Dependability Modelling of Repairable Systems using Bayesian Networks

Martin Neil *, David Marquez**

** Department of Computer Science, Queen Mary University of London, London, E1 4NS, UK (Tel:; e-mail: martin@ dcs.qmul.ac.uk) and Agena Ltd, 32 Hatton Garden, London, EC1N 8DL, UK*
*** Department of Computer Science, Queen Mary University of London, London, E1 4NS, UK, (e-mail: marquezd@dcs.qmul.ac.uk)*

Abstract: We present a Hybrid Bayesian Network (HBN) framework to model the availability of renewable systems. We use an approximate inference algorithm for HBNs that involves dynamically discretizing the domain of all continuous variables and use this to obtain accurate approximations for the renewal or repair time distributions for a system. We show how we can use HBNs to model corrective repair time, logistics delay times and scheduled maintenance time distributions and combine these with time to failure distributions to derive system availability. Example models are presented and are accompanied by detailed descriptions of how repair (renewal) distributions might be modelled using HBNs.

Keywords: Bayesian Networks; Bayesian Software; Systems Dependability; Dynamic Discretization, Renewable Systems, Systems Availability.

1. INTRODUCTION

We have used Bayesian Networks (BNs) in a range of real-world applications of system dependability assessment; see for example Neil et al. (2001, 2003, 2008). In such applications it is inevitable that there will be a mixture of discrete and continuous nodes (the resulting BNs are called hybrid, (HBNs)). The traditional approach to handling (non-Gaussian) continuous nodes is static: you have to discretize the continuous domains using some pre-defined range and intervals. However, this approach is unacceptable for critical type systems where there is a demand for reasonable accuracy. To overcome this problem, we have developed a new and powerful approximate algorithm for performing inference in HBNs. We use a process of dynamic discretization of the domain of all continuous variables in the HBN. The approach is influenced by the work of Kozlov and Koller (1997) using entropy error as the basis for approximation. We differ from their approach by integrating an iterative approximation scheme within existing BN software architectures, such as in Junction Tree (JT) propagation, Jensen at al. (1990) . Thus, rather than support separate data structures and a new propagation algorithm we use the data structures commonly used in JT algorithms.

The power and flexibility of the approach is demonstrated here by applying it to estimate the availability of repairable systems represented by a series of models each designed to model distinct stages in the renewal process: logistics delays, repairs and scheduled maintenance. Traditionally, modelling these events has relied on Monte Carlo simulation, involving many repeated simulation runs. In contrast to the simulation approach, we show how our HBN algorithms can be used to represent repair and support processes and the durations involved, under any assumptions for renewal time distributions (lognormal, exponential, etc.) The modelling has been made possible using the commercial general-purpose Bayesian Network software tool AgenaRisk, Agena Ltd (2009).

2. BACKGROUND

BNs have been widely used to represent full probability models in a compact and intuitive way. In the BN framework the independence structure in a joint distribution is characterised by a directed acyclic graph, with nodes representing random variables (which can be discrete or continuous, and may or may not be observable), and directed arcs representing causal or influential relationship between variables, Pearl (1993). The conditional independence assertions about the variables, represented by the lack of arcs, reduce significantly the complexity of inference and allow the underlying joint probability distribution to be decomposed as a product of local conditional probability distributions (CPD) associated with each node and its respective parents. If the variables are discrete, the CPDs can be represented as Node Probability Tables (NPTs), which list the probability that the child node takes on each of its different values for each combination of values of its parents. Since a BN encodes all relevant qualitative and quantitative information contained in a full probability model, it is an excellent tool for many types of probabilistic inference where we need to compute the posterior probability distribution of some variables of interest (unknown parameters and unobserved data) conditioned on some other variables that have been observed.

A range of robust and efficient propagation algorithms has been developed for exact inference on Bayesian networks

with discrete variables, Shenoy et al. (1990), Jensen at al. (1990), Lauritzen et al. (1993). The common feature of these algorithms is that the exact computation of posterior marginal distributions is performed through a series of local computations over a secondary structure, a tree of clusters, enabling calculation of the marginal without computing the joint distribution. See also Huang et al. (1996).

The present generation of BN software tools attempt to model continuous nodes by numerical approximations using static discretization as implemented in a number of software tools (e.g., Hugin, Netica). Although discretization allows approximate inference in a hybrid BN without limitations on relationships among continuous and discrete variables, current software implementations require users to define a uniform discretization of the states of any numeric node (whether it is continuous or discrete) as a sequence of pre-defined intervals, which remain static throughout all subsequent stages of Bayesian inference regardless of any new conditioning evidence. The more intervals you define, the more accuracy you can achieve, but at a heavy cost of computational complexity. This is made worse by the fact that you do not necessarily know in advance where the posterior marginal distribution will lie on the continuum for all nodes and which ranges require the finer intervals. It follows that where a model contains numerical nodes having a potentially large range, results are necessarily only crude approximations.

Alternatives to discretization have been suggested by Moral et al. (2001) and Cobb et al. (2003), who describe potential approximations using mixtures of truncated exponential (MTE) distributions, Koller et al. (1999), who combine MTE approximations with direct sampling (Monte Carlo) methods, and Murphy (1999) who uses variational methods. There have also been some attempts for approximate inference on hybrid BNs using Markov Chain Monte Carlo (MCMC) approaches, Shacter (1989). However, constructing dependent samples that mixed well (i.e., that move rapidly throughout the support of the target posterior distribution) remains a complex task.

3. DYNAMIC DISCRETIZATION

Let X be a continuous random node in the BN. The range of X is denoted by Ω_X, and the probability density function (PDF) of X, with support Ω_X, is denoted by f_X. The idea of discretization is to approximate f_X as follows:

1. Partition Ω_X into a set of intervals $\Psi_X = \{w_j\}$, and
2. Define a locally constant function $\tilde{f}_X$ on the partitioning intervals.

Discretization operates in much the same way when X takes integer values but in this paper we will focus on the case where X is continuous. As Kozlov and Koller (1997), we use an upper bound of the Kullback-Leibler (KL) metric between two density functions f and g:

$$D(f \parallel g) = \int_S f(x) \log \frac{f(x)}{g(x)} dx$$

as an estimate of the relative entropy error induced by the discretized function. Under the KL metric, the optimal value for the discretized function $\tilde{f}$ is given by the mean of the function f in each of the intervals of the discretized domain. The main task reduces then to finding an optimal discretization set $\Psi_X = \{\omega_j\}$.

Our approach to dynamic discretization searches Ω_X for the most accurate specification of the high-density regions given the model and the evidence, calculating a sequence of discretization intervals in Ω_X iteratively. At each stage in the iterative process, a candidate discretization, Ψ_X, is tested to determine whether the relative entropy error of the resulting discretized probability density $\tilde{f}_X$ is below a given threshold, defined according to the trade off between the acceptable degree of precision and computation time.

By dynamically discretizing the model we achieve more accuracy in the regions that matter and incur less storage space over static discretizations. Moreover, we can adjust the discretization any time in response to new evidence to achieve greater accuracy. A detailed description of the dynamic discretization algorithm is given in Neil et al. (2007). In outline, dynamic discretization follows these steps:

1. Convert the BN to a junction tree (JT) and choose an initial discretization for all continuous variables.
2. Calculate the NPT of each node given the current discretization
3. Enter evidence and perform global propagation on the junction tree, using standard JT algorithms.
4. Query the BN to get posterior marginal distributions for each node, compute the approximate relative entropy error, and check if it satisfies the convergence criteria.
5. If not, create a new discretization for the node by splitting those intervals with highest entropy error.
6. Repeat the process by recalculating the NPTs and propagating the BN, and then querying to get the marginal distributions and then split intervals with highest entropy error.
7. Continue to iterate until the model converges to an acceptable level of accuracy.

Nodes with continuous functions can be declared as a piecewise set of constant values and/or functions and these are supported during dynamic discretization. The relevant subdomains are inserted for each constant/function at the start of the discretization process and then the algorithm initiated. These subdomains are maintained, unless a constant or function is assigned zero probability during evidence propagation, in which case the relevant subdomain is removed from the node's domain. Only those subdomains with non zero probability are then represented in the posterior marginal distribution for a node at the end of the process. At its most basic it means the algorithm can perform arithmetic calculations: although seemingly trivial no other BN algorithm can perform this task.

4. MODELLING RENEWAL DISTRIBUTIONS USING HBNs

In this section we use the dynamic discretization approach to estimate the distribution of the renewal times of a continuous use repairable system. Bayesian modelling offers a suitable framework for assessing the repair durations of such systems, allowing us to integrate information from multiple sources at different levels of granularity, as well as expert opinion. Here we are interested in modelling repair times, logistics delays and scheduled maintenance. The model considered here is based on the following assumptions.

4.1 Modelling Repair Times using Mixtures of Lognormal Repair Time Distributions

The Inherent Availability of a system, Elsayed (1996), is partly determined by the corrective repair time of the system. For complex systems there may be a variety of repair locations or “lines”, each of which may specialise in particular classes of repair where the repairs take a variable time. Examples of repair lines might be: users, first line support, second line support and manufacturer support (these are often numbered as 0, 1, 2, 3, 4). Each of these lines of repair can be characterised by a repair time distribution; in this case we will assume a lognormal distribution (often with parameters defined by user supplied median, m, and 95$^{\text{th}}$ percentile repair times, α). The lognormal parameters can also be derived from historical data or the parameters can be learned using an HBN model as might be done when learning repair rates as described in Marquez et al. (2008).

If we represent the repair lines as the states of a discrete labelled variable, L, with $L = \{0,1,2,3,4\}$, then formally the marginal distribution of the repair time, τ, is given by the mixture of the repair time distributions for each repair line, τ:

$$f(\tau) = \sum_{l=0}^{4} f(\tau \mid L = l)P(L = l)$$

with $f(\tau \mid L = l) = LogNormal(m_l, \alpha_l)$. Table 1 shows an example with five lines of repair and median and 95th percentile estimates for each. The HBN used to model this problem is shown in Figure 1.

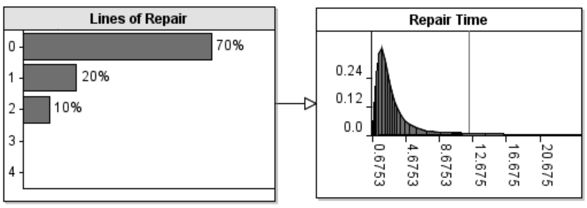

Fig. 1. HBN modelling a five repair level system with a mixture of lognormal repair times. Marginal distributions superimposed on the graph.

The mean repair time of the system as calculated from the HBN is 3.74 hours and the 95th percentile is 12.19 hours.

Table 1 Lognormal repair times for each repair line

L (lines of repair)	**0**	**1**	**2**	**3**	**4**
$P(L)$	0.7	0.2	0.1	0	0
m_l (hours)	2	4	12	-	-
α_l (hours)	4	8	24	-	-

4.2 Modelling Logistics Delay Times using Mixtures of Normal Logistics Delay Time Distributions

Operational availability of a system is dependent on the logistics delay time as well as the repair time. Here we wish to model the logistics delay time for a system which has a logistics chain represented by a number of discrete delay events such as administration, resourcing, sparing and recovery, and which might themselves have different duration distributions depending on the line of repair involved.

Formally we represent the total logistics delay time, T, as the sum of the delay times, T_i, for the delay events $i = 1, 2, ..., n$: $T = \sum_{i=1}^{n} T_i$. The duration of each delay event will depend on the line of repair and the conditions under which they are performed. For example recovery may only be required in a small number of cases, and will only be enacted as repair line 1. Likewise the delay time for spares will be a function of whether they are actually required or whether they are at hand.

The conditioning event set we will represent as a series of Boolean switch variables, S_j, $j = 1, 2, ..., m$, that can be used, in conjunction with the repair line variable, L, to index each delay time node and assign a delay time distribution to that node, such that:

$$f(T_i \mid S_j = Yes, L = l) = TNormal(\mu_{jl}, \sigma_{jl}^2, 0, \infty)$$
$$f(T_i \mid S_j = No, L = l) = 0$$

Here again, the parameters of the right truncated Gaussian normal distribution can be derived from historical data or expert knowledge. The switch variables, S_j, are set with probabilities to indicate the proportion of repair events that involve each delay stage. So if a discrete stage does not take place the delay time is zero and a Normal distribution otherwise (and this can be implemented using the support offered for piecewise functions and constants in the dynamic discretization algorithm).

The marginal distributions for the logistics delay times are therefore:

$$f(T_j) = \sum_{L, S_j} f(T_j \mid S_j, L)P(L)P(S_j), \quad j = 1, ..., m.$$

Figure 2 shows the HBN for an example logistics delay problem, with Boolean variables S_1 = Administration performed?, S_2 = Resources needed?, S_3 = Spares required?, S_4 = Spares available?, and S_5 = Recovery required?. Each of these influence the delay times, $T_1 =$ Delay time for administration, $T_2 =$ Delay time for resources, $T_3 =$ delay time for spares, $T_4 =$ delay time for recovery.

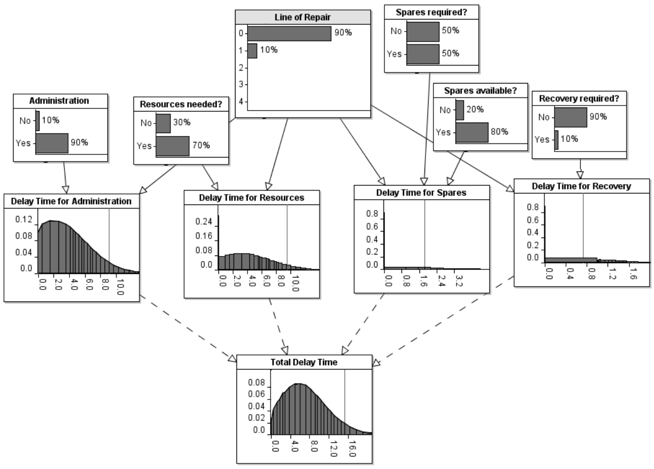

Fig. 2. HBN modelling a four delay event logistics chain with a mixture of normal delay times dependent on discrete delay events. Marginal distributions superimposed on the graph.

Table 2 shows an example with three active lines of repair and mean and s.d. parameter values for each delay event. Also, the prior probabilities for the switch variables are:

$$P(S_1 = \text{Yes}) = 0.9$$
$$P(S_2 = \text{Yes}) = 0.7$$
$$P(S_3 = \text{Yes}) = 0.5$$
$$P(S_4 = \text{Yes}) = 0.8$$
$$P(S_5 = \text{Yes}) = 0.1$$

The prior probabilities for each line of repair are shown in Table 2. For simplicity three repair lines are not used.

The resulting distribution for the total delay time for this example is shown in Figure 3. The mean total delay time for the system as calculated from the HBN is $E(R) = 7.17$ hours and the 95th percentile is 15.32 hours.

It is straightforward to see how we could extend the modelling approach used here to scheduled maintenance activities and to more complex supply processes containing cascading sets of conditional events.

Table 2 Logistical delay parameter values

L (lines of repair)		**0**	**1**	**2**	**3**	**4**
Probability of line	$p(L)$	0.9	0.1	0	0	0
Administration time (hours)	μ	2	2	-	-	-
	σ	4	4			
Resource time (hours)	μ	3	4	-	-	-
	σ	4	4			
Spares time (hours)	μ	1	2	-	-	-
	σ	2	2			
Recovery time (hours)	μ	0	0	-	-	-
	σ	1	1			

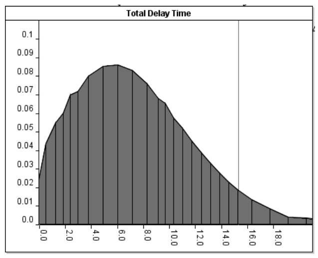

Fig. 3. Probability distribution for total delay time as a mixture of truncated Normal delay times distributions.

5. BAYESIAN MODEL FOR SYSTEM RELIABILITY

Let us now consider a system comprised of n subsystems $k = 1, 2, ..., n$, but we have not as yet been able to directly test the reliability of each one. Instead we can estimate the reliability of each subsystem from historical data gathered on an arbitrary number of similar subsystems drawn from the same family as the subsystem of interest. Thus the unseen subsystem is considered to be exchangeable with the historical subsystems.

In order to assess the reliability of the similar subsystems, we assume that a series of reliability tests have been conducted under the same operational settings or that directly comparable field data is available. The data resulting from these tests consist in the time to failure observed at the point of failure (assuming no censoring). Here we make the assumption that the time to failure (TTF) distribution, τ_k , for a component in the system is Weibull (the TTF is not constant over time i.e. wear out):

$$f_{\tau_k}(t \mid \lambda, \psi) = \frac{\psi}{\lambda}\left(\frac{t}{\lambda}\right)^{\psi - 1} e^{-(t/\lambda)^{\psi}}$$

where λ is the shape parameter and ψ is the scale parameter.

However, because reliability data can be sparse and heavily censored, additional information in the form of expert judgement plays an important role in the definition of statistical reliability models. Here we choose illustrative distributions for the Weibull shape and scale parameters, but in practice these might be elicited by experts. For example here we use:

$$\lambda \sim Triangular(1.5, 2)$$

$$\psi \sim Uniform(10E-6, 200000)$$

There are two points to make about the example distributions we have used for the prior parameters. Since no joint conjugate prior is available when λ, ψ are both assumed unknown, their prior distributions are specified independently. In our HBN modeling approach there is no restriction on the prior distributions for the parameters of the model.. Secondly, the prior distributions are based on past experience, and in this particular case, asking experts to use a triangular or uniform distribution is relatively easier compared to using other more complex distributions.

The HBN model used to calculate the subsystem TTF for a subsystem, τ_k, $k = 1, 2, ..., n$, is shown in Figure 4.

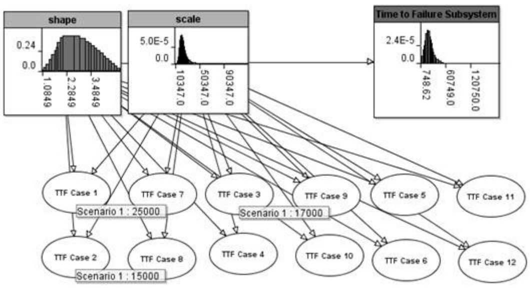

Fig. 4. HBN model for inferring TTF of a subsystem from similar subsystems.

For a system S comprised of n subsystems $k = 1, 2, ..., n$ and arranged in series (OR) configuration, the TTF, τ_S, is simply $\tau_S = \min_k \{\tau_k\}$ and if configured in parallel $\tau_S = \max_k \{\tau_k\}$. To calculate the system level TTF from more complex gates such as PAND see Marquez et al. (2008). Once we have the TTF distribution for the system we can then easily derive the Mean Time Between Failures statistic (MTBF).

6. AVAILABILITY

Availability is defined as the long run probability of the system being available for use at any point in time, Billinton and Allan (1992). This is expressed as a point estimate and calculated as from the mean delay and reliability point estimates:

Inherent Availability: $$A_I = \frac{MTTF}{MTTF + MRT}$$

, where MRT is the mean repair time and $MTTF$ is the mean time to failure.

Operational Availability: $$A_O = \frac{MTTF}{MTTF + MRT + MDT}$$

, where MDT is the mean logistical delay time.

Given that the summary statistics are constants we need to calculate the values for A_I and A_O in the HBN using the support for constants in the dynamic discretization algorithm.

7. SCHEDULED MAINTENANCE

For the purpose of failure prevention, scheduled maintenance is typically carried out at fixed intervals or when fixed exposure targets have been reached. For many applications it is convenient to model scheduled maintenance frequency and duration times as deterministic constants.

Given this deterministic assumption we can characterise the total availability, related to scheduled maintenance, A_S, of the scheduled maintenance event, as the product of the availability induced by each scheduled maintenance event, A_i, $i = 1, 2, ..., n$:

$$A_S = \prod_{i=1}^{n} A_i$$

We can therefore adjust the system level inherent and operational availabilities, to account for scheduled maintenance, and obtain revised values, $A_I^{'}$ and $A_O^{'}$, where:

$$A_I^{'} = A_I A_S \text{ and } A_O^{'} = A_O A_S$$

We can then calculate these within the HBN using the same algorithm as before.

8. APPLICATION SUPPORT

Practical modelling of repairable systems using HBNs requires a modelling environment to design the model structure, encode design assumptions and specify model dependencies. The model described here was developed using AgenaRisk Enterprise Edition (EE), Agena Ltd. (2009). This allows the user to declare the system structure and specify how the application should automatically generate the corresponding HBN model from this structure. Behind the scenes AgenaRisk EE applies a relational object model to encode the allowable abstract system hierarchy and specify how this is linked to the required HBN models. The relational model is then encoded in XML and connectivity for data import/export is supported via a database interface or via a CSV file format interface for connecting to Excel.

The resulting application can then be executed with the AgenaRisk desktop as is shown by the screenshot given in Figure 5.

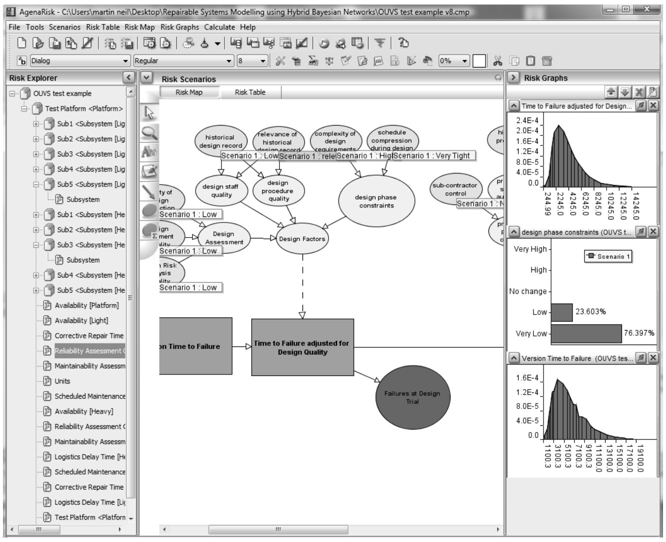

Fig. 5. Screenshot showing AgenaRisk application support for building and executing the HBN

9. CONCLUSIONS

We have shown how we can use Hybrid Bayesian Networks (HBNs) to model renewable systems involving repairs and logistical delays. Particular focus has been given to modelling mixtures of lognormally distributed repair distributions and this has been extended into the logistical delays case, involving the summation of delay times from many discrete events with normally distributed delay times.

We show how to assess the reliability of subsystems using a Bayesian model coupled with component aging assumption and integrating data with expert opinion. These subsystem TTF distributions could then be aggregated to provide a system level estimate using appropriate logical gates that reflect system structure. The combined scheme of dynamic discretization and robust propagation algorithms on HBNs can be used to obtain accurate results, offering a viable alternative to Monte Carlo simulation approaches, implemented within an easy-to-use and user friendly environment. Finally we show the simple step of calculating the operational and inherent availability of the system from the various delay and failure distributions and also demonstrated how point values for availabilities, with and without scheduled maintenance, can be obtained.

REFERENCES

Agena Ltd. (2009), www.agenarisk.com.

Billinton R. and Allan R. N. (1992) *Reliability Evaluation of Engineering Systems: Concepts and Techniques*. Plenum Press, New York, 1992.

Cobb, B., and Shenoy, P. (2003). Inference in Hybrid Bayesian Networks with Mixtures of Truncated Exponentials, *University of Kansas, School of Business, working paper* 294.

Elsayed, E. (1996). *Reliability Engineering*, Addison Wesley, Reading, MA.

Huang, C., and Darwiche, A. (1996). Inference in belief networks: A procedural guide. *Int. J. Approx. Reasoning* 15(3), 225-263.

Hugin. www.hugin.com. 2005.

Jensen, F., Lauritzen, S.L., and Olesen, K. (1990). Bayesian updating in recursive graphical models by local computations, *Computational Statistics*, 4, 260-282.

Koller, D., Lerner, U., and Angelov, D. (1999). A general algorithm for approximate inference and its applications to Hybrid Bayes Nets, in K.B Laskey and H. Prade (eds.), *Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence*. 324-333.

Kozlov, A.V., and Koller, D. (1997). Nonuniform dynamic discretization in hybrid networks, in D. Geiger and P.P. Shenoy (eds.), *Uncertainty in Artificial Intelligence*, 13, 314-325.

Lauritzen, S.L., and Speigelhalter, D.J. (1988). Local computations with probabilities on graphical structures and their application to expert systems (with discussion), *Journal of the Royal Statistical Society Series B*, 50(2), 157-224.

Marquez, D., Neil, M., and Fenton, N. (2008). Improved reliability modeling using Bayesian networks and dynamic discretisation. Submitted to *Reliability Engineering & System Safety*, November 2008.

Moral, S., Rumı, R., and Salmeron, A. (2001). Mixtures of truncated exponentials in hybrid Bayesian networks, in P. Besnard and S. Benferhart (eds.), *Symbolic and Quantitative Approaches to Reasoning under Uncertainty*, Lecture Notes in AI, 2143. 156-167.

Murphy, K. (1999). A variational approximation for Bayesian networks with discrete and continuous latent variables, in Laskey and H. Prade (eds.), *Uncertainty in AI*, 15, 467-475.

Neil, M., Fenton, N., Forey, S., and Harris, R. (2001). Using Bayesian belief networks to predict the reliability of military vehicles, *IEE Computing and Control Engineering*, 12(1), 11-20.

Neil, M., Malcolm, B., and Shaw, R. (2003). Modeling an air traffic control environment using Bayesian belief networks, *21st International System Safety Conference*, Ottawa, Ontario, Canada.

Neil, M., Tailor, M., Marquez, D., Fenton, N., and Hearty, P. (2008). Modelling dependable systems using hybrid Bayesian networks. *Reliability Engineering & System Safety*. 93(7), 933-939.

Neil, M., Tailor, M,. and Marquez, D. (2007). Inference in Bayesian networks using dynamic discretization, *Journal of Statistics and Computing*, Springer Netherlands, 17(3).

Netica. www.norsys.com. 2005.

Pearl, J. (1993). Graphical models, causality, and intervention, *Statistical Science*, 8(3), 266-273.

Shacter, R., and Peot, M. (1989). Simulation approaches to general probabilistic inference on belief networks. *In Proceedings of the 5th Annual Conference on Uncertainty in AI* (UAI). 221-230.

Shenoy, P., and Shafer, G. (1990). Axioms for probability and belief-function propagation, *Readings in uncertain reasoning*, 575-610. Morgan Kaufmann Publishers Inc.