Practical Aspects of Solving Hybrid Bayesian Networks Containing Deterministic Conditionals *

Prakash P. Shenoy[1†], Rafael Rumí[2‡], Antonio Salmerón[2§]
[1]School of Business, University of Kansas,
Lawrence, KS 66045-7601 USA
[2]Department of Mathematics, University of Almería,
E-04120 Almería, Spain

Abstract

In this paper we discuss some practical issues that arise in solving hybrid Bayesian networks that include deterministic conditionals for continuous variables. We show how exact inference can become intractable even for small networks, due to the difficulty in handling deterministic conditionals (for continuous variables). We propose some strategies for carrying out the inference task using mixtures of polynomials and mixtures of truncated exponentials. Mixtures of polynomials can be defined on hypercubes or hyper-rhombuses. We compare these two methods. A key strategy is to re-approximate large potentials with potentials consisting of fewer pieces and lower degrees/number of terms. We discuss several methods for re-approximating potentials. We illustrate our methods in a practical application consisting of solving a stochastic PERT network.

1. Introduction

Hybrid Bayesian networks are Bayesian networks (BNs) that include a mix of discrete and continuous random variables. A random variable is *discrete* if its state space is countable, and is *continuous* otherwise. In a BN, each variable is associated with a conditional distribution for it given each state of its parents. A conditional distribution for a variable is said to be *deterministic* if its variances are all zeroes (for each state of its parents).

The first proposal of an exact algorithm for hybrid BNs was developed for the case in which the joint distribution of all variables is a *mixture of Gaussians* (MoG).[1] Some limitations of the MoG model are that it is restricted to

*Preliminary versions of this paper were presented at the ISDA'2011 conference and PGM'2012 workshop

†Author to whom all correspondence should be addressed; e-mail:pshenoy@ku.edu
‡rrumi@ual.es
§antonio.salmeron@ual.es

network topologies in which discrete variables do not have continuous parents, and each conditional for a continuous variable has to be a conditional linear Gaussian, i.e., a Gaussian distribution whose mean is a linear function of its continuous parents, and whose variance is a constant. Also, during the solution phase, continuous variables have to be marginalized before discrete ones and this restriction can lead to large cliques,[2] and consequently large memory storage requirements. It has been shown that inference in MoG networks is NP-hard, even for network structures for which inference in the discrete case is easy.[2]

A more general technique is based on the use of *mixtures of truncated exponentials* (MTEs).[3] The MTE model does not impose any topological restrictions on its networks, and is compatible with any efficient algorithm for exact inference that requires only the combination and marginalization operations, such as the Shenoy-Shafer[4,5] and variable elimination methods.[6] Furthermore, MTEs have shown a remarkable ability for fitting many commonly used univariate probability density functions (PDFs).[7]

Hybrid BNs can also be solved by discretizing the continuous variables, so that all the existing methodology for discrete BNs can be applied with any further modification. The most prominent proposal in this direction is the so-called *dynamic discretization*,[8] where inference is carried out iteratively in an any-time manner, seeking for better representations of high density areas. A study of the complexity of the MTE approach versus discretization can be found in existing literature.[9,10]

The most recent proposal for dealing with hybrid BNs is based on the use of *mixtures of polynomials* (MOPs).[11] Like MTEs, MOPs have high expressive power, but the latter are superior in dealing with deterministic conditionals for continuous variables.[5,11] Also, a MOP approximation of a PDF can be easily found using Lagrange interpolating polynomials with Chebyshev points.[12] Both MTEs and MOPs can be seen as instantiations of a more general framework known as *mixtures of truncated basis functions* (MoTBFs).[13]

In this paper, we discuss some practical issues that have to be addressed in order to make inference in hybrid BNs tractable, even for small problems, when deterministic conditionals are present. Dynamic discretization can also be used in this context,[8,14] but the approach we follow here is based on the use of MOPs and MTEs as discretization is in fact a particular case of these models.[13] A key strategy is re-approximation of MOPs (MTEs) with fewer pieces and lower degrees (fewer terms) during the solution phase. We compare the sizes, computation time and accuracy of MOPs defined on hypercubes and hyper-rhombuses. We compare the performance of MOPs and MTEs in this context, through an example arising from a stochastic PERT network.[15] We also carry out an experiment illustrating how the complexity grows with the size of the model.

Contributions The main contributions of this paper are as follows. First, in the case of MOPs, they can be defined on either hypercubes or hyper-rhombuses when deterministic conditionals are present in the network. In this paper, we do a small comparison of these two possibilities. Second, in the case of MOPs or MTEs, we describe a re-approximation method by dropping pieces. Third, in the case of MOPs, we describe a re-approximation method using Lagrange interpolating polynomials with Chebyshev points.[12] Fourth, in the case of MTEs, we describe a re-approximation method using numeric least squares. Fifth, we demonstrate the efficacy of the re-approximation methods by solving a small hybrid Bayesian network with 2 discrete variables, and 10 continuous variables of which 4 have deterministic conditionals. We were unable to solve this problem without using re-approximations.

Limitations Some limitations of our work are as follows. First, we do not describe the worst-case complexity of inference in hybrid Bayesian networks with deterministic conditionals using MOPs/MTEs. Instead, we carry out an experiment illustrating how the complexity grows with the size of the model. Due to the fact that MOP/MTE-based methods are more general than MoG methods, we suspect that it is at least as bad as MoG networks, if not worse. Second, although we do a small comparison of hypercubes and hyper-rhombuses in the case of MOPs, a more systematic study is needed with more examples. Based on our limited set of experiments, we believe that hypercubes are more tractable than hyper-rhombuses, as long as an appropriate approximation can be found for each hypercube. Third, for all the re-approximations methods we have described, there are many judgments that have to be made for which we have no systematic rules for doing so. Fourth, it would be useful to know approximately the size of problems that can be solved using the re-approximation methods. Fifth, it would be useful to have some error bounds on the results of inference using MOP/MTE-methods, with or without re-approximations. All of these limitations need further research.

2. MTEs and MOPs

We will use uppercase letters to denote random variables, and boldfaced uppercase letters to denote random vectors, e.g. $\mathbf{X} = \{X_1, \ldots, X_n\}$, and its state space will be written as $\Omega_{\mathbf{X}}$. Lowercase letters x (or $\mathbf{x}$) will denote elements of Ω_X (or $\Omega_{\mathbf{X}}$). The MTE model[3] is defined as follows.

Definition 1. *Let* $\mathbf{X}$ *be a mixed* n*-dimensional random vector. Let* $\mathbf{Y} = (Y_1, \ldots, Y_d)^{\mathsf{T}}$ *and* $\mathbf{Z} = (Z_1, \ldots, Z_c)^{\mathsf{T}}$ *be its discrete and continuous parts respectively. A function* $f : \Omega_{\mathbf{X}} \mapsto \mathbb{R}_0^+$ *is a* mixture of truncated exponentials

(MTE) potential *if for each fixed value* $\mathbf{y} \in \Omega_{\mathbf{Y}}$ *of the discrete variables* $\mathbf{Y}$*, the potential over the continuous variables* $\mathbf{Z}$ *is defined as:*

$$f(\mathbf{z}) = a_0 + \sum_{i=1}^{m} a_i \exp\left\{\mathbf{b}_i^{\mathsf{T}} \mathbf{z}\right\}, \tag{1}$$

for all $\mathbf{z} \in \Omega_{\mathbf{Z}}$*, where* $a_i \in \mathbb{R}$ *and* $\mathbf{b}_i \in \mathbb{R}^c$*,* $i = 1, \ldots, m$*. We also say that* f *is an MTE potential if there is a partition* $D_1, \ldots, D_k$ *of* $\Omega_{\mathbf{Z}}$ *into hypercubes and in each one of them,* f *is defined as in Eq.* (1)*. In this case, we say* f *is a* k*-piece,* m*-term MTE potential.*

Mixtures of polynomials (MOPs) were initially proposed as modeling tools for hybrid BNs.[11] The original definition is similar to MTEs, in the sense that they are piecewise functions defined on hypercubes. A more general definition, where the hypercube condition is relaxed, can be stated as follows.[12]

Definition 2. *Let* $\mathbf{X}, \mathbf{Y}$ *and* $\mathbf{Z}$ *be as in Definition 1. A function* $f : \Omega_{\mathbf{X}} \mapsto \mathbb{R}_0^+$ *is a* mixture of polynomials (MOP) potential *if for each fixed value* $\mathbf{y} \in \Omega_{\mathbf{Y}}$ *of the discrete variables* $\mathbf{Y}$*, the potential over* $\mathbf{Z}$ *is defined as:*

$$f(\mathbf{z}) = P(\mathbf{z}), \tag{2}$$

for all $\mathbf{z} \in \Omega_{\mathbf{Z}}$*, where* $P(\mathbf{z})$ *is a multivariate polynomial in variables* $\mathbf{Z} = (Z_1, \ldots, Z_c)^{\mathsf{T}}$*. We also say that* f *is a MOP potential if there is a partition* $D_1, \ldots, D_k$ *of* $\Omega_{\mathbf{Z}}$ *into* hyper-rhombuses *and in each one of them,* f *is defined as in Eq.* (2)*.*

The fact that the elements in the partition are hyper-rhombuses, means that for any ordering of the variables $Z_1, \ldots, Z_c$, for each D_i it holds that $l_{1i} \leq z_1 \leq u_{1i}, l_{2i}(z_1) \leq z_2 \leq u_{2i}(z_1), \ldots, l_{ci}(z_1, \ldots, z_{c-1}) \leq z_c \leq u_{ci}(z_1, \ldots, z_{c-1})$, where l_{1i} and u_{1i} are constants, and $l_{ji}(z_1, \ldots, z_{j-1})$ and $u_{ji}(z_1, \ldots, z_{j-1})$ are linear functions of $z_1, \ldots, z_{j-1}$ for $j = 2, \ldots, c$, and $i = 1, \ldots, k$. Figure 1 shows the difference between a hypercube and a hyper-rhombus.

MTEs and MOPs are closed under multiplication, addition, and integration. However, integrating over hyper-rhombuses is in general more complex than over hypercubes. The advantage is that by using hyper-rhombuses, it is easier to represent models such as the *conditional linear Gaussian*, where the conditional distribution of a variable may depend on the values of its continuous parents in the network. Unfortunately, MTEs cannot be defined on hyper-rhombuses, as the integration operation would not remain closed for that class.

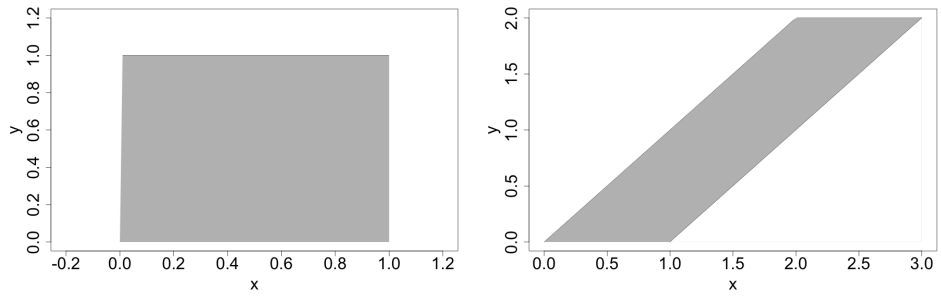

Figure 1: A hypercube defined by $0 \leq x \leq 1$, $0 \leq y \leq 1$ (left) and a hyper-rhombus defined by $0 \leq y \leq 2$, $y \leq x \leq y+1$ (right). Note that the borders of the region are not constant in the hyper-rhombus case.

3. Inference in Hybrid BNs with Deterministic Conditionals

The inference task in hybrid BNs with deterministic conditionals has been already studied.[5] Traditional algorithms for inference in hybrid BNs rely on the assumption that all the conditionals in the network belong to the same class, and that the operations used during inference are closed within the same class. This is not necessarily the case for deterministic conditionals.

For instance, assume we have a BN with three continuous variables B, C and D, such that $D = \max\{B, C\}$. In such case, existing procedures[5] cannot be directly applied. However, the max deterministic function can be converted to a linear function by introducing an auxiliary variable A with two states a and na, which denote whether $B \geq C$ or $B < C$, respectively, obtaining the equivalent representation displayed in the right hand side of Figure 2, where $D = B$ if $A = a$ and $D = C$ if $A = na$. Note that after the transformation, the distribution is the same as in the original network. However, the price to pay is the introduction of additional complexity, as there will be a new variable with as many parents as de deterministic variable. This gives an idea of the increased complexity when dealing with deterministic conditionals with respect to plain hybrid BNs.

Whether or not a deterministic conditional can be properly handled, depends on the model used. We will analyze the situation from the point of view of MTEs and MOPs.

3.1. MTE Representation of Conditionals

MTEs can be used to accurately approximate several univariate distributions.[7] The approximation of conditional densities using MTEs is more dif-

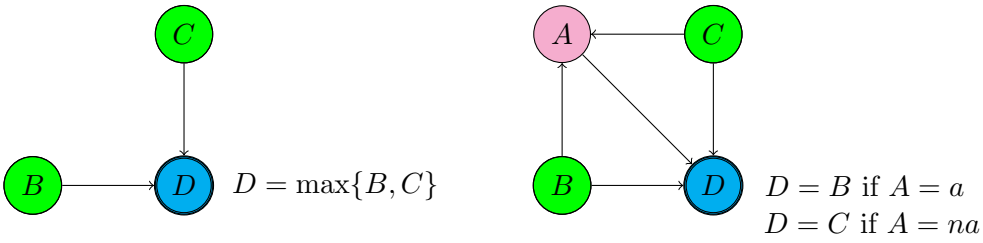

Figure 2: A *max* conditional (left) and its transformation (right).

ficult,[16] as the hypercube condition in the definition of an MTE function means that the value of a continuous parent has to be a constant.[17] Therefore, this implies that a conditional density is approximated by MTEs by partitioning the state space of the continuous parents into hypercubes, and then fitting a univariate MTE in each hypercube. The resulting conditional MTE density is called a *mixed tree*.[18]

An additional problem is found when attempting to deal with deterministic conditionals. MTEs are not closed with respect to the *convolution* operation required by the *sum* conditional. Consider two independent random variables $X_1, X_2 \sim Exp(\mu = 1)$ and $Y = X_1 + X_2$. The marginal PDF of Y is $Gamma[r = 2, \mu = 1]$, which is not an MTE function. The reason is that such a marginal is obtained through the so-called *convolution* operation as

$$f_Y(y) = \int_{-\infty}^{\infty} f_{X_1}(x_1)\, f_{X_2}(y - x_1)\, \mathrm{d}x_1 = \int_0^y e^{-x_1}\, e^{-(y-x_1)}\, \mathrm{d}x_1 = y\, e^{-y} \quad (3)$$

for $y > 0$, which is not an MTE according to Definition 1. This is a consequence of the fact that even though $f_{X_2}(x_2)$ is defined on hypercubes, $f_{X_2}(y - x_1)$ is no longer defined on hypercubes, and therefore, the limits of integration in Eq. (3) are not all constants.

One solution to this problem is to approximate the function $f_{X_2}(y - x_1)$ on hypercubes using a *mixed tree* approach.[18] In order to illustrate the mixed tree approach, consider a hybrid BN formed by variables X_1, X_2 and X_3. $X_1 \sim N(3, 1)$, $X_2|x_1 \sim N(6 + 2x_1, 2^2)$ and $X_3 = X_1 + X_2$. If $Z \sim N(0, 1)$, $f(\cdot)$ is an MTE approximation of the PDF of Z, and $Y = \sigma Z + \mu$, where $\sigma > 0$ and μ are real constants, then $Y \sim N(\mu, \sigma^2)$, and an MTE approximation of the PDF of Y is given by $g(y) = \frac{1}{|\sigma|} f(\frac{y-\mu}{\sigma})$. Suppose $f(\cdot)$ is a 2-piece MTE approximation of the PDF of $N(0, 1)$ on the domain $(-3, 3)$, where the two pieces are $(-3, 0)$ and $[0, 3)$.[7] Then, $g_1(x_1) = f(x_1 - 3)$ is an MTE approximation of the PDF of $X_1 \sim N(3, 1)$ on the domain $(0, 6)$. However, $g_2(x_1, x_2) = f(\frac{x_2 - 6 - 2\,x_1}{2})/2$ is not an MTE since it would be defined on regions such as $-3 < \frac{x_2 - 6 - 2\,x_1}{2} < 0$, which are not hypercubes. So we partition the domain of X_1 into equal-size intervals, and assume that

x_1 is a constant in each interval equal to the mid-point of the interval. Thus, $g_{2c}(x_1, x_2)$, as described in Eq. (4), is a 3-point mixed tree MTE approximation of $g_2(x_1, x_2)$ representing the conditional PDF of X_2 given x_1.

$$g_{2c}(x_1, x_2) = \begin{cases} f(\frac{x_2-8}{2})/2 & \text{if } 0 < x_1 < 2, \\ f(\frac{x_2-12}{2})/2 & \text{if } 2 \leq x_1 < 4, \\ f(\frac{x_2-16}{2})/2 & \text{if } 4 \leq x_1 < 6. \end{cases} \tag{4}$$

Notice that this mixed-tree method can also be used with MOPs. Next, we wish to compute the marginal PDF of X_3. The conditional associated with X_3 is $g_3(x_1, x_2, x_3) = \delta(x_3 - x_1 - x_2)$, where δ is the Dirac delta function (see[5] for a detailed description on the use of Dirac delta functions for handling deterministic relationships). To compute the marginal of X_3, we first marginalize X_2 and then X_1. The result after marginalizing X_2 is $g_4(x_1, x_3) = \int_{-\infty}^{\infty} g_{2c}(x_1, x_2)\, \delta(x_3 - x_1 - x_2))\mathrm{d}x_2 = g_{2c}(x_1, x_3 - x_1)$, which represents the conditional of X_3 given x_1.

Now we notice that $g_4(x_1, x_3)$ is not defined on hypercubes anymore since we have regions such as $a \leq x_3 - x_1 < b$, where a and b are constants. So we approximate $g_4(x_1, x_3)$ by $g_{5c}(x_1, x_3)$ using mixed trees as follows:

$$g_{5c}(x_1, x_3) = \begin{cases} g_{2c}(1, x_3 - 1) & \text{if } 0 < x_1 < 2, \\ g_{2c}(3, x_3 - 3) & \text{if } 2 \leq x_1 < 4, \\ g_{2c}(5, x_3 - 5) & \text{if } 4 \leq x_1 < 6. \end{cases} \tag{5}$$

Notice that $g_{5c}(x_1, x_3)$ is an MTE function since it is defined on hypercubes. Next we marginalize X_1 as follows (resulting in the marginal PDF $g_{6c}(\cdot)$ of X_3):

$$g_{6c}(x_3) = \int_{-\infty}^{\infty} g_1(x_1)\, g_{5c}(x_1, x_3)\, \mathrm{d}x_1. \tag{6}$$

Since MTEs are closed under multiplication and integration, $g_{6c}(\cdot)$ is an MTE function. A cost of the mixed-tree approach to maintain the hypercube nature of the pieces is the increase in the number of pieces. Thus, if $f(\cdot)$ is a 1-piece MTE approximation of the PDF of $N(0, 1)$, then $g_1(d_1)$ is a 1-piece MTE potential, $g_{2c}(d_1, d_3)$ is a 3-piece MTE potential and $g_{5c}(d_1, c_3)$ is a 3-piece MTE potential. Another cost is loss of accuracy. We have used 3-point mixed trees for our illustration. We could use more points to improve accuracy of the computed potentials, but this would mean more pieces. We will discuss this in more detail in Section 3.3 for the case of MOPs.

3.2. MOP Representation of Conditionals

The problem of fitting univariate and multi-variate PDFs using MOPs has already been approached using the Taylor series approximation of differen-

tiable functions.[11] More recently, the ability of MOPs to fit multivariate conditional linear Gaussian distributions was significantly improved by means of allowing the functions to be defined into hyper-rhombuses rather than into hypercubes.[12] Also, an improved method for finding MOP approximations based on Lagrange interpolating polynomials with Chebyshev points has been proposed.[12]

To illustrate this improvement, consider the BN used in Section 3.1 formed by three continuous variables X_1, X_2 and X_3. Suppose $f(\cdot)$ is a 2-piece, 3-degree MOP approximation of the PDF of $N(0,1)$ on the domain $(-3,3)$.[12] Then, $g_1(x_1) = f(x_1 - 3)$ is a 2-piece, 3-degree MOP approximation of the PDF of $X_1 \sim N(3,1)$ on the domain $(0,6)$. Finally, $g_2(x_1, x_2) = f(\frac{x_2-6-2x_1}{2})/2$ is a 2-piece, 3-degree MOP approximation of the conditional PDF of X_2 given x_1. Notice that $g_2(x_1, x_2)$ is defined on hyper-rhombuses, e.g., $-3 < \frac{x_2-6-2x_1}{2} < 0$, etc, and not on hypercubes.

Unlike MTEs, MOPs are closed under the operations required for *sum* conditionals. Thus, after the elimination of X_2, $g_5(x_1, x_3) = g_2(x_1, x_3 - x_1)$ is a MOP, and after the elimination of X_1, $g_6(x_3) = \int_{-\infty}^{\infty} g_1(x_1)\, g_5(x_1, x_3)\, \mathrm{d}x_1$ is also a MOP. So in the case of MOPs, we have a choice of using MOPs defined on hypercubes (using mixed trees) or on hyper-rhombuses. In the next subsection, we compare the two alternatives in terms of the sizes of MOP potentials (pieces and degrees), computation time, and accuracy of resulting MOP potentials. All computations were done in *Mathematica*® v. 8.0.4, running on an *Apple* iMac with 3.4 GHz *Intel* Core i7 processor and 16 GB memory. [1]

3.3. Hypercube vs. Hyper-rhombus MOP Representations of Gaussian PDFs

In order to compare the hypercube and the hyper-rhombus approximations of Gaussian PDFs, we do a small experiment. Consider again the BN formed by X_1, X_2 and X_3. We will represent the PDFs of X_1 and X_2 using hypercubes and hyper-rhombuses. We will compute the marginal PDFs of X_2 and X_3, and compare the sizes of resulting MOP marginals, time required for the computation of marginals, and the quality of the approximation.

3.3.1. Using Hypercube MOPs

We start with $f(z)$, a 2-piece, 3-degree MOP approximation of the PDF of $N(0,1)$ on the interval $(-3,3)$.[11] Then as discussed before, $g_1(x_1) = f(x_1 - 3)$ is a 2-piece, 3-degree MOP approximation of the PDF of $N(3,1)$ on $(0,6)$, and $g_{2c}(x_1, x_2)$ as defined in Eq. (4) is a 6-piece, 3-degree hypercube MOP approximation of the conditional PDF of X_2 given X_1. The marginal

[1] The *Mathematica*® notebooks used in this paper can be downloaded from `http://elvira.ual.es/DetCond`

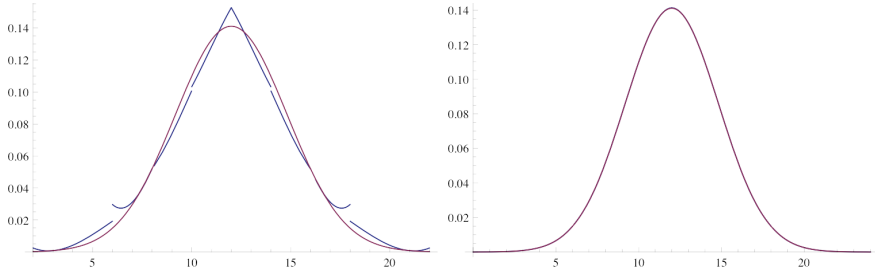

Figure 3: Left: A plot of $g_{4c}(\cdot)$ (blue) overlaid on the PDF of $N(12, 8)$ truncated to $(2, 22)$ (red). Right: A plot of $g_4(\cdot)$ (blue) overlaid on the PDF of $N(12, 8)$ truncated to $(0, 24)$ (red).

PDF of X_2 is given by

$$g_{4c}(x_2) = \int_{-\infty}^{\infty} g_1(x_1)\, g_{2c}(x_1, x_2)\, \mathrm{d}x_1. \tag{7}$$

It takes approximately 1.41 seconds to compute the integral above and results in a 8-piece, 3-degree MOP approximation of the PDF of X_2 on the interval $(2, 22)$. The exact marginal distribution of X_2 is $N(12, 8)$. A plot of g_{4c} overlaid on the plot of the PDF of $N(12, 8)$ truncated to $(2, 22)$ is shown at the left in Figure 3.

We will measure the accuracy of a PDF with respect to another defined on the same domain by four different measures, the Kullback-Leibler (KL) divergence, maximum absolute deviation, absolute error of the mean, and absolute error of the variance. These measures are defined as follows.[12]

If f is a PDF on the interval (a, b), and g is a PDF that is an approximation of f such that $g(x) > 0$ for $x \in (a, b)$, then the *KL divergence* between f and g, denoted by $KL(f, g)$, is defined as

$$KL(f, g) = \int_a^b \ln\left(\frac{f(x)}{g(x)}\right) f(x)\, dx. \tag{8}$$

$KL(f, g) \geq 0$, and $KL(f, g) = 0$ if and only if $g(x) = f(x)$ for all $x \in (a, b)$.

The *maximum absolute deviation* between f and g, denoted by $MAD(f, g)$, is given by:

$$MAD(f, g) = \sup\{|f(x) - g(x)| : a < x < b\}. \tag{9}$$

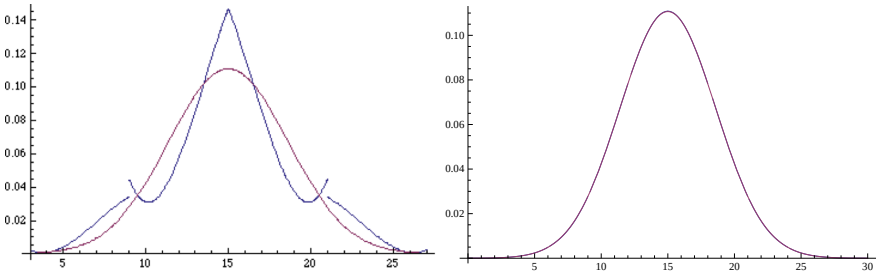

Figure 4: Left: A plot of $g_{6c}(\cdot)$ (blue) overlaid on the PDF of $N(15, 13)$ truncated to $(3, 27)$ (red). Right: A plot of $g_6(\cdot)$ (blue) overlaid on the PDF of $N(15, 13)$ truncated to $(0, 30)$ (red). Note that $g_{6c}(\cdot)$ is computed using hypercube MOPs while $g_6(\cdot)$ is obtained using hyper-rhombuses MOPs.

The *absolute error of the mean*, denoted by $AEM(f, g)$, and the *absolute error of the variance*, denoted by $AEV(f, g)$, are given by:

$$AEM(f, g) = |E(f) - E(g)|, \quad AEV(f, g) = |V(f) - V(g)|, \tag{10}$$

where $E(\cdot)$ and $V(\cdot)$ stand for expected value and variance of a PDF.

The goodness of fit measures for g_{4c} vs. f_{X_2}, the PDF of $N(12, 8)$ truncated to (2, 22), are displayed in Table 3, *hypercube* column.

Next, we compute the PDF of X_3. After marginalization of X_2, we have $g'_5(x_1, x_3) = g_{2c}(x_1, x_3 - x_1)$, which is not defined on hypercubes. So we approximate g'_5 by $g_{5c}(x_1, x_3)$ as defined in Eq. (5). After marginalization of X_1, we get $g_{6c}(x_3)$, which is a 6-piece, 3-degree MOP approximation of the PDF of X_3 defined on the domain $(3, 27)$. It takes approximately 1.31 seconds to compute $g_{6c}(\cdot)$. The exact marginal distribution of X_3 is $N(15, 13)$. A plot of g_{6c} overlaid on the plot of the PDF of $N(15, 13)$ truncated to $(3, 27)$ is shown at the left in Figure 4.

The goodness of fit measures for g_{6c} vs. f_{X_3}, the PDF of $N(15, 13)$ truncated to [3, 27], are displayed in Table 3, *hypercube* column.

3.3.2. Using Hyper-rhombus MOPs

As in the case of hypercubes, we start with $f(\cdot)$, a 2-piece, 3-degree MOP approximation of the PDF of $N(0, 1)$. Then $g_1(x_1) = f(x_1 - 3)$ is a MOP approximation of the PDF of X_1, and $g_2(x_1, x_2) = f(\frac{x_2 - 6 - 2x_1}{2})/2$ is a hyper-rhombus MOP approximation of the conditional PDF of $X_2|x_1$. The marginal PDF of X_3 is computed as

$$g_4(x_2) = \int_{-\infty}^{\infty} g_1(x_1)\, g_2(x_1, x_2)\, \mathrm{d}x_1. \tag{11}$$

It takes approximately 3.44 seconds to do this integral, and $g_4(\cdot)$ is computed as a 4-piece, 7-degree MOP on the domain $(0, 24)$. The exact marginal

distribution of X_2 is $N(12, 8)$. A plot of g_4 overlaid on the plot of the PDF of $N(12, 8)$ truncated to $(0, 24)$ is shown at the right in Figure 3.

The goodness of fit measures for g_4 vs. f'_{X_2}, the PDF of $N(12, 8)$ truncated to $[0, 24]$, are displayed in Table 3, *hyper-rhombus* column.

Finally, we compute the marginal PDF of X_3 as follows:

$$g_6(x_3) = \int_{-\infty}^{\infty} g_1(x_1)\, g_2(x_1, x_3 - x_1)\, \mathrm{d}x_1. \tag{12}$$

It takes approximately 4.98 seconds to do the integration in Eq.(12), and $g_6(\cdot)$ is computed as a 9-piece, 7-degree MOP on the domain $(0, 30)$. The exact marginal distribution of X_3 is $N(15, 13)$. A plot of g_6 overlaid on the plot of the PDF of $N(15, 13)$ truncated to $(0, 30)$ is shown at the right in Figure 4. The goodness of fit measures for g_6 vs. f'_{X_3}, the PDF of $N(15, 13)$ truncated to $(0, 30)$, are displayed in Table 3, *hyper-rhombus* column.

3.3.3. Comparison

We will compare the computed marginals of X_2 and X_3 in terms of the sizes of MOPs, time required for computation, and accuracy.

Sizes of MOPs. Consider the sizes of the MOP approximations of the marginals of X_2 and X_3 displayed in Table 1.

Size of MOPs	Hypercube	Hyper-rhombus
Marg. PDF of X_2	8-piece	4-piece
g_{4c} vs. g_4	3-degree	7-degree
Marg. PDF of X_3	6-piece	9-piece
g_{6c} vs. g_6	3-degree	7-degree

Table 1: Sizes of the MOP approximations of the marginals of X_2 and X_3.

For the marginal PDF of X_2, the hypercube MOP $g_{4c}(\cdot)$ has more pieces but fewer degrees than the corresponding MOP $g_4(\cdot)$. To see why, $g_{4c}(\cdot)$ is computed as described in Eq. (7). $g_1(x_1)$ is a 2-piece, 3-degree MOP and $g_{2c}(x_1, x_2)$ is a 6-piece, 3-degree MOP. After multiplication, $g_1(x_1)\, g_{2c}(x_1, x_2)$ is a 8-piece, 6-degree MOP defined on hypercubes. After integration with respect to d_1, the degrees associated with d_1 disappear, but none of the pieces do, resulting in a 8-piece, 3-degree MOP $g_{4c}(x_2)$. On the other hand, $g_4(\cdot)$ is defined as in Eq.(11). $g_1(x_1)$ is a 2-piece, 3-degree MOP, and $g_2(x_1, x_2)$ is a 2-piece, 3-degree MOP defined on hyper-rhombus. After multiplication, $g_1(x_1)\, g_2(x_1, x_2)$ is a 4-piece, 6-degree MOP defined on hyper-rhombus regions. After integration with respect to x_1, the result is a 6-piece, 7-degree MOP. Two of the six pieces are defined on singleton points ($x_2 = 18$, and $x_2 = 22$). The reason why the degree increases to 7 is because when integrating a MOP defined on a rhombus, the limits of integration are in terms

of x_2, and since $\int x_1^n \, \mathrm{d}x_1 = \frac{x_1^{n+1}}{n+1}$, the degree of the resulting MOP increases from 6 to 7.

For the marginal PDF of X_3, the hypercube MOP $g_{6c}(\cdot)$ has fewer pieces and fewer degrees than the corresponding MOP $g_6(\cdot)$. To see why, $g_{6c}(\cdot)$ is computed as described in Eq. (6). $g_1(x_1)$ is a 2-piece, 3-degree MOP and $g_{5c}(x_1, x_3)$ is a 6-piece, 3-degree MOP. After multiplication, $g_1(x_1)\, g_{5c}(x_1, x_3)$ is a 8-piece, 6-degree MOP defined on hypercubes. After integration with respect to x_1, the degrees associated with x_1 disappear, and three of the pieces do, resulting in a 6-piece, 3-degree MOP $g_{4c}(x_3)$. Two of the six pieces are defined on singleton regions ($x_3 = 9$ and 15). On the other hand, $g_6(\cdot)$ is defined as in Eq. (12). $g_1(x_1)$ is a 2-piece, 3-degree MOP, and $g_2(x_1, x_3 - x_1)$ is a 2-piece, 3-degree MOP defined on hyper-rhombus. After multiplication, $g_1(x_1)\, g_2(x_1, x_3 - x_1)$ is a 4-piece, 6-degree MOP defined on hyper-rhombus regions. After integration with respect to x_1, the result is a 9-piece, 7-degree MOP. One of the nine pieces is defined on a singleton point ($x_3 = 15$).

In summary, although mixed-tree hypercube MOPs initially require more pieces than hyper-rhombus MOPs, after integration, MOPs defined on hypercubes tend to lose pieces and lose degrees, whereas MOPs defined on hyper-rhombuses tend to increase pieces (some of these are defined on lower dimensional regions), and increase degrees.

Computation Time. Next, consider the times required for the computation of the marginals of X_2 and X_3 displayed in Table 2. Notice that the time required is a random variable. We repeated the experiment 10 times under identical conditions for both cases and computed the *mean* and *standard error* (SE) of the mean. The 97.5 percentile t-statistic is 2.25 and a 95% confidence interval for the mean is $mean \pm 2.25\, SE$.

Table 2: Times required for the computation of the marginals of X_2 and X_3.

Mean time in seconds (SE)	Hypercube	Hyper-rhombus
Marg. PDF of X_2	1.42	3.33
g_{4c} vs. g_4	(0.01)	(0.01)
Marg. PDF of X_3	1.35	4.92
g_{6c} vs. g_6	(0.009)	(0.003)

Integrating a MOP defined on hypercube regions is faster than integrating a MOP defined on hyper-rhombus regions. For the latter, we have to first solve linear inequalities and then do the integration where the limits of integrations are the solutions of the inequalities.

Accuracy. Finally, consider the accuracy of the computed marginals for X_2 and X_3 in terms of KL-divergence, maximum absolute deviation,

absolute error in the mean and absolute error in the variance, shown in Table 3.

Hyper-rhombus MOPs are more accurate than mixed-tree hypercube MOPs by two orders of magnitude or more. We can increase accuracy of mixed-tree hypercube MOPs by increasing # mixed-tree points (up to a point), but this will increase sizes and computation time. An important point is that in the case of mixed-tree hypercube MOPs, we have a choice of accuracy vs. computation time/sizes, which we don't with hyper-rhombuses.

Table 3: Accuracy of the computed marginals for X_2 and X_3.

Accuracy	Hypercube	Hyper-rhombus
Marg. PDF of X_2	KL: 0.0082	KL: 0.00002
g_{4c} vs. g_4	MAD: 0.0092	MAD: 0.00004
	AEM: 0.0	AEM: 0.0
	AEV: 1.1653	AEV: 0.0188
Marg. PDF of X_3	KL: 0.0480	KL: 0.00003
g_{6c} vs. g_6	MAD: 0.0360	MAD: 0.00004
	AEM: 0.0	AEM: 0.0
	AEV: 2.6735	AEV: 0.0284

4. Re-approximation of MOPs/MTEs

As we saw in the preceding section, in the process of integrating MOPs using convolutions, we may get pieces defined on lower-dimensional regions, which have no probabilities associated with them. For example, $g_6(x_3)$ is a 11-piece, 7-degree MOP. Three of the eleven pieces are defined on singleton regions, and these pieces have no probabilities associated with them. By re-approximating MOPs, we can reduce # pieces and degrees (# pieces and terms for MTEs), which will increase computational efficiency at a small cost in accuracy. We will describe two methods for re-approximating potentials. The first method is applicable both for MOPs and MTEs, whilst the second re-estimates the parameters of the potentials in a different way for MOPs and for MTEs.

4.1. Re-approximation by Dropping Pieces

A simple method of re-approximation is simply dropping pieces that are defined on lower-dimensional regions. Since there are no probabilities associated with such pieces, dropping these pieces causes no additional loss of accuracy. In some cases, we get pieces that have very low probabilities associated with them. In this case, we can drop such low probability

pieces. One should make sure that the total probability associated with the dropped pieces stays below some limit (e.g., 0.05) so that the loss of accuracy of computation is not much. It is necessary to re-normalize the potentials after dropping pieces with positive probabilities.

In Sec. 5, we describe some re-approximations of some intermediate functions obtained while solving a sample hybrid Bayesian network using MOPs and MTEs. Notice that this method cannot be used to lower the degree of a MOP potential, or the number of exponential terms in a MTE potential.

4.2. Re-approximation of MOPs using LIP with Chebyshev points

Another method for re-approximating MOP potentials is by using LIPs (Lagrange Interpolating Polynomials) with Chebyshev points.[12] To illustrate this, consider $g_6(\cdot)$, the marginal PDF of X_3 from the example in Section 3.3.2, which is defined on non-singleton intervals $(0, 6)$, $(6, 9)$, $(9, 12]$, $(12, 15)$, $(15, 18]$, $(18, 21)$, $(21, 24)$, $(24, 30)$. Let's re-approximate $g_6(\cdot)$ using 5 pieces as follows: $(0, 6)$, $[6, 12)$, $[12, 18)$, $[18, 24)$, $[24, 30)$. Currently, we do not have a theory for the choice of pieces. Our strategy is to use fewer pieces by merging pieces of the MOP being approximated, and to keep the sizes of the pieces as equal as possible.

For each interval, we compute the Chebyshev points starting with a number of points n initially set to a small number. A good starting choice is $n = 4$. We compute the 3-degree LIP that passes through these 4-points. We need to verify that the LIP is non-negative on the interval. If not, we increase n. If the function being approximated is strictly positive over the interval, then we are guaranteed to find an interpolating polynomial that is non-negative for some n. This is because when we increase the number of Chebyshev points by 1, the maximum error between the LIP and the polynomial being approximated is reduced by a factor of 2. If the smallest n that results in a polynomial that is non-negative is too high, we reduce the width of the interval (i.e., use more pieces) and re-start.

For approximating $g_6(\cdot)$, using $n = 5$ for all five pieces results in a MOP that is non-negative on all the five pieces. Next we normalize the resulting 5-piece, 4-degree MOP so the total area under the MOP is 1. Let $g_{6r}(\cdot)$ denote the 5-piece, 4-degree MOP approximation of g_6 found using the above procedure. The accuracy measures of $g_{6r}(\cdot)$ compared to $g_6(\cdot)$, of g_{6r} compared to f'_{X_3}, the exact PDF of X_3 truncated to $(0, 30)$, and of g_6 compared to f'_{X_3}, the exact PDF of X_3 truncated to $(0, 30)$ are shown in Table 4. The comparison suggests that g_{6r} approximates f'_{X_3} similarly to g_6, and we conclude that $g_{6r}(\cdot)$ is a good approximation of g_6.

The LIP method applies also for two or higher dimensional functions. There also exists Chebyshev points theory for two-dimensional regions.[19] For regions in 3 or higher dimensions, we can use some extensions of the

Table 4: Accuracy of the approximations of the marginal PDF of X_3 using the LIP method.

Approximations	KL	MAD	AEM	AEV
$g_{6r}(\cdot)$ vs. $g_6(\cdot)$	0.00004	0.00003	0.0	0.0013
g_{6r} vs. f'_{X_3}	0.00005	0.00004	0.0	0.0297
g_6 vs. f'_{X_3}	0.00003	0.00004	0.0	0.0284

one-dimensional Chebyshev point theory. One problem with using LIP with Chebyshev points in two or higher dimensions is non-negativity. It may be necessary to increase the degree of the MOP potential to a very high number making computations numerically unstable. In such cases, we can resort to the idea behind mixed-tree approximations, and use a one-dimensional LIP method with Chebyshev points to approximate a two or higher dimensional function. In Sec. 5, we describe a re-approximation of a MOP (using a one-dimensional LIP method with Chebyshev points) that is obtained as an intermediate function while solving a sample hybrid BN using MOPs defined on hypercubes.

4.3. Re-approximation of MTEs Using Numeric Least Squares

We discuss how to re-approximate MTEs by reducing the number of pieces and exponential terms. The idea behind the re-approximation method is similar to the one used for MOPs, but using a different mathematical tool. In this case, we rely on the Levenberg-Marquardt (LM) algorithm,[20,21] available via the command *FindFit* provided by *Mathematica*®. The LM algorithm is for minimizing the sum of the squares of the deviations of the fitted model from the exact one. It is controlled by a so-called *damping parameter*, λ, which is automatically adjusted in each iteration. If the reduction in the sum of squares is fast, λ is set to a small value and the LM algorithm becomes similar to the Gauss-Newton method. Otherwise, λ is set to a high value, and the LM algorithm becomes closer to a gradient descent method. Selecting appropriate starting values for the set of parameters to estimate is an important issue in order to avoid local optima. Since the MTE approximation to the Gaussian distribution is quite accurate, we will use the parameter estimates of this approximation as starting values for the LM algorithm. The steps to re-approximate an MTE density f for a given partition of the domain, and a fixed number of terms are detailed in Algorithm 1.

Note that this procedure is flexible. For example, if the shape of the function to re-approximate is not too irregular (for instance, if it has smooth parts), we can set a different number of terms in different regions, with fewer

NLS_re-approximation(f,n,$\mathcal{P}$)
Input: A univariate MTE f, an integer $n > 0$ and a partition $\mathcal{P}$ of the support of f.
Output: A re-approximation of f with n exponential terms in each element of $\mathcal{P}$.
begin
Let μ and σ be the mean and standard deviation obtained from f.
Let g_0 be an MTE approximation of a Normal density with parameters μ and σ.[22]
Let $S_1, \ldots, S_m$ be the elements of $\mathcal{P}$.
for $i \leftarrow 1$ **to** m **do**
Select a sample of equally distributed points $x_j \in S_i$.
Find the n-term MTE g_i defined on S_i that best fits the points $\{x_j, f(x_j)\}$ according to the LM algorithm.
end
Define $g(x)$ as

$$g(x) = g_i(x) \text{ if } x \in S_i, \quad i = 1, \ldots, n.$$

Normalize $g(x)$ so that

$$\int_{\cup_i S_i} g(x)\mathrm{d}x = \int_{\cup_i S_i} f(x)\mathrm{d}x$$

return g.
end

Algorithm 1: An algorithm for re-approximating MTEs using the LM method.

terms in the more uniform ones. Similar to MOPs, there is not yet a formal method to divide the domain. We followed the strategy of splitting the domain if the shape of the function differs from Gaussian shape. This is also related to a successful selection of the starting points, as it requires that the density in the piece being re-approximated somehow resembles a Gaussian shape, which is not always the case. Another possible strategy for splitting the domain, already explored in the literature,[23] consists of choosing points corresponding to extremes and inflection points. Such heuristic is appropriate when the number of exponential terms is low, but when the number of exponential terms is allowed to be higher, splitting by those points my be useless, as the MTE function is able to accurately represent changes in inflection and extremes.[13]

In the case of re-approximating 2 or higher dimensional potentials, we use the same procedure as MOPs, i.e. using mixed trees and fixing one dimension whilst re-approximating the function for the remaining dimensions. In the

case of MTEs this is not difficult, since this is the approach selected to define the conditional distributions (parent variables can only affect the partition of the domain, not the expression of the density itself).

Note that the re-approximation technique can lead to very accurate solutions, in which the complexity reduction is minimal, or to very smooth approximations, but with a big decrease in the complexity. It depends on the number of pieces selected and the number of exponential terms included (or degree of the polynomial in the MOP case). Finding a tradeoff between these two extremes is still an open question. This issue has been addressed from the point of view of learning, when a data sample is available,[24] proposing a heuristic based on finding the configuration that maximizes the Bayes information criterion (BIC) score of the model.

4.4. Experimental evaluation

In order to evaluate how the complexity of inference grows and how our proposed solution scales up, we have conducted a series of experiments over networks of increasing complexity as depicted in Figure 5. The experiments consisted of computing the marginal for Y_1 in network (a) and Y_2 in (b). The initial distributions in the networks are such that $X_1 \sim \mathcal{N}(0,1)$, $X_i|x_{i-1} \sim \mathcal{N}(x_{i-1},1)$, $i = 2,3$, $Y_1 = X_1 + X_2$ and $Y_2 = Y_1 + X_3$. Therefore, we have deterministic conditionals for Y_1, Y_2.

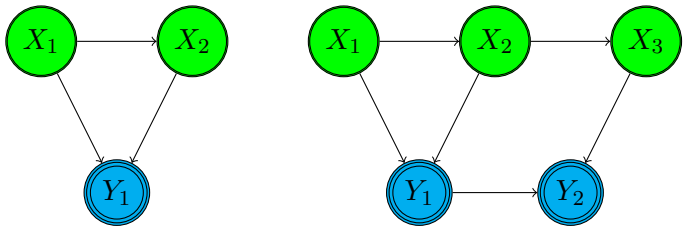

Figure 5: Networks used in the experimental evaluation.

We attempted to evaluate each network using the next three schemes:

- Hyper-rhombus MOPs. In this case, no approximations are carried out. Only pieces that correspond to singletons (and that, therefore, have null probability allocated) are removed.
- Hypercube MOPs with 3 pieces per variable. In this approach, when the hypercube is lost, re-approximation using LIPs with Chebyshev points is employed.

- Hypercube MTEs with 3 pieces per variable. Similarly, when the hypercube condition is lost, re-approximation using numerical least squares is applied.

Table 5: Results of the experiments carried out over the networks in Figure 5. The exact mean and variance for network (a) are $\mu = 0$, $\sigma^2 = 5.76$, while for (b) these values are $\mu = 0$, $\sigma^2 = 13.9995$. Note that Mathematica® is unable to compute the KL divergence for the estimated MTEs.

Net	Model	Domain	$\hat{\mu}$	$\hat{\sigma}^2$	KL	MAD	AEM	AEV	Time
(a)	MOP	Hyper-cube	0	5.44	0.073458	0.0806	0	0.3208	412.84
(a)	MOP	Hyper-rhombus	0	4.87	0.013378	0.01651	0	1.10387	47.47
(b)	MOP	Hyper-rhombus	0	13.65	0.00041	0.00069	0	0.348	672.44
(a)	MTE	Hyper-cube	-0.0008	5.1296	−−	0.1049	0.000822	0.638831	1181.82
(b)	MTE	Hyper-cube	-0.00158	16.07	−−	0.04883	0.00158	2.071	3273.65

The results of the experiments are shown in Table 5, where for each estimated model we have measured the mean and variance $(\hat{\mu}, \hat{\sigma}^2)$, maximum absolute deviation (MAD), absolute error in the mean (AEM), absolute error in the variance (AEV) and computing time. Note that, even though MOPs with hyper-rhombuses do not carry out approximations during inference, it underestimates the variance due to the fact that, in this experiment, we consider Gaussian variables, whose domain is infinite, while MOPs and MTEs have finite domain by definition.

In what concerns the growth of the complexity of inference, computing time of MOPs with hyper-rhombuses (i.e. with no approximation) is over 10 times higher in network (b) with respect to network (a). Note also that hyper-cube MOP for network (a) is much more costly (close to 10 times more) than MOP hyper-rhmobuses. It indicates that the time required to interpolate the approximate densities using LIP is not negligible. Regarding MTEs, the added complexity due to the inclusion of X_3 and Y_2 in network (b) causes the execution time to be close to 3 times higher than in network (a). Adding an extra pair of variables X_4, Y_3 makes the problem intractable with any of the methods.

5. Case Study: Solving a PERT Hybrid Bayesian Network

We will illustrate the inference process in hybrid BNs using a *stochastic PERT network.*[15] This problem has previously been addressed using BNs,[25] but the approach we propose here is more flexible and general, as it allows to compute the full distribution of the variables of interest, and not

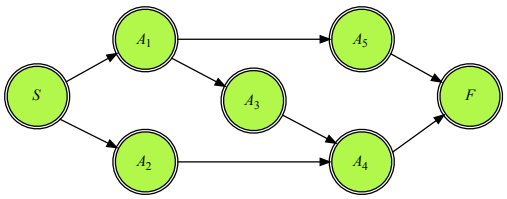

Figure 6: A PERT network with five activities

only their expectations and variances. PERT stands for *Program Evaluation and Review Technique*, and is one of the commonly used project management techniques.[26] PERT networks are directed acyclic networks where the nodes represent duration of activities and the arcs represent precedence constraints in the sense that before we can start any activity, all the parent activities have to be completed. The term *stochastic* refers to the fact that the duration of activities are modeled as continuous random variables. A previous proposal on using hybrid BNs for project scheduling under uncertainty is based on adapting the so-called *critical path method* to incorporate uncertainty, and representing the resulting model as a hybrid BN that is afterwards solved using dynamic discretization.[27] Our solution sidesteps the discretization problem by directly employing MTEs and MOPs.

Figure 6 shows a PERT network with 5 activities $A_1, \ldots, A_5$. Nodes S and F represent the start and finish times of the project. The links among activities mean that an activity cannot be started until after all its predecessors have been completed. Assume we are informed that the durations of A_1 and A_3 are positively correlated, and the same is true with A_2 and A_4. Then, this PERT network can be transformed into a BN as follows.

Let D_i and C_i denote the duration and the completion time of the activity i, respectively. The activity nodes in the PERT network are replaced with activity completion times in the BN. Next, activity durations are added

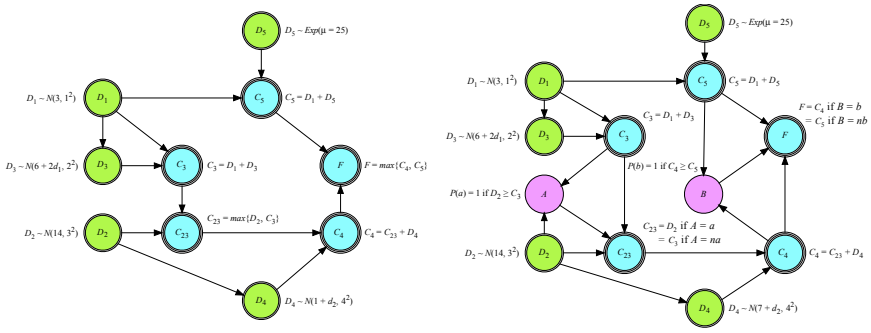

Figure 7: A BN (left) and a hybrid BN (right) representing the PERT network in Figure 6.

with a link from D_i to C_i, so that each activity will be represented by two nodes, its duration D_i and its completion time C_i. Notice that the completion times of the activities which do not have any predecessors will be the same as their durations. Hence, activities A_1 and A_2 will be represented just by their durations, D_1 and D_2. As A_3 and A_1 have positively correlated durations, a link will connect D_1 and D_3 in the BN. For the same reason, another link will connect D_2 and D_4. The completion time of A_3 is $C_3 = D_1 + D_3$. Let $C_{23} = \max\{D_2, C_3\}$ denote the completion time of activities A_2 and A_3. The completion time of activity A_5 is $C_5 = D_1 + D_5$, and for activity A_4, it is $C_4 = C_{23} + D_4$.

We assume that the project start time is zero and each activity is started as soon as all the preceding activities are completed. Accordingly, F represents the completion time of the project, which is the maximum of C_5 and C_4. The resulting PERT BN is given in Figure 7 (left).

Notice that the conditionals for the variables C_3, C_{23}, C_4, C_5 and F are deterministic, in the sense that their conditional distributions given their parents have zero variances. On the other hand, variables $D_1, \ldots, D_5$ are continuous random variables, and their corresponding conditional distributions are depicted next to their corresponding nodes in Figure 7. The parameters μ (mean) and σ^2 (variance) of the Normal distribution are in units of *days* and $days^2$, respectively. The parameter μ (mean) of the exponential distribution is in units of *days*.

Using the re-approximation techniques described in Section 4, we were able to solve the PERT hybrid Bayesian network using MOPs defined on

hypercubes and using MTEs. Using MOPs on hypercubes, evaluation of the entire Mathematica notebook (all computations including re-approximation) takes about 169.15 seconds (2.82 minutes). Using MTEs, evaluation of the entire Mathematica notebook (all computations including re-approximation) takes about 142.65 seconds (about 2.4 minutes). Details of all computations including re-approximations are given in the next sub-sections. Without re-approximations of some of the intermediate functions, we get a "lack of memory" warning message, even though the available memory was 16GBs. A reason for this is as follows. During the solution of the PERT hybrid Bayesian network problem, we get an intermediate function f_4 for $\{C_3, C_5\}$ that is a 23-piece, 8-degree MOP, and another intermediate function f_7 for $\{C_3, C_4\}$ that is a 17-piece, 7-degree MOP. In the process of marginalizing C_3, we have to multiply f_4 and f_7, and this results in a MOP that has up to 391 pieces and 17-degrees, and Mathematica runs out of memory during the marginalization process.

5.1. Solution Using Hypercube MOPs

We start with a 2-piece, 3-degree MOP $f_1(\cdot)$ as an approximation of the PDF of the standard normal. Using $f_1(\cdot)$, we define a 2-piece, 3-degree MOP approximation of the PDFs of D_1 and D_2. Using mixed trees, we define a 6-piece, 3-degree MOP of the conditional PDFs of D_3 and D_4. We used a 2-piece, 3-degree MOP approximation of the $Exp(25)$ density for D_5.

The initial potentials in the PERT hybrid BN in the right panel of Figure 7 are summarized in Table 6.

Table 6: Summary of the initial potentials in the PERT hybrid BN. Type CD means *conditional density* and DC stands for *deterministic conditional*.

Variable	Potential	Type	Variable	Potential	Type
D_1	f_{D_1}	CD	A	$p_{Aa}(d_2, c_3) = P(A = a \mid d_2, c_3)$ $p_{Ana}(d_2, c_3) = P(A = na \mid d_2, c_3)$	CD
D_2	f_{D_2}	CD	C_3	$f_{C_3}(d_1, d_3, c_3) = \delta(c_3 - d_1 - d_3)$	DC
D_3	f_{D_3}	CD	C_4	$f_{C_4}(c_{23}, c_4, d_4) = \delta(c_4 - c_{23} - d_4)$	DC
D_4	f_{D_4}	CD	C_5	$f_{C_5}(d_1, d_5, c_5) = \delta(c_5 - d_1 - d_5)$	DC
D_5	f_{D_5}	CD	C_{23}	$f_{C_{23a}}(d_2, c_3, c_{23}) = \delta(c_{23} - d_2)$ if $A = a$ $f_{C_{23na}}(d_2, c_3, c_{23}) = \delta(c_{23} - c_3)$ if $A = na$	DC
			F	$f_{Fb}(c_4, c_5, f) = \delta(f - c_4)$ if $B = b$ $f_{Fnb}(c_4, c_5, f) = \delta(f - c_5)$ if $B = nb$	DC
			B	$p_{Bb}(c_4, c_5) = P(B = b \mid c_4, c_5)$ $p_{Bnb}(c_4, c_5) = P(B = nb \mid c_4, c_5)$	CD

The goal is to compute the marginal density for F. To that end, we choose an elimination order of the remaining variables in the network, namely

$D_5, D_3, D_1, D_4, D_2, C_{23}, A, C_3, C_4, C_5$, and B. Different elimination orders can influence the complexity of the inference process. A discussion on the selection of the elimination order in hybrid BNs can be found in the related literature.[9]

Table 7: Potentials computed during the elimination of the variables in the PERT hybrid BN. The first column indicates the variables being eliminated, and the second contains the potentials obtained after eliminating them.

Variable	Potential computed
D_5	$f_2(d_1, c_5) = \int_{-\infty}^{\infty} f_{D_5}(d_5)\, f_{C_5}(d_1, d_5, c_5)\, \mathrm{d}d_5 = f_{D_5}(c_5 - d_1)$
D_3	$f_2(d_1, c_5) = \int_{-\infty}^{\infty} f_{D_5}(d_5)\, f_{C_5}(d_1, d_5, c_5)\, \mathrm{d}d_5 = f_{D_5}(c_5 - d_1)$
D_1	$f_4(c_3, c_5) = \int_{-\infty}^{\infty} f_{D_1}(d_1)\, f_{2c}(d_1, c_5)\, f_{3c}(d_1, c_3) \mathrm{d}d_1$
D_4	$f_5(c_{23}, d_2, c_4) = \int_{-\infty}^{\infty} f_{D_4}(d_2, d_4)\, f_{C_4}(c_{23}, c_4, d_4)\, \mathrm{d}d_4 = f_{D_4}(d_2, c_4 - c_{23})$
D_2	$f_{6a}(c_3, c_{23}, c_4) = \int_{-\infty}^{\infty} f_{D_2}(d_2)\, f_{C_{23a}}(d_2, c_3, c_{23})\, f_{5c}(c_{23}, d_2, c_4)\, p_{Aa}(d_2, c_3)\, \mathrm{d}d_2$ $= f_{D_2}(c_{23})\, f_{5c}(c_{23}, c_{23}, c_4)\, p_{Aa}(c_{23}, c_3)$ $f_{6na}(c_3, c_{23}, c_4) = f_{C_{23na}}(d_2, c_3, c_{23}) \int_{-\infty}^{\infty} f_{D_2}(d_2)\, f_{5c}(c_{23}, d_2, c_4)\, p_{Ana}(d_2, c_3)\, \mathrm{d}d_2$ $= \delta(c_{23} - c_3) \int_{-\infty}^{c_3} f_{D_2}(d_2)\, f_{5c}(c_{23}, d_2, c_4)\, \mathrm{d}d_2$
C_{23}	$f_{7a}(c_3, c_4) = \int_{-\infty}^{\infty} f_{6a}(c_3, c_{23}, c_4)\, \mathrm{d}c_{23}$ $= \int_{-\infty}^{\infty} f_{D_2}(c_{23})\, f_5(c_{23}, c_{23}, c_4)\, p_{Aa}(c_{23}, c_3)\, \mathrm{d}c_{23}$ $= \int_{c_3}^{\infty} f_{D_2}(c_{23})\, f_5(c_{23}, c_{23}, c_4)\, \mathrm{d}c_{23}$ $f_{7na}(c_3, c_4) = \int_{-\infty}^{\infty} f_{6na}(c_3, c_{23}, c_4)\, \mathrm{d}c_{23} = \int_{-\infty}^{c_3} f_{D_2}(d_2)\, f_5(c_3, d_2, c_4)\, \mathrm{d}d_2$
A	$f_7(c_3, c_4) = f_{7a}(c_3, c_4) + f_{7na}(c_3, c_4)$
C_3	$f_8(c_4, c_5) = \int_{-\infty}^{\infty} f_{4r}(c_3, c_5)\, f_{7r}(c_3, c_4)\, \mathrm{d}c_3$
C_4	$f_{9b}(c_5, f) = \int_{-\infty}^{\infty} p_{Bb}(c_4, c_5)\, f_{Fb}(c_4, c_5, f)\, f_8(c_4, c_5)\, \mathrm{d}c_4$ $= \int_{-\infty}^{\infty} p_{Bb}(c_4, c_5)\, \delta(f - c_4)\, f_8(c_4, c_5)\, \mathrm{d}c_4$ $= p_{Bb}(f, c_5)\, f_8(f, c_5)$ $f_{9nb}(c_5, f) = \int_{-\infty}^{\infty} f_{Fnb}(c_4, c_5, f)\, p_{Bnb}(c_4, c_5)\, f_8(c_4, c_5)\, \mathrm{d}c_4$ $= \delta(f - c_5) \int_{-\infty}^{c_5} f_8(c_4, c_5)\, \mathrm{d}c_4$
C_5	$f_{10b}(f) = \int_{-\infty}^{\infty} f_{9b}(c_5, f)\, \mathrm{d}c_5 = \int_{-\infty}^{f} f_8(f, c_5)\, \mathrm{d}c_5$ $f_{10nb}(f) = \int_{-\infty}^{\infty} f_{9nb}(c_5, f)\, \mathrm{d}c_5 = \int_{-\infty}^{f} f_8(c_4, f)\, \mathrm{d}c_4$
B	$f_{11}(f) = f_{10b}(f) + f_{10nb}(f)$

The potentials obtained after eliminating each variable are shown in Table 7. The deletion of D_5 is carried out by computing $f_2(d_1, c_5)$. Notice that f_2 is not defined on hypercubes. So we approximate f_2 by f_{2c} by using a 3-point mixed tree approximation. Next we delete D_3 by computing $f_3(d_1, c_3)$. Notice that f_3 is not defined on hypercubes. So we approximate f_3 by f_{3c} by using a 3-point mixed tree approximation.

Next we delete D_1 by computing $f_4(c_3, c_5)$. It is computed as a 23-piece, 8-degree MOP, and takes 5.05 seconds to be calculated. Next, we re-approximate f_4 by f_{4r}, as follows. 9 of the 23 pieces are defined on 1-dimensional regions. Thus, we can safely drop these pieces resulting in a 14-piece MOP. Further examination of these 14 pieces reveals that 6 of

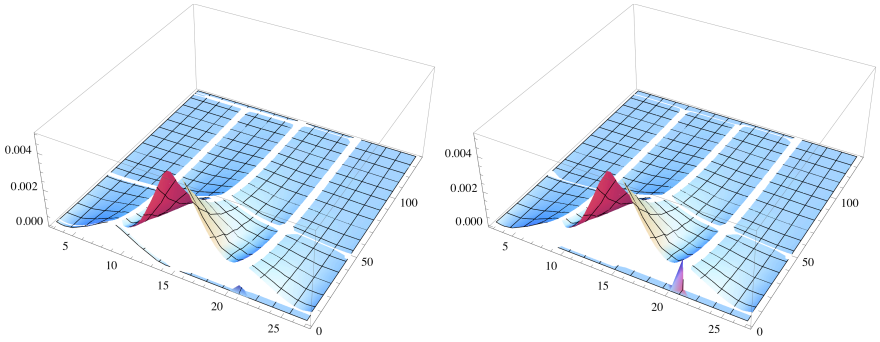

Figure 8: 3-D plots of f_4 (left) and f_{4r} (right).

the pieces have very small probabilities (0.000017 to 0.026) associated with them. The total probability associated with these 6 pieces is ≈ 0.044. After we drop these 6 pieces and re-normalize the potential, we obtain a 8-piece MOP $f_{4r}(c_3, c_5)$ that can be used in place of $f_4(c_3, c_5)$. Figure 8 shows 3-D plots of f_4 and f_{4r}.

Next, we delete D_4 by computing $f_5(c_{23}, d_2, c_4)$, which is not defined on hypercubes. As before, we re-define f_5 by f_{5c}, which is defined on a 3-point mixed tree hypercube.

Next, the deletion of D_2 yields the functions $f_{6a}(c_3, c_{23}, c_4)$ and $f_{6na}(c_3, c_{23}, c_4)$. Afterwards, we marginalize out C_{23} obtaining $f_{7a}(c_3, c_4)$ and $f_{7na}(c_3, c_4)$.

Next we delete A by computing $f_7(c_3, c_4)$. Potential f_7 is computed as a 17-piece, 7-degree MOP and it requires 16.82 seconds to be computed. Given the large number of pieces, which may lead to unmanageable functions after a new multiplication, we re-approximate f_7 by f_{7r}, an 8-piece, 7-degree MOP using LIPs with Chebyshev points as follows.

f_7 is computed on the domain $(3 < c_3 < 27) \times (1.125 < c_5 < 137.5)$. We will approximate this potential by a 8-piece, 7-degree MOP f_{7r} as follows. Consider a 4-way partition of $(3 < c_3 < 27)$ as follows: $(3, 11)$, $[11, 17)$, $[17, 23)$, and $[23, 27)$. Consider the region $3 < c_3 < 11$. We approximate $f_7(7, c_4)$ (a 1-dimensional function, $c_3 = 7$ is the mid-point of the interval) using LIP with Chebyshev points as discussed earlier by a 2-piece, 7-degree MOP and use this approximation for the region $(3 < c_3 < 11) \times (11 < c_5 < 59)$. By doing this for all four pieces of the partition of the domain of C_3, we obtain a 8-piece, 7-degree MOP approximation f_{7r} of the two-dimensional MOP f_7. Figure 9 shows 3-D plots of f_7 and f_{7r}.

Next, after elimination of C_3, $f_8(c_4, c_5)$ is computed as a 53-piece, 10-degree MOP in 26.01 seconds. The elimination of C_4 consists of computing $f_{9b}(c_5, f)$ and $f_{9nb}(c_5, f)$. The elimination of C_5 takes 108.19 seconds, required to calculate $f_{10b}(f)$ and $f_{10nb}(f)$.

Finally, by eliminating B, we obtain $f_{11}(f)$, which is a 22-piece, 11-

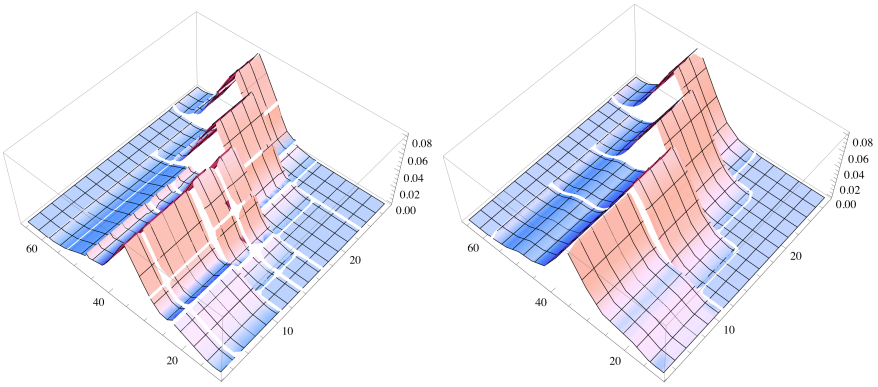

Figure 9: 3-D plots of f_7 (left) and f_{7r} (right).

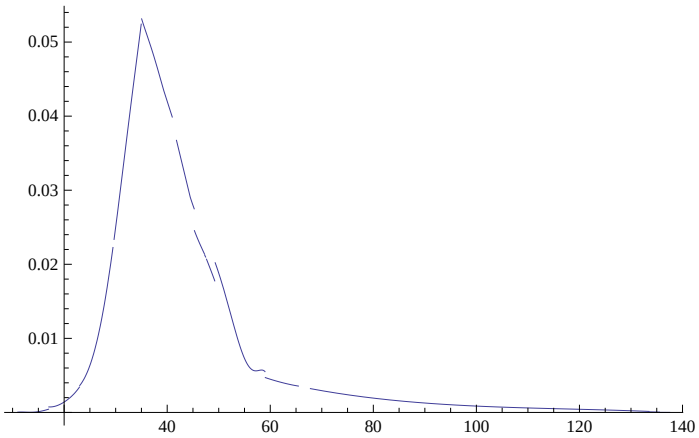

Figure 10: The MOP approximation f_{11} of the marginal of F.

degree MOP representation of the PDF of F. Using f_{11}, we compute the expectation and standard deviation of F: $\mu_F = 43.44$ days and $\sigma_F = 15.71$ days, respectively. A plot of f_{11} is shown in Figure 10. Evaluation of the entire Mathematica notebook (all computations including re-approximation) takes about 169.15 seconds (2.8 minutes).

5.2. Solution using MTEs

The inference process to solve the PERT network using MTEs is exactly the same as in the previous section using MOPs, but using MTE potentials instead. Therefore, we will only describe here the differences about the sizes of the potentials and the computations times, as well as the result obtained.

In Section 3.1, we explained how to avoid the problem of obtaining non-MTE potentials when doing the convolution operation. However, there is another problem when dealing with the *max* operations involved in the net-

Table 8: Parameter values of the MTE approximation to the Gaussian distribution.

a_i	73750.9	-235407	146016	146855	-140195	8980.02
b_i	-0.045451	-0.027365	-0.006084	-0.006084	0.013520	0.065411

work, in particular when computing f_{7a}, f_{7na}, f_{10b}, and f_{10nb}. In all those cases, because the probability potentials for discrete variables A and B are not defined on hypercubes, one of the integration limits is a variable, which yields a non-MTE potential. As an example illustrating this issue consider an MTE potential as follows:

$$f(x, y) = 1 + e^{3x+2y} \quad \text{if } 3 \leq x \leq 5,\ 1 \leq y \leq 5, \text{ and } x \geq y. \tag{13}$$

Integrating out y we obtain

$$\int_1^x f(x, y)dy = x - 1 + \frac{1}{2}e^{5x} - \frac{1}{2}e^{3x+2}, \quad \text{if } 3 \leq x \leq 5, \tag{14}$$

which is not an MTE function, as it contains a polynomial term. In this case, the problem is due to the constant term $a_0 = 1$ in $f(x, y)$, and it can be avoided by not including it in Definition 1 of an MTE function. Using this version of MTEs, the PERT network can be solved within the MTE class. Thus, instead of using the original definition, for solving the PERT network we use

$$f(\mathbf{z}) = \sum_{i=1}^{m} a_i \exp\left\{\mathbf{b}_i^{\mathsf{T}}\mathbf{z}\right\}, \tag{15}$$

in all the MTE potentials involved in the solution.

So, we start with a 1-piece, 6-terms (without a constant term) MTE, $f_1(\cdot)$, as an approximation of the PDF of the standard normal PDF on the domain $[-2.5, 2.5]$ (see the parameters in Table 8), obtained using the LM algorithm (see Section 4.3) and using as initial values the ones obtained by Langseth et al.[22] in their approximation of the PDF of the standard normal density. Using $f_1(\cdot)$, we define a 1-piece, 6-terms MTE approximation of the PDF of D_1 and D_2. Using mixed trees, we define a 2-piece, 6-terms MTE approximation of the conditional PDFs of D_3 and D_4. Since the PDF of the exponential distribution is already a 1-piece, 1-term MTE, no approximation is needed for the PDF of D_5.

In order to compute the marginal for F, we use the same elimination order as for the case of MOPs. After the elimination of D_5, D_3, and D_1, we compute a 11-piece, 6-terms MTE $f_4(c_3, c_5)$ in 4.54 seconds. We re-approximate f_4 as follows.

One of the 11 pieces is a singleton point, which due to the continuous nature of the distribution can be excluded. Three of the remaining pieces have very low probabilities (0.00005 to 0.019) with a total mass of 0.02594.

After dropping these pieces and re-normalizing, the density is a 4-piece MTE $f_{4s}(c_3, c_5)$ that can be used in place of $f_4(c_3, c_5)$. Figure 11 shows f_4 and f_{4s}.

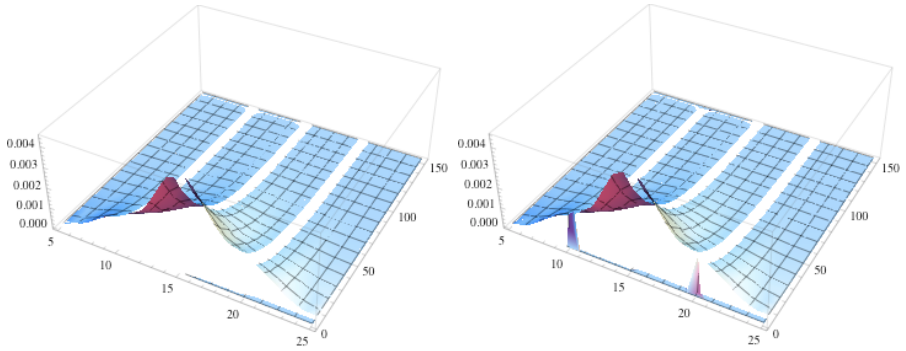

Figure 11: MTEs corresponding to f_4 (left) and f_{4s} (right).

After further elimination of D_4, D_2, C_{23} and A, we compute a 16-piece, 72-terms MTE $f_7(c_3, c_4)$ in 31.96 seconds, which we re-approximate as a 7-piece potential f_{7s}, with 6 terms for 5 of the pieces and 2 terms for the remaining 3 pieces.

After further elimination of C_3, we obtain a 20-piece, and between 4 and 10 terms MTE $f_8(c_4, c_5)$ in 29.21 seconds. Then we delete C_4, and C_5 (need 59.14 seconds), and B, obtaining a 5-piece, and between 2 and 38 terms MTE approximation of the marginal PDF of F. A graph of it is displayed in Figure 12. We compute the expectation and standard deviation: $E(F) = 43.79$ days and $\sigma_F = 16.17$ days, respectively. The evaluation of the entire Mathematica notebook (all computations) takes about 142.65 seconds (about 2.4 minutes).

5.3. Solution Using Simulation

In order to have a clearer idea of the true marginal for F, we computed an estimate of it by simulating a sample of size $n = 1,000,000$ of $D_1, \ldots, D_5$ using plain Monte Carlo simulation and then computing the corresponding values of C_3, C_{23}, C_4, C_5 and F for each record in the sample, according to their definition. Then, we fitted a Gaussian kernel density to the values obtained for F. The result is displayed in Figure 13. In this case, point estimates of the expectation and standard deviation of F are $\hat{E}(F) = 36.19$ and $\hat{\sigma}_F = 20.28$ days, respectively.

We have also computed confidence intervals for both parameters, using the statistic $\frac{\bar{X}-\mu}{s/\sqrt{n}} \sim t(n-1)$ for the mean and the statistic $\frac{(n-1)S^2}{\sigma^2} \sim \chi^2(n-1)$ for the standard deviation. The obtained 95% confidence interval

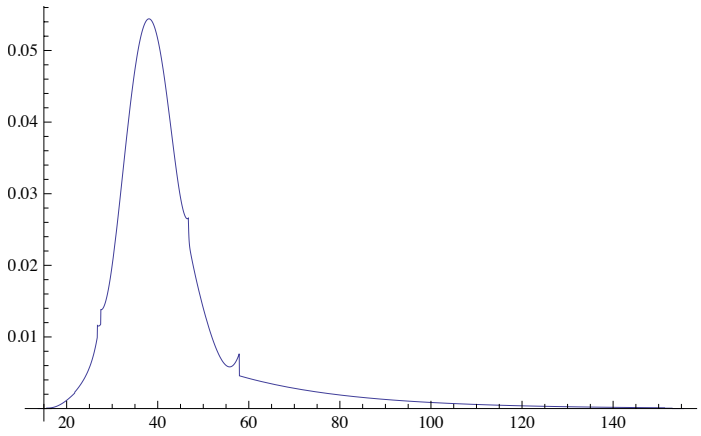

Figure 12: The MTE approximation of the marginal of F.

for the mean of F was $(36.15, 36.23)$, whilst for the standard deviation the result was $(20.25, 20.31)$.

Notice that we are able to compute the solution using Monte Carlo simulation just because in this example no observed variables are considered. If some of the variables were observed, then the Monte Carlo approximation may not converge. As an illustration, consider a piece of a Bayesian network with three continuous variables A, B and C, where $C = A + B$, i.e. C has a deterministic conditional, and A and B take values on the interval $(0, 1)$. Assume that C is observed to be equal to the value $C = 0.5$. In this scenario, it is likely that many of the configurations generated during the Monte Carlo sampling are finally discarded because they are incompatible with the observation. For instance, if we have sampled a value $A = 0.7$ then any value sampled for B will be incompatible with the observation $C = 0.5$, as $A + B$ will be greater than 0.5 with total certainty.

Unlike Monte Carlo approximations, the solution we propose in this paper is not affected by evidence.

6. Summary and Conclusions

We have described some practical issues in solving hybrid BNs that include deterministic conditionals using MTEs and MOPs. As an illustration, we have solved a PERT hybrid BN consisting of 2 discrete and 10 continuous variables, 5 of which have linear deterministic conditionals.

One key observation is that in the process of solving the PERT hybrid BN, some of the intermediate potentials have a large number of pieces, some of which are defined on lower dimensions and which have no useful information. One solution to this is to re-approximate these potentials with a smaller number of pieces and fewer degrees/terms. In the case of MOPs/MTEs, this can be done by dropping pieces on lower-dimensional regions (that have no probabilities) or pieces that have very small probabilities. In the case of MOPs, this can also be done using LIPs with Chebyshev points. In the case of MTEs, this can be done using the LM algorithm.

Some limitations of our methods are as follows. Currently, doing the re-approximations needs manual interventions to determine the number of pieces, the split points defining the pieces, and the number of terms in the case of MTEs. More work needs to be done in making these judgments.

We plan to solve the PERT hybrid BN using MOPs defined on hyper-rhombuses to keep the number of pieces to a minimum, and compare the running time and accuracy with the corresponding results using hypercubes. Shenoy[12] describes the use of LIPs with Chebyshev points for approximating univariate and bivariate functions by MOPs. However, more work needs to be done in re-approximating high-dimensional joint and conditional functions by MOP using LIPs with Chebyshev points.

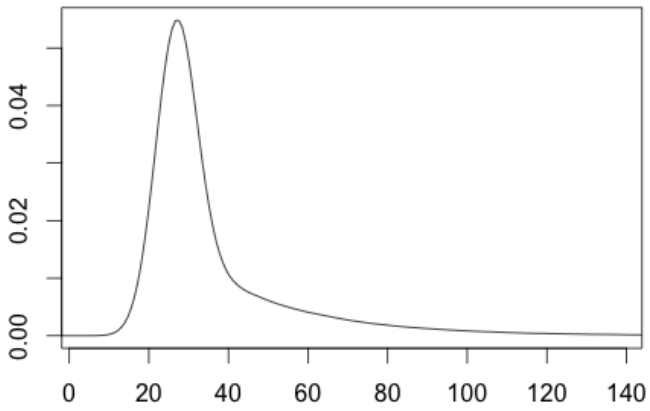

Figure 13: A Kernel approximation of the marginal PDF of F.

Acknowledgment

This work has been supported by the Spanish Ministry of Economy and Competitiveness, through projects TIN2010-20900-C04-01 and TIN2010-20900-C04-02, by Junta de Andalucía through project P11-TIC- 7821 and by ERDF (FEDER) funds, and by a sabbatical partially funded by the University of Kansas, Lawrence, KS, to one of the co-authors. Thanks to Serafín Moral for suggesting the re-approximation strategy. Some portions of this paper have appeared in the proceedings of the ISDA'2011 conference and PGM'2012 workshop.[28,29]

References

1. Lauritzen S. Propagation of probabilities, means and variances in mixed graphical association models, J Am Statist Assoc 1992; 87:1098–1108.

2. Lerner U, Parr R. Inference in hybrid networks: Theoretical limits and practical algorithms. In: Breese J, Koller D, editors. Uncertainty in Artificial Intelligence: Proceedings of the Seventeenth Conference. Morgan Kaufmann, San Francisco, CA; 2001. pp. 310–318.

3. Moral S, Rumí R, Salmerón A. Mixtures of truncated exponentials in hybrid Bayesian networks. ECSQARU'01. Lect Notes Artif Intell 2143. 2001. pp. 135–143.

4. Shenoy PP, Shafer G. Axioms for probability and belief function propagation. In: Shachter R, Levitt T, Lemmer J, Kanal L., editors. Uncertainty in Artificial Intelligence 4. North Holland, Amsterdam; 1990. pp. 169–198.

5. Shenoy PP, West JC. Extended Shenoy-Shafer architecture for inference in hybrid Bayesian networks with deterministic conditionals. Int J Approx Reason 2011;52:805–818.

6. Zhang N, Poole D. Exploiting causal independence in Bayesian network inference. J Artif Intell Res 1996;5:301–328.

7. Cobb BR, Shenoy PP, Rumí R. Approximating probability density functions with mixtures of truncated exponentials. Stat Comput 2006;16:293–308.

8. Neil M, Tailor M, Marquez D. Inference in hybrid Bayesian networks using dynamic discretization. Stat Comput 2007;17:219–233.

9. Rumí R, Salmerón A. Approximate probability propagation with mixtures of truncated exponentials. Int J Approx Reason 2007;45:191–210.

10. Langseth H, Nielsen TD, Rumí R, Salmerón A. Inference in hybrid Bayesian networks. Reliab Eng Syst Safe 2009;94:1499–1509.

11. Shenoy PP, West JC. Inference in hybrid Bayesian networks using mixtures of polynomials. Int J Approx Reason 2011;52:641–657.

12. Shenoy P. Two issues in using mixtures of polynomials for inference in hybrid Bayesian networks. Int J Approx Reason 2012;53:847–866.

13. Langseth H, Nielsen TD, Rumí R, Salmerón A. Mixtures of truncated basis functions. Int J Approx Reason 2012;53:212–227.

14. Marquez D, Neil M, Fenton N. Improved reliability modeling using Bayesian networks and dynamic discretization. Reliab Eng Syst Safe 2010;95:412–425.

15. Cinicioglu EN, Shenoy PP. Using mixtures of truncated exponentials for solving stochastic PERT networks. In: Kroupa T, Vejnarová J, editors. Proceedings of the 8th Workshop on Uncertainty Processing (WUPES-09). University of Economics, Prague. 2009. pp. 269–283.

16. Cobb BR, Rumí R, Salmerón A. Modeling conditional distributions of continuous variables in Bayesian networks. Lect Comp Sci 2005;3646:36–45.

17. Langseth H, Nielsen TD, Rumí R, Salmerón A. Maximum likelihood learning of conditional MTE distributions. ECSQARU'09. Lect Notes Artif Intell 5590 2009; pp. 240–251.

18. Moral S, Rumí R, Salmerón A. Approximating conditional MTE distributions by means of mixed trees. ECSQARU'03. Lect Notes Artif Intell 2711 2003; pp. 173–183.

19. Xu Y. Lagrange interpolation on Chebyshev points of two variables. J Approx Theory 1996;87:220–238.

20. Levenberg K. A method for the solution of certain non-linear problems in least squares. Q Appl Math 1944;2:164–168.

21. Marquardt D. An algorithm for least-squares estimation of nonlinear parameters, SIAM J Appl Math 1963;11:431–441.

22. Langseth H, Nielsen TD, Rumí R, Salmerón A. Parameter estimation and model selection for mixtures of truncated exponentials. Int J Approx Reason 2010;51:485–498.

23. Rumí R, Salmerón A, Moral S. Estimating mixtures of truncated exponentials in hybrid Bayesian network. Test 2006;15:397–421.

24. Langseth H, Nielsen TD, Pérez-Bernabé I, Salmerón A. Learning mixtures of truncated basis functions from data. Int J Approx Reason 2014;55:940–956.

25. Cho S. A linear Bayesian stochastic approximation to update project duration estimates. Eur J Oper Res 2009;196:585–593.

26. Malcolm DG, Roseboom JH, Clark CE, Fazar W. Application of a technique for research and development program evaluation. Oper Res 1959;7:646–669.

27. Khodakerami V, Fenton N, Neil M. Project scheduling: Improved approach to incorporating uncertainty using Bayesian networks. Proj Man J 2007;38:39–49.

28. Shenoy PP, Rumí R, Salmerón A. Some practical issues in inference in hybrid Bayesian networks with deterministic conditionals. In: Ventura S, Abraham A, Cios K, Romero C, Marcelloni F, Benitez JM, Gibaja E, editors. Proceedings of the 2011 Eleventh International Conference on Intelligent Systems Design and Applications. IEEE Research Publishing Services. Piscataway, NJ, 2011. pp. 605–610.

29. Rumí R, Salmerón A, Shenoy PP. Tractable inference in hybrid Bayesian networks with deterministic conditionals using re-approximations. In: Cano A, Gómez-Olmedo M, Nielsen TD, editors. Proceedings of the 6th European Workshop on Probabilistic Graphical Models. Granada, Spain, 2012. pp. 275–282.