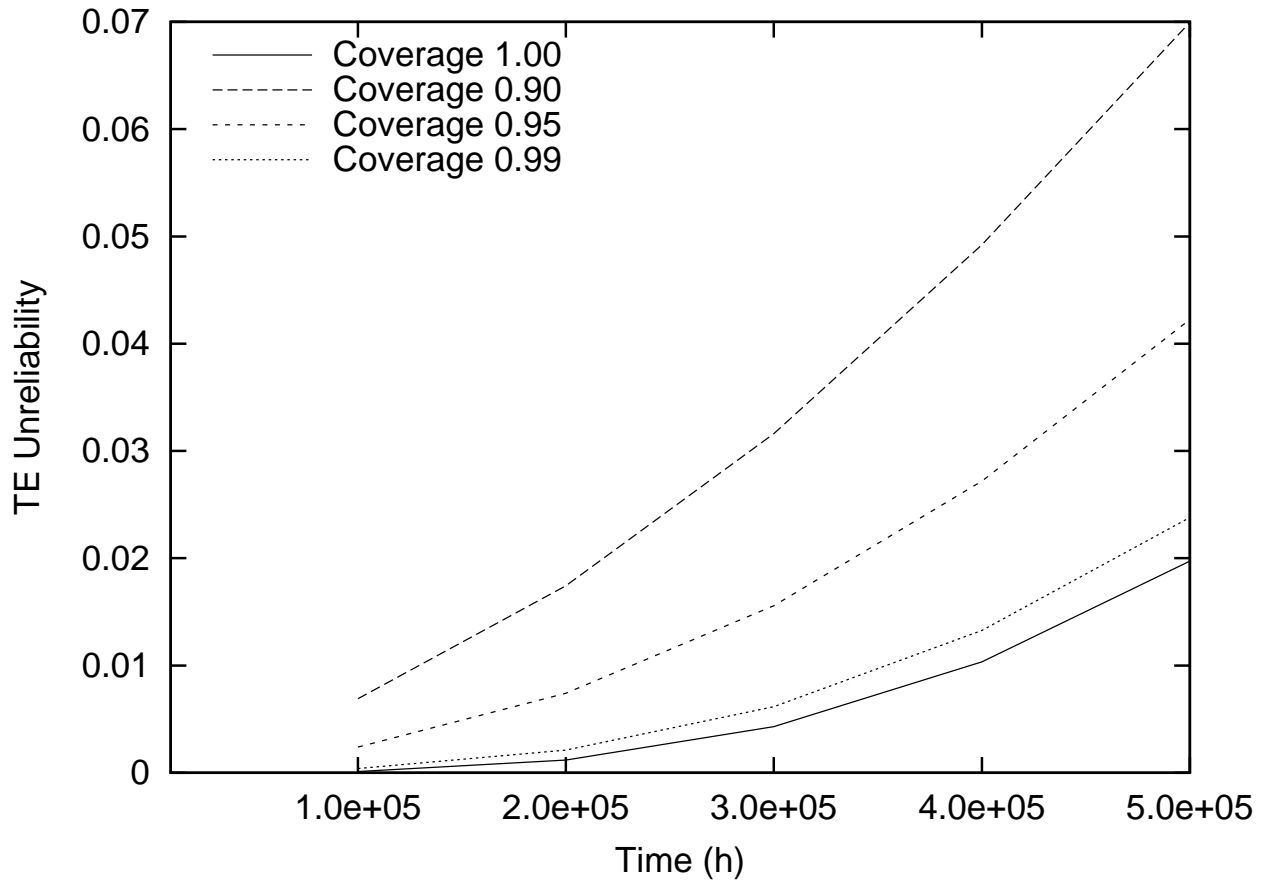
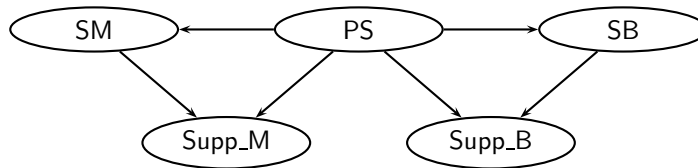Figure 11: System unreliability for different coverage factors.

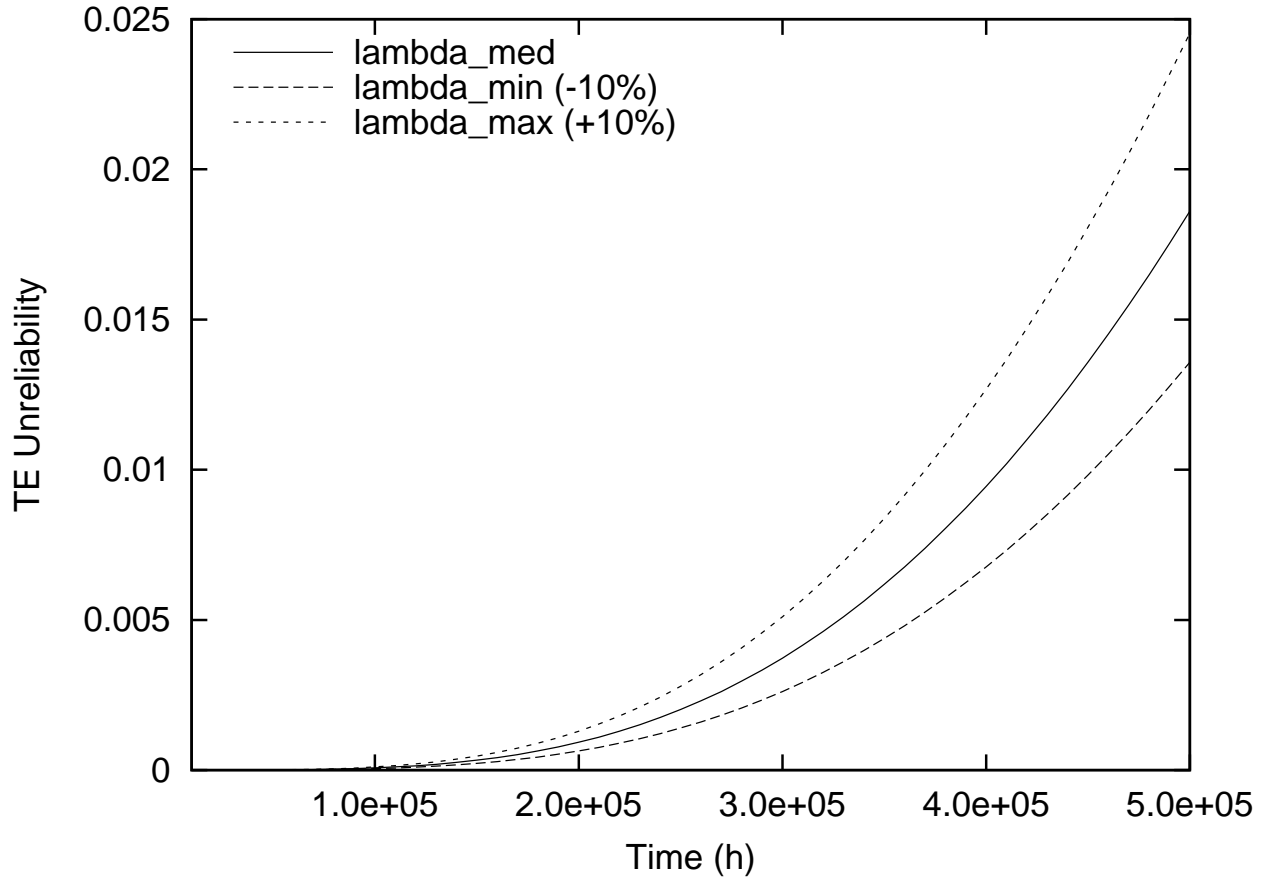Figure 12: Portion of the BN showing the influence of PS.

Figure 13: Sensitivity analysis of the TE Unreliability.

$\Lambda_{PS} \sim \Gamma(\alpha, \beta)$. It seems reasonable to choose the hyper-parameters α and β such that Λ_{PS} gets expectation corresponding to the value given in Table 1: $E[\Lambda_{PS}] = \alpha \cdot \beta = 3.0 \ 10^{-7}$. In the BN structure, PS gets Λ_{PS} as a new parent, see Figure 14 where only the relevant part of the model is included. We must also define its CDF, and do that by insisting that

$$P(\mathsf{PS} = \mathtt{F}|\Lambda_{PS} = \lambda_{PS}) = 1 - \exp(-\lambda_{PS} \cdot t).$$

We can now take the parameter uncertainty into account by calculating the system's unavailability at a given mission time t from our (extended) BN. We note, however, that the system's unavailability is essentially linear in Λ_{PS} (in particular we have that $1 - \exp(-\Lambda_{PS} \cdot t) \approx \Lambda_{PS} \cdot t$ for small values of $\Lambda_{PS} \cdot t$). The effect of the parameter uncertainty is therefore negligible in our example, and the numerical results are omitted. A more interesting situation occurs when the parameter uncertainty induce dependence between the different components, and this is examined in Section 6.7.

6.7 Components sharing a common environment

When modelling complex coherent systems, it is quite common to assume that components can be considered independent, even when they are operating in a common environment. Several researchers have been trying to overcome this defect by explicitly modelling the correlation between components' life-lengths that the
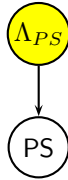
Figure 14: Λ_{PS} is the failure rate of PS. It is modelled as a random variable.

shared environment introduce (see, e.g., [78, 79, 80]). In this subsection we will elaborate on a solution to this problem described by Lindley and Singpurwalla [79].

Consider a parallel system of two components. The components have life-lengths T_1 and T_2 respectively, and the system's life-length is given as $R = \max(T_1, T_2)$. In [79] it is assumed that when the components are operating in a controlled laboratory environment, their life-lengths T_i are exponentially distributed with known parameters λ_i ($i = 1, 2$). Next, the two components are exposed to some common environment, and this introduces a correlation between T_1 and T_2: A rough environment will lead to reduced life-lengths for both components, whereas a gentle environment would imply that the expected life-lengths of both components were increased. We use a random variable E to model the effect of the common environment. It is assumed that the effect of the environment is proportional to the failure rate, that is, $T_i|\{E = \xi\}$ is exponentially distributed with parameter $\xi\lambda_i$. A correlation between T_1 and T_2 is introduced if E is not observed, whereas $T_1 \perp\!\!\!\perp T_2|E$ (compare Figure 15 (a) to Figure 2 (b)). Lindley and Singpurwalla [79] continue their modelling by assuming E to follow a Gamma distribution with known parameters, and (amongst other things) derive the marginal distribution of R when E is unobserved.
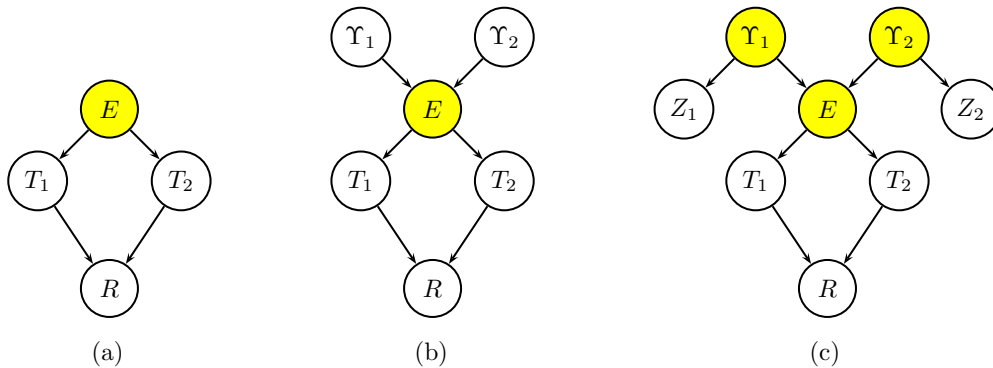

Figure 15: (a) Two components in a parallel system have life-lengths T_1 and T_2 respectively, giving the system a life-length of $R = \max(T_1, T_2)$. The random variables T_1 and T_2 are traditionally assumed independent, but when exposed to a common environment, E, a dependence is introduced. (b) Covariates Υ_1 and Υ_2 are measured to infer properties of the environment. (c) The model is enhanced by introducing measurement error on the covariates. Z_1 and Z_2 denote the measured values of Υ_1 and Υ_2 respectively. See text for further details.

We can extend this example by assuming that we can characterize the environmental effect E by regression. That is, we presume the existence of a number of covariates $\Upsilon_1, \ldots, \Upsilon_\ell$ such that E follows a distribution with parameters defined as functions of these covariates. The corresponding model with $\ell = 2$ is shown in Figure 15 (b). Finally, in Figure 15 (c) we have included measurement uncertainty for the covariates. This model applies if we are not able to measure the covariates (Υ_i) themselves; only the noisy measurements (Z_i)

are observed.

To exemplify the use of this model, we quantify the CDFs as follows: The covariates Υ must be assumed to be realizations of some distribution for this to work. Here we have no a priori information, and they are therefore allocated vague prior distributions (Gaussian distributions with expectation 0 and variance 10^6 were used). Measurements are assumed unbiased with variance 0.1, i.e., $Z_i|\{\Upsilon_i = v_i\} \sim N(v_i, 0.1)$. We follow [79] and let the environment be determined by a Gamma distribution, $E \sim \Gamma(r, \mu)$. We define the rate by exponential regression, $\mu = \exp(-\boldsymbol{\beta}^{\mathrm{T}} v)$, and assume known shape $r = 2$. $\beta_1 = 0.01$ and $\beta_2 = 0.04$ were chosen rather arbitrarily in this example. Finally, $T_i|\{E = \xi\}$ follows the exponential distribution with parameter $\xi\lambda_i$; $\lambda_1 = 3 \cdot 10^{-3}$ and $\lambda_2 = 2 \cdot 10^{-3}$ respectively. We used BUGS [70] to calculate $P(R > 1000 \,|\, Z_1 = 1, Z_2 = 2) = 0.29$ and $\mathrm{corr}(T_1, T_2 \,|\, Z_1 = 1, Z_2 = 2) = 0.90$. If we fail to model the correlation between the two life-lengths, we will calculate $P(R > 1000 \,|\, Z_1 = 1, Z_2 = 2) = 0.41$.

This example highlights the importance of being able to make mathematical models that we actually believe in. BNs can be a framework to make such models, also in the context of reliability analysis.

7 Conclusions

In this paper we have considered the applicability of Bayesian networks for reliability analysis. BNs constitute a modelling framework which is particularly easy to use for interaction with domain experts, and this makes it a useful tool in practice. This is evident, for instance, by the success BNs currently obtain in software reliability evaluation tasks. Furthermore, as BNs rest upon probability theory, many of the fundamental discussions obstructing other modelling frameworks are avoided. The sound mathematical formulation has been utilized to generate efficient learning methods. BNs are equipped with an efficient calculation scheme, which often makes them preferable to traditional tools like fault trees.

Many BN tools are available to the practitioners. Examples of commercial tools available online include Hugin (`http://www.hugin.com/`), BayesiaLab (`http://www.bayesia.com/`) and Netica (`http://www.norsys.com/`). BUGS (`http://www.mrc-bsu.cam.ac.uk/bugs/`) is a general-purpose modelling framework where inference is based on simulation. For a comprehensive list of software tools, see the one maintained by Kevin P. Murphy [81].

Acknowledgments

References

[1] R. E. Barlow, Using influence diagrams, in: C. A. Clarotti, D. V. Lindley (Eds.), Accelerated life testing and experts' opinions in reliability, 1988, pp. 145–157.

[2] R. G. Almond, An extended example for testing graphical belief, Tech. Rep. 6, Statistical Sciences Inc. (1992).

[3] N. E. Fenton, B. Littlewood, M. Neil, L. Strigini, A. Sutcliffe, D. Wright, Assessing dependability of safety critical systems using diverse evidence, IEE Proceedings Software Engineering 145 (1) (1998) 35–39.

[4] B. A. Gran, The use of Bayesian belief networks for combining disparate sources of information in the safety assessment of software based systems, Ph.D. thesis, Department of Mathematical Sciences, Norwegian University of Science and Technology, doktor Ingeniør avhandling 2002:35 (2002).

[5] F. V. Jensen, U. Kjærulff, B. Kristiansen, H. Langseth, C. Skaanning, J. Vomlel, M. Vomlelová, The SACSO methodology for troubleshooting complex systems, Artificial Intelligence for Engineering, Design, Analysis and Manufacturing 15 (5) (2001) 321–333.

[6] R. Barco, R. Guerrero, G. Hylander, L. M. Nielsen, M. Partanen, S. Patel, Automated troubleshooting of mobile networks using Bayesian networks, in: Proceedings of the IASTED International Conference on Communications Systems and Networks, ACTA Press, Anaheim, CA, 2002, pp. 105–110.

[7] G. Arroyo-Figueroa, L. E. Sucar, Temporal Bayesian network of events for fault diagnosis and prediction in thermal power plants, Presented at the First Bayesian Applications Modeling Workshop [23], `http://www.intel.com/research/events/UAI03_workshop/` (2003).

[8] O. Kipersztok, G. Provan, A framework for diagnostic inference of commercial aircraft systems, Presented at the First Bayesian Applications Modeling Workshop [23], `http://www.intel.com/research/events/UAI03_workshop/` (2003).

[9] H. Langseth, F. V. Jensen, Decision theoretic troubleshooting in coherent systems, Reliability Engineering and System Safety 80 (1) (2003) 49–61.

[10] A. L. Madsen, U. B. Kjærulff, J. Kalwa, M. Perrier, M. Ángel Sotelo, Applications of probabilistic graphical models to diagnosis and control of autonomous vehicles, Presented at the Second Bayesian Applications Modeling Workshop [24], `http://www.intel.com/research/events/bayesian2004/` (2004).

[11] H. Langseth, B. H. Lindqvist, A maintenance model for components exposed to several failure modes and imperfect repair, in: K. Doksum, B. H. Lindqvist (Eds.), Mathematical and Statistical Methods in Reliability, Quality, Reliability and Engineering Statistics, World Scientific, Singapore, 2003, Ch. 27, pp. 415–430.

[12] G. Weidl, A. L. Madsen, E. Dahlquist, Object oriented Bayesian network for industrial process operation, Presented at the First Bayesian Applications Modeling Workshop [23], `http://www.intel.com/research/events/UAI03_workshop/` (2003).

[13] J. G. Torres-Toledano, L. E. Sucar, Bayesian networks for reliability analysis of complex systems, in: Proceedings of the 6th Ibero-American Conference on AI (IBERAMIA 98), no. 1484 in Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin, Germany, 1998, pp. 195–206.

[14] L. Portinale, A. Bobbio, Bayesian networks for dependability analysis: an application to digital control reliability, in: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Fransisco, CA., 1999, pp. 551–558.

[15] A. Bobbio, L. Portinale, M. Minichino, E. Ciancamerla, Improving the analysis of dependable systems by mapping fault trees into Bayesian networks, Reliability Engineering and System Safety 71 (3) (2001) 249–260.

[16] A. P. Tchangani, Reliability analysis using Bayesian networks, Studies in Informatics and Control 10 (3) (2001) 181–188.

[17] H. Langseth, Bayesian networks with applications in reliability analysis, Ph.D. thesis, Department of Mathematical Sciences, Norwegian University of Science and Technology, Doktor Ingeniør avhandling 2002:121. Available at `http://www.math.ntnu.no/~helgel/thesis/` (2002).

[18] M. Ingleby, M. West, Causal influence coefficients: A localised maximum entropy approach to Bayesian inference, in: K. Doksum, B. H. Lindqvist (Eds.), Mathematical and Statistical Methods in Reliability, Quality, Reliability and Engineering Statistics, World Scientific Publishing Co., Singapore, 2003, Ch. 4, pp. 45–56.

[19] S. Montani, L. Portinale, A. Bobbio, Dynamic Bayesian networks for modeling advanced fault tree features in dependability analysis, To appear in the Proceedings of the Sixteenth European Conference on Safety and Reliability, Tri City, Poland (2005).

[20] H. Boudali, J. B. Dugan, A temporal Bayesian network framework, Presented at the Fourth International Conference on Mathematical Methods in Reliability (MMR-04), Santa Fe, NM (2004).

[21] H. Boudali, J. B. Dugan, A new Bayesian network approach to solve dynamic fault trees, Presented at the 51st Annual Reliability & Maintainability Symposium, Alexandria, VA (2005).

[22] J. Solano-Soto, L. E. Sucar, A methodology for reliable system design, in: Proceedings of the 4th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, Vol. 2070 of Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin, Germany, 2001, pp. 734–745.

[23] J. M. Agosta, O. Kipersztok, K. B. Laskey, K. W. Przytula, I. Rish (Eds.), Proceedings of the First Bayesian Applications Modeling Workshop, `http://www.intel.com/research/events/UAI03_workshop/`, 2003.

[24] O. Kipersztok, J. M. Agosta, R. Almond, B. Kappen, K. B. Laskey, K. W. Przytula, I. Rish (Eds.), Proceedings of the Second Bayesian Applications Modeling Workshop, `http://www.intel.com/research/events/bayesian2004/`, 2004.

[25] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann Publishers, San Mateo, CA., 1988.

[26] R. G. Cowell, A. P. Dawid, S. L. Lauritzen, D. J. Spiegelhalter, Probabilistic Networks and Expert Systems, Statistics for Engineering and Information Sciences, Springer-Verlag, New York, NY, 1999.

[27] F. V. Jensen, Bayesian Networks and Decision Graphs, Springer-Verlag, New York, NY, 2001.

[28] B. Abramson, J. Brown, W. Edwards, A. Murphy, R. L. Winkler, Hailfinder: A Bayesian system for forcasting severe weather, International Journal of Forecasting 12 (1996) 57–71.

[29] E. Charniak, Bayesian networks without tears, AI Magazine 12 (4) (1991) 50–63.

[30] J. Whittaker, Graphical models in applied multivariate statistics, John Wiley & Sons, Chichester, UK, 1990.

[31] J. Vomlel, Methods of probabilistic knowledge integration, Ph.D. thesis, Faculty of Electrical Engineering, Czech Technical University (1999).

[32] M. Druzdzel, L. van der Gaag, Building probabilistic networks: Where do the numbers come from? - a guide to the literature, Technical Report UU-CS-2000-20, Institute of Information & Computing Sciences, University of Utrecht, The Netherlands (2000).

[33] C. Skaanning, A knowledge acquisition tool for Bayesian-network troubleshooters, in: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Fransisco, CA., 2000, pp. 549–557.

[34] Z. Ghahramani, G. E. Hinton, The EM algorithm for mixtures of factor analyzers, Technical Report CRG-TR-96-1, Department of Computer Science, University of Toronto, Canada (1996).

[35] E. J. Henley, H. Kumamoto, Reliability Engineering and Risk Assessment, Prentice Hall, Englewood Cliffs, NJ, 1981.

[36] N. G. Leveson, Safeware: System Safety and Computers, Addison-Wesley, New York, NY, 1995.

[37] S. M. Mahoney, K. B. Laskey, Network engineering for complex belief networks, in: Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Fransisco, CA., 1996, pp. 389–396.

[38] D. Koller, A. Pfeffer, Object-oriented Bayesian networks, in: Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Fransisco, CA., 1997, pp. 302–313.

[39] O. Bangsø, P.-H. Wuillemin, Top-down construction and repetitive structures representation in Bayesian networks, in: Proceedings of the Thirteenth International Florida Artificial Intelligence Research Society Conference, The AAAI Press, Menlo Park, CA., 2000, pp. 282–286.

[40] U. Kjærulff, A computational scheme for reasoning in dynamic probabilistic networks, in: Proceedings of the Eighth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco, California, 1992, pp. 121–129.
URL ftp://ftp.cs.auc.dk/pub/reports/papers/kjaerulff92b.ps.%Z

[41] A. Bobbio, G. Franceschinis, R. Gaeta, L. Portinale, Parametric fault tree for the dependability analysis of redundant systems and its high-level Petri net semantics, IEEE Transactions on Software Engineering 29 (3) (2003) 270–287.

[42] A. Bobbio, S. Montani, L. Portinale, Parametric dependability analysis through probabilistic Horn abduction, in: Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Fransisco, CA., 2003, pp. 65–72.

[43] D. Heckerman, J. S. Breese, A new look at causal independence, in: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco, CA., 1994, pp. 286–292.

[44] C. Boutilier, N. Friedman, M. Goldszmidt, D. Koller, Context-specific independence in Bayesian networks, in: Proceedings of the Twelfth Conference on Uncertainty in Artificial Intelligence, San Fransisco, CA., 1996, pp. 115–123.

[45] N. L. Zhang, D. Poole, Exploiting causal independence in Bayesian networks inference, Journal of Artificial Intelligence Research 5 (1996) 301–328.

[46] J. Pearl, Fusion, propagation, and structuring in belief networks, Artificial Intelligence 29 (3) (1986) 241–288.

[47] F. J. Díez, Parameter adjustment in Bayes networks: The generalized Noisy-OR, in: Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Francisco, CA., 1993, pp. 99–105.

[48] A. Cano, S. Moral, Using classification trees to represent potentials in probabilistic causal networks, EURO XV/INFORMS XXXIV Joint International Meeting (Barcelona, Spain) (1997).

[49] D. J. Spiegelhalter, S. L. Lauritzen, Sequential updating of conditional probabilities on directed graphical structures, Networks 20 (1990) 579–605.

[50] A. P. Dempster, N. M. Laird, D. B. Rubin, Maximum likelihood from incomplete data via the EM algorithm, Journal of the Royal Statistical Society, Series B 39 (1977) 1–38.

[51] S. L. Lauritzen, The EM-algorithm for graphical association models with missing data, Computational Statistics and Data Analysis 19 (1995) 191–201.

[52] G. F. Cooper, E. Herskovits, A Bayesian method for the induction of probabilistic networks from data, Machine Learning 9 (1992) 309–347.

[53] D. Heckerman, D. Geiger, D. M. Chickering, Learning Bayesian networks: The combination of knowledge and statistical data, Machine Learning 20 (1995) 197–243, also available as Microsoft Research Technical Report MSR-TR-94-09.

[54] N. Friedman, The Bayesian structural EM algorithm, in: Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Fransisco, CA., 1998, pp. 129–138.

[55] S. Geman, D. Geman, Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images, IEEE Transactions on Pattern Analysis and Machine Intelligence 6 (1984) 721–741.

[56] S. G. Bøttcher, Learning Bayesian networks with mixed variables, in: Proceedings of the Eighth International Workshop in Artificial Intelligence and Statistics, Morgan Kaufmann Publishers, San Francisco, CA., 2001, pp. 149–156.

[57] N. Friedman, I. Nachman, Gaussian process networks, in: Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers, San Fransisco, CA., 2000, pp. 211–219.

[58] V. Romero, R. Rumí, A. Salmerón, Structural learning of Bayesian networks with mixtures of truncated exponentials, in: Probabilistic Graphical Models 2004, 2004, pp. 177–184.

[59] J. Pearl, Causality – Models, Reasoning, and Inference, Cambridge University Press, Cambridge, UK, 2000.

[60] T. P. Speed, Complexity, calibration and causality in influence diagrams, in: R. M. Oliver, J. Q. Smith (Eds.), Influence Diagrams, Belief Nets and Decision Analysis, John Wiley & Sons, New York, NY, 1990, pp. 49–63.

[61] S. L. Lauritzen, Causal inference from graphical models, in: O. E. Barndorff-Nielsen, D. R. Cox, C. Klüppelberg (Eds.), Complex Stochastic Systems, Chapman & Hall, London, UK, 2001, pp. 63–107.

[62] A. P. Dawid, Influence diagrams for causal modelling and inference, International Statistical Review 70 (2) (2002) 161–189.

[63] J. Pearl, Causal diagrams for empirical research, Biometrika 82 (4) (1995) 669–710.

[64] H. Langseth, Analysis of survival times using Bayesian networks, in: Proceedings of the Ninth European Conference on Safety and Reliability, A. A. Balkema, 1998, pp. 647–654.

[65] F. V. Jensen, S. L. Lauritzen, K. G. Olesen, Bayesian updating in causal probabilistic networks by local computations, Computational Statistics Quarterly 4 (1990) 269–282.

[66] K. G. Olesen, U. Kjærulff, F. Jensen, F. V. Jensen, B. Falck, S. Andreassen, S. K. Andersen, A MUNIN network for the median nerve - A case study on loops, Applied Artificial Intelligence 3 (1989) 385–404.

[67] U. Kjærulff, Triangulation of graphs — algorithms giving small total state space, Research Report R-90-09, Department of Computer Science, Aalborg University, Denmark (1990).
URL ftp://ftp.cs.auc.dk/pub/reports/tech-reports/R90-09.ps.%Z

[68] S. Moral, R. Rumí, A. Salmerón, Mixtures of truncated exponentials in hybrid Bayesian networks, in: Sixth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty, Vol. 2143 of Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin, Germany, 2001, pp. 145–167.

[69] W. Gilks, S. Richardson, D. Spiegelhalter, Markov Chain Monte Carlo in practice, Interdisciplinary Statistics, Chapmann & Hall, London, UK, 1996.

[70] W. Gilks, A. Thomas, D. J. Spiegelhalter, A language and program for complex Bayesian modelling, The Statistician 43 (1994) 169–178.
URL `http://www.mrc-bsu.cam.ac.uk/bugs/`

[71] A. Bobbio, E. Ciancamerla, G. Franceschinis, R. Gaeta, M. Minichino, L. Portinale, Sequential application of heterogeneous models for the safety analysis of a control system: a case study, Reliability Engineering and System Safety 81 (2003) 269–280.

[72] B. D'Ambrosio, Local expression languages for probabilistic dependence, International Journal of Approximate Reasoning 11 (1) (1994) 55–81.

[73] F. G. Cozman, JavaBayes: Bayesian networks in Java, `http//www.cs.cmu.edu/~javabayes/` (2001).

[74] D. Nilsson, Finding the M most probable configurations in probabilistic expert systems, Statistics and Computing 8 (2) (1998) 159–173.

[75] S. Amari, J. B. Dugan, R. Misra, A separable method for incorporating imperfect fault-coverage into combinatorial models, IEEE Transactions on Reliability 48 (1999) 267–274.

[76] S. Doyle, J. B. Dugan, A. Patterson-Hine, A combinatorial approach to modeling imperfect coverage, IEEE Transactions on Reliability 44 (1995) 87–94.

[77] R. Sahner, K. Trivedi, A. Puliafito, Performance and Reliability Analysis of Computer Systems: An Example-based Approach Using the SHARPE Software Package, Kluwer Academic Publishers, Norwell, MA., 1996.

[78] M. Shaked, A concept of positive dependence for exchangeable random variables, The Annals of Statistics 5 (1977) 505–515.

[79] D. V. Lindley, N. D. Singpurwalla, Multivariate distributions for the lifelengths of components of a system sharing a common environment, Journal of Applied Probability 23 (1986) 418–431.

[80] P. A. Currit, N. D. Singpurwalla, On the reliability function for a system of components sharing a common environment, Journal of Applied Probability 26 (1988) 763–771.

[81] K. P. Murphy, Software packages for graphical models/Bayesian networks, `http://www.ai.mit.edu/~murphyk/Software/BNT/bnsoft.html` (2004).

[82] H. Langseth, Bayesian networks in reliability: Recent developments, invited talk at the Fourth International Conference on Mathematical Methods in Reliability *MMR-04*, Santa Fe, NM, June 21–24. (2004).

[83] L. Portinale, A. Bobbio, S. Montani, From AI to dependability: How Bayesian networks can be used for reliability modeling and analysis, invited talk at the Fourth International Conference on Mathematical Methods in Reliability *MMR-04*, Santa Fe, NM, June 21–24. (2004).