Solving Stochastic PERT Networks Exactly Using Hybrid Bayesian Networks

Esma Nur Cinicioglu and **Prakash P. Shenoy**
School of Business
University of Kansas, Lawrence, KS 66045 USA
esmanur@ku.edu, pshenoy@ku.edu

Abstract

In this paper, we describe how a stochastic PERT network can be formulated as a Bayesian network. We approximate such PERT Bayesian network by mixtures of Gaussians hybrid Bayesian networks. Since there exists algorithms for solving mixtures of Gaussians hybrid Bayesian networks exactly, we can use these algorithms to make inferences in PERT Bayesian networks.

1 Introduction

Program Evaluation and Review Technique (PERT) was invented in 1958 for the POLARIS missile program by the Program Evaluation branch of the Special Projects Office of the U. S. Navy, assisted by consultants from Booz, Allen and Hamilton [13]. A parallel technique called *Critical Path Method* (CPM) was invented around the same time by Kelley and Walker [9]. Both PERT and CPM are project management techniques whose main goal is to manage the completion time of a large project consisting of many activities with precedence constraints, i.e., constraints that specify which other activities that need to be completed prior to starting an activity.

In PERT, a project is represented by a directed acyclic network where the nodes represent duration of activities and the arcs represent precedence constraints. In classical PERT, duration of activities are assumed to be known constants, and the task is to identify a "critical path" from start-time to finish-time such that the project completion time is the sum of the duration of the activities on the critical path. These activities are called *critical*, since a project could be delayed if these activities were not completed in the scheduled time. In stochastic PERT, activities are considered as random variables with probability distributions, and the main task is to compute the marginal probability distribution of the project completion time.

The problem of computing the marginal probability distribution of the project completion time is a difficult problem. Thus many approximate techniques have been developed. A classic solution proposed by Malcolm *et al*. [13] is to assume that all activities are independent random variables and that each activity has an approximate beta distribution parameterized by three parameters: mean time m, minimum (optimistic) completion time a, and maximum (pessimistic) completion time b. The expected duration of each activity is then approximated by $(a + 4m + b)/6$, and its variance is approximated by $(b - a)^2/36$. Using the expected duration times, the critical path is computed using the classical deterministic method. The mean and variance of the distribution of the project completion time is then approximated as the sum of the expected durations and the sum of variances of the activities on a critical path.

Another approximation is to assume that all activity durations are independent and having the Gaussian distribution [15]. The completion time of an activity i is given by $C_i = \text{Max}\{C_j \mid j \in \Pi(i)\} + D_i$, where C_j denotes the completion time of activity j, D_j denotes the duration of activity j, and $\Pi(i)$ denotes the parents (immediate predecessors) of activity i. The maximum of two independent Gaussian random variables is not Gaussian. However, the distribution of C_i is assumed to be Gaussian with the parameters estimated from the parameters of the parent activities. Depending on the values of the parameters, this assumption can lead to large errors.

Kulkarni and Adlakha [10] compute the distribution and moments of project completion time assuming that the activity durations are independent and having the exponential distribution with finite means. They call such stochastic PERT networks Markov networks.

If we don't assume independence of activity durations, the problem of computing the marginal distribution of the project completion time becomes computationally intractable for large projects. One solution to this problem is to use Monte Carlo techniques with variance reduction techniques to estimate the distribution of project completion time or its moments [2, 5, 7, 17, 18]. Another solution is to provide lower bounds for the expected project completion time [see e.g., 4, 6, and 14]. Elmaghraby [4] provides a review of Monte Carlo and bounding techniques.

Jenzarli [8] suggests the use of Bayesian networks to model the dependence between activity durations and completions in a project. However, such Bayesian networks are difficult to solve exactly since they may contain a mix of discrete and continuous random variables. One solution recommended by Jenzarli is to use Markov chain Monte Carlo techniques to estimate the marginal distribution of project completion time.

In this paper, we explore the use of exact inference in hybrid Bayesian networks using mixtures of Gaussians proposed by Shenoy [16] to compute the exact marginal distribution of project completion time. Activities durations can have any distribution, and may not be all independent. We model dependence between activities using a Bayesian network as suggested by Jenzarli [8]. We approximate non-Gaussian conditional distributions by mixtures of Gaussians, and we reduce the resulting hybrid Bayesian network to a mixture of Gaussian Bayesian networks. Such hybrid Bayesian networks can be solved exactly using the algorithm proposed by Lauritzen and Jensen [2001], which is implemented in Hugin, a commercially-available software package. We illustrate our approach using a small PERT network with five activities.

2 An Example of a PERT Bayes Net

Consider a PERT network as shown in Figure 5 with five activities, $A_1, \ldots, A_5$. S denotes project start time, and F denotes project completion time. The directed arrows in a PERT network denote precedence constraints. The precedence constraints are as follows. A_3 and A_5 can only be started after A_1 is completed, and A_4 can only be started after A_2 and A_3 are completed. The project is completed after all five activities are completed.

Using the technique described in Jenzarli [8], we will describe the dependencies of the activities by a Bayesian network. Let D_i denote the duration of activity i, and let C_i denote the earliest completion time of activity i. Let C_{23} denote earliest completion time of activities 2 and 3. Since our goal is to compute the marginal distribution of the earliest completion time of the project, we will assume that each activity will be started as soon as possible (after completion of all preceding activities). Also, we assume that $S = 0$ (with probability 1).

The interpretation of PERT networks as Bayes Nets allows us to depict the activity durations that are dependent on each other. For instance, in the current example durations of activities 1 and 3 and durations of activities 2 and 4 are positively correlated. Considering the dependencies between the activities, we convert the PERT network to a Bayes net following two basic steps. First activity durations are replaced with activity completion times, second activity durations are added with an arrow from D_i to C_i so that each activity is represented by two nodes. However, notice that the activities 1 and 2 are represented just by their durations, as D_1 and D_2. The reason for that is that they are starting activities and since they do not have any predecessors, the completion times of the activities will be the same as their durations.

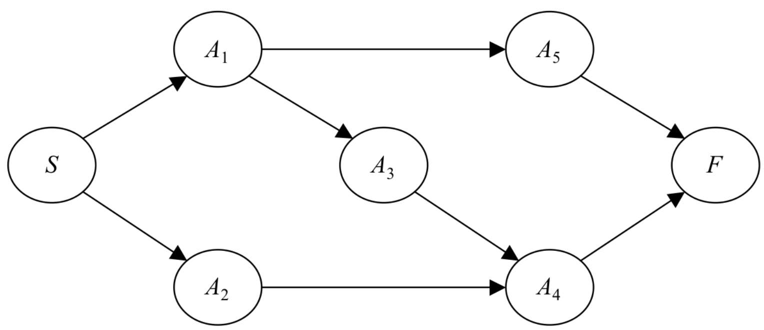

Figure 1: A stochastic PERT network with five activities.

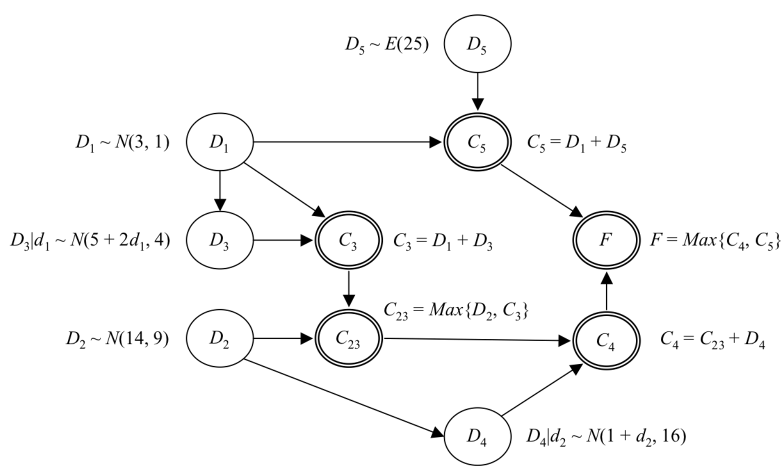

Figure 2: A Bayes net representation of the dependencies of the activities in the PERT network of Figure 1.

3. Mixtures of Gaussians Bayesian Networks

Mixtures of Gaussians (MoG) hybrid Bayesian networks were initially studied by Lauritzen [1992]. These are Bayesian networks with a mix of discrete and continuous variables. The discrete variables cannot have continuous parents, and all continuous variables have the so-called conditional linear Gaussian distributions. This means that the conditional distributions at each continuous node have to be Gaussian such that the mean is a linear function of its continuous parents, and the variance is a constant. MoG Bayesian networks have the property that for each instantiation of the discrete variables, the joint conditional distribution of the continuous variables is multivariate Gaussian. Hence the name 'mixtures of Gaussians.' An example of a MoG Bayesian network is as shown in Figure 3.

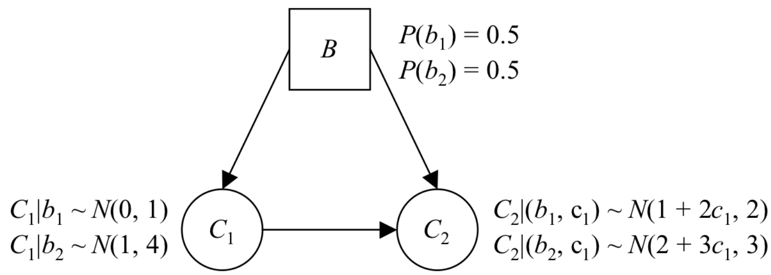

Figure 3: An example of a MoG Bayes net.

4 Converting a non-MoG Bayes Net to a MoG Bayes Net

Consider the Bayes Net shown in Figure 2. It is not a MoG Bayesian network since D_5 has a non-Gaussian distribution, and C_{23} and F have a non-linear conditional Gaussian distribution.

We will convert this Bayes net to a MoG Bayes net so that we can use the Lauritzen-Jensen algorithm to compute marginals of the MoG Bayes net. Before we do so, we first explain how we can approximate a non-Gaussian distribution by a MoG distribution, and how we can approximate a max deterministic function by a MoG distribution.

4.1 Non-Gaussian Distributions

In this subsection, we will describe how the exponential distribution $E[1]$ can be approximated by a MoG distribution.

Let A denote a chance variable that has the exponential distribution with mean 1, denoted by $E[1]$, and let f_A denote its probability density function (PDF). Thus

$$f_A(x) = e^{-x} \quad \text{if } 0 \le x$$
$$= 0 \quad \text{otherwise}$$

In approximating the PDF f_A by a mixture of Gaussians, we first need to decide on the number of Gaussian components needed for an acceptable approximation. In this particular problem, more the components used, better will be the approximation. However, more components will lead to a bigger computational load in making inferences. We will measure the goodness of an approximation by estimating the Kullback-Liebler divergence measure between the target distribution and the corresponding MoG distribution.

Suppose we use five components. Then we will approximate f_A by the mixture PDF $g_A = p_1\, \varphi_{\mu_1,\sigma_1} + \ldots + p_5\, \varphi_{\mu_5,\sigma_5}$, where φ_{μ_i,σ_i} denote the PDF of a uni-variate Gaussian distribution with mean μ_i and standard deviation $\sigma_i > 0$, $p_1, \ldots, p_5 \geq 0$, and $p_1 + \ldots + p_5 = 1$. To estimate the mixture PDF, we need to estimate fourteen free parameters, e.g., $p_1, \ldots, p_4, \mu_1, \ldots, \mu_5, \sigma_1, \ldots, \sigma_5$. To find the values of the 14 free parameters, we solve a non-linear optimization problem as follows:

Find $p_1, \ldots, p_4, \mu_1, \ldots, \mu_5, \sigma_1, \ldots, \sigma_5$, so as to minimize $\delta(f_A, g_A)$

subject to: $p_1 \geq 0, \ldots, p_4 \geq 0, p_1 + \ldots + p_4 \leq 1, \sigma_1 \geq 0, \ldots, \sigma_5 \geq 0$,

where $\delta(f_A, g_A)$ denotes a distance measure between two PDFs. A commonly used distance measure is Kullback-Leibler divergence δ_{KL} defined as follows:

$$\delta_{KL}(f_A, g_A) = \int_S f_A(x) ln\left(\frac{f_A(x)}{g_A(x)}\right) dx$$

In practice, we solve a discrete version of the non-linear optimization problem by discretizing both f_A and g_A using a large number of bins. To discretize g_A, we assume that the domain of φ_{μ_i,σ_i} extends only from $\mu_i - 3\sigma_i$ to $\mu_i + 3\sigma_i$. With probability greater than 0.99, the domain of $E[1]$ extends from [0, 4.6]. To match the domain of the $E[1]$ distribution, we constrain the values $\mu_i - 3\sigma_i \geq 0$ and $\mu_i + 3\sigma_i \leq 4.6$ for $i = 1, \ldots, 5$. Suppose we divide the domain into n equally sized bins. Let f_i and g_i denote the probability masses for the i^{th} bin corresponding to PDFs f_A and g_A, respectively Then the discrete version of the non-linear programming problem can be stated as follows:

Minimize $\sum_{i=1}^{n} f_i(x) ln\left(\frac{f_i(x)}{g_i(x)}\right)$

subject to: $p_1 \geq 0, \ldots, p_4 \geq 0, p_1 + \ldots + p_4 \leq 1$,

$\sigma_1 \geq 0, \ldots, \sigma_5 \geq 0$,

$\mu_1 - 3\sigma_1 \geq 0, \ldots, \mu_5 - 3\sigma_5 \geq 0$,

$\mu_1 + 3\sigma_1 \leq 4.6, \ldots, \mu_5 + 3\sigma_5 \leq 4.6$

One can use the solver in Excel to solve such optimization problems taking care to avoid local optimal solutions. An optimal solution computed in Excel with n = 100 (shown rounded to 3 digits) is shown in Table 1.

i	p_i	μ_i	σ_i
1	0.051	0.032	0.011
2	0.135	0.143	0.048
3	0.261	0.415	0.138
4	0.341	1.014	0.338
5	0.212	2.300	0.767

Table 1: Parameters of the MoG Approximation to the $E[1]$ distribution.

A graph of the two PDFs overlaid over each other is shown in Figure 4.

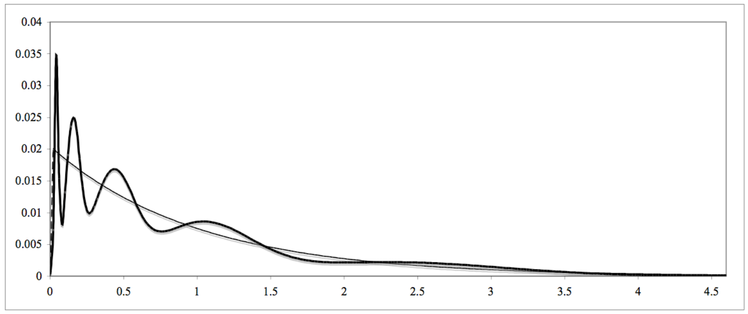

Figure 4: A 5-component MoG approximation (solid) of the $E[1]$ distribution (dashed).

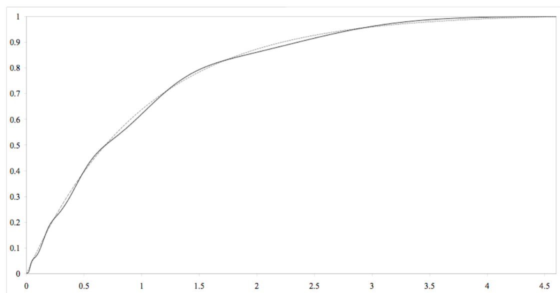

Figure 5: The CDFs of the $E[1]$ distribution (dashed), and its MoG approximation (solid).

To measure the goodness of the approximation, we can compute the Kullback-Leibler (KL) divergence of the two distributions over the domain [0, 4.6] where both densities are positive. The KL divergence is approximately 0.021. We can also compare moments. The mean and variance of the $E[1]$ distribution are 1 and 1. The mean and variance of the MoG approximation are 0.96 and 0.76. We can also compare the cumulative distribution functions (CDF). A graph of the two CDFs overlaid over each other is shown in Figure 5.

If we need to get a MoG approximation of the $E[\lambda]$ distribution, we can derive it easily from the MoG approximation of the $E[1]$ distribution. If $X \sim E[1]$, and $Y = \lambda X$, then $Y \sim E[\lambda]$. Thus, to get a MoG approximation of $E[25]$, e.g., the mixture weights p_i's don't change, but we need to multiply each mean μ_i and σ_i in Table 1 by 25.

4.2 Maximum of Two Random Variables

In this subsection, we will describe how a deterministic variable that is a maximum of two random variables can be approximated by a MoG Bayes net. Consider a Bayes net as shown in Figure 6. We will describe how this non-MoG Bayes net can be converted to a MoG Bayes net.

The Bayes net shown in Figure 6 has three random variables: $C_3 \sim N(14, 3)$, $D_2 \sim$ N(14, 9). D_2 and C_3 are independent, and C_{23} is conditionally deterministic. Since the deterministic function is not linear, the joint distribution of C_3, D_2, and C_{23} is not multivariate normal.

In approximating a Bayes net with MoG Bayes net, we will assume that the effective domain of a univariate Gaussian distribution with mean μ and standard deviation σ is (μ – 3σ, μ + 3σ). Thus, the domain of C_3 is (3.18, 24.82) and the domain of D_2 is (5, 23).

Our first step is to introduce a new discrete random variable B_{23} as shown in Figure 7. B_{23} has two states b_2 and b_3, and with C_3 and D_2 as parents. The conditional distributions of B_{23} are as follows: $P(b_2|c_3, d_2) = 1$ if $c_3 - d_2 \leq 0$, and $P(b_2|c_3, d_2) = 0$ if $c_3 - d_2 > 0$. Thus we can think of B_{23} as an indicator random variable which is in state b_2 when $c_3 \leq d_2$, and in state b_3 when $c_3 > d_2$. We make B_{23} a parent of C_{23}, and the conditional distributions of C_{23} can now be expressed as a conditional linear Gaussian as shown in Figure 7. Notice that the conditional distributions of all continuous variables are conditional linear Gaussians. The Bayes net is not a MoG Bayes net yet since B is a discrete random variable with continuous parents C_3 and D_2.

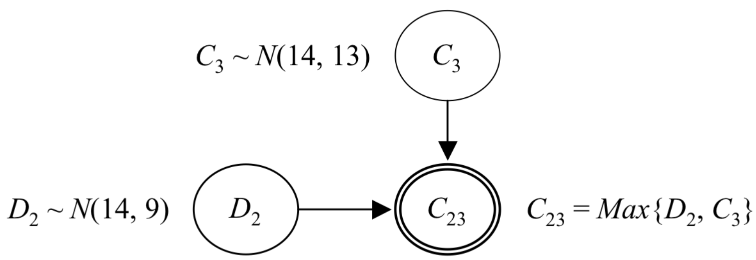

Figure 6: A Bayes net with a max deterministic variable

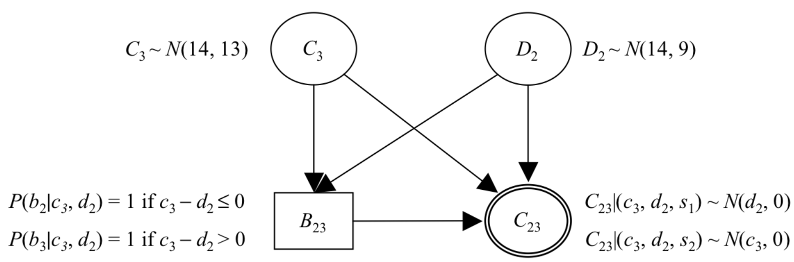

Figure 7: The augmented Bayes net with an additional discrete random variable B_{23}

Our next step is to do a sequence of arc reversals so that the resulting Bayes net is an equivalent MoG Bayes net. We need to reverse arcs (C_3, B_{23}) and (D_2, B_{23}) in either sequence. Suppose we reverse (C_3, B_{23}) first and then (D_2, B_{23}).

Since C_3 is continuous and B_{23} is discrete, we use the notation of mixed potentials introduced by Cobb and Shenoy [2006]. Let ε denote the mixed potential at C_3, and let β denote the mixed potential at B_{23}. Thus, $\varepsilon(c_3) = (1, \varphi_{14,\sqrt{13}}(c_3))$, where $\varphi_{\mu,\sigma}(.)$ denotes the probability density function of a univariate Gaussian distribution with mean μ and standard deviation σ, and β is as follows:

$$\beta(b_2, c_3, d_2) = (1, \iota) \quad \text{if } c_3 - d_2 \leq 0$$
$$= (0, \iota) \quad \text{if } c_3 - d_2 > 0$$

$\beta(b_3, c_3, d_2) = (0, \iota)$ if $c_3 - d_2 \le 0$
$= (1, \iota)$ if $c_3 - d_2 > 0$

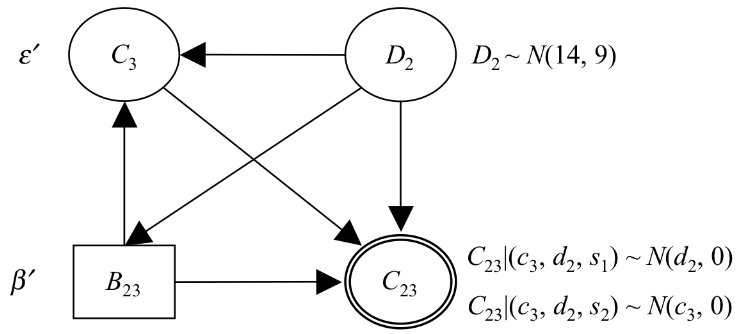

Figure 8: The Bayes net resulting from reversal of arc (C_3, B_{23})

Now reverse the arc (C_3, B_{23}), we first combine ε and β (obtaining ε⊗β), then compute the marginal of ε⊗β by removing C_3 (obtaining $(\varepsilon\otimes\beta)^{-C_3}$, and finally divide ε⊗β by $(\varepsilon\otimes\beta)^{-C_3}$ (obtaining $(\varepsilon\otimes\beta)/(\varepsilon\otimes\beta)^{-C_3}$). The potential $(\varepsilon\otimes\beta)^{-C_3}$ (= β′, say) is then associated with B_{23} and the potential $(\varepsilon\otimes\beta)/(\varepsilon\otimes\beta)^{-C_3}$ (= ε′, say) is associated with C_3 in the new Bayes net shown in Figure 8. The details of these potentials are as follows.

$$(\varepsilon\otimes\beta)(b_2, c_3, d_2) = (1, \varphi_{14,\sqrt{13}}(c_3)) \text{ if } c_3 - d_2 \le 0$$
$$= (0, \varphi_{14,\sqrt{13}}(c_3)) \text{ if } c_3 - d_2 > 0$$
$$(\varepsilon\otimes\beta)(b_3, c_3, d_2) = (0, \varphi_{14,\sqrt{13}}(c_3)) \text{ if } c_3 - d_2 \le 0$$
$$= (1, \varphi_{14,\sqrt{13}}(c_3)) \text{ if } c_3 - d_2 > 0$$

$$(\varepsilon\otimes\beta)^{-C_3}(b_2, d_2) = \beta'(b_2, d_2) = (\int_{-\infty}^{d_2} \varphi_{14,\sqrt{13}}(c_3)dc_3, \iota)$$

$$(\varepsilon\otimes\beta)^{-C_3}(b_3, d_2) = \beta'(b_3, d_2) = (\int_{d_2}^{\infty} \varphi_{14,\sqrt{13}}(c_3)dc_3, \iota)$$

Let $P(b_2|d_2)$ denote $\int_{-\infty}^{d_2} \varphi_{14,\sqrt{13}}(c_3)dc_3$. Notice that $\int_{d_2}^{\infty} \varphi_{14,\sqrt{13}}(c_3)dc_3 = 1 - P(b_2|d_2)$. The details of ε′ are as follows.

$$\varepsilon'(b_2, c_3, d_2) = (1/P(b_2|d_2), \varphi_{14,\sqrt{13}}(c_3)) \text{ if } c_3 - d_2 \le 0$$
$$= (0, \varphi_{14,\sqrt{13}}(c_3)) \text{ if } c_3 - d_2 > 0$$
$$\varepsilon'(b_3, c_3, d_2) = (0, \varphi_{14,\sqrt{13}}(c_3)) \text{ if } c_3 - d_2 \le 0$$
$$= (1/(1 - P(b_2|d_2)), \varphi_{14,\sqrt{13}}(c_3)) \text{ if } c_3 - d_2 > 0$$

Notice that the conditional probability densities of C_3 (given B_{23} and D_2) are no longer conditional linear Gaussians. Later (after we are done with arc reversals), we will approximate these conditional distributions by mixtures of Gaussians.

Next we need to reverse (D_2, B_{23}). Suppose the potential at D_2 is denoted by δ. Details of the arc reversal are as follows:

$\delta(d_2) = (1, \varphi_{14,3}(d_2))$, $\beta'(b_2, d_2) = (P(b_2|d_2), \iota)$,
$\beta'(b_3, d_2) = (1 - P(b_2|d_2), \iota)$
$(\delta\otimes\beta')(d_2, b_2,) = (P(b_2|d_2), \varphi_{14,3}(d_2)) = (1, P(b_2|d_2)\ \varphi_{14,3}(d_2))$
$(\delta\otimes\beta')(d_2, b_3) = (1-P(b_2|d_2), \varphi_{14,3}(d_2)) = (1, (1 - P(b_2|d_2)\ \varphi_{14,3}(d_2))$

$$(\delta\otimes\beta')^{-D_2}(b_2) = \beta''(b_2) = (\int_{-\infty}^{\infty} P(b_2|d_2)\varphi_{14,3}(d_2)dd_2, \iota) = (0.5, \iota)$$

$$(\delta\otimes\beta')^{-D_2}(b_3) = \beta''(b_3) = (\int_{-\infty}^{\infty} (1 - P(b_2|d_2)\varphi_{14,3}(d_2)dd_2, \iota) = (0.5, \iota)$$

$(\delta\otimes\beta')/(\delta\otimes\beta')^{-D_2}(d_2, b_2) = \delta'(d_2, b_2) = (1, 2P(b_2|d_2)\ \varphi_{14,3}(d_2))$
$(\delta\otimes\beta')/(\delta\otimes\beta')^{-D_2}(d_2, b_3) = \delta'(d_2, b_3) = (1, 2(1 - P(b_2|d_2))\ \varphi_{14,3}(d_2))$

Notice that the potential δ' represents conditional probability densities for D_2 given b_2 and b_3. Clearly, these are not conditional linear Gaussians. The revised Bayes net is shown in Figure 9.

Notice that the Bayes net in Figure 9 is almost a MoG BN except for the fact that the potentials ε' and δ' are not conditional linear Gaussians. We can approximate these potentials by mixtures of Gaussian potentials using the optimization technique described in Shenoy [2006].

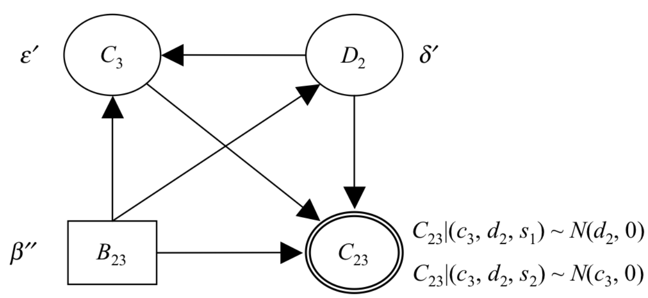

Figure 9: The Bayes net after reversal of arc (D_2, B_{23})

First, we need to approximate δ', the conditional density functions $\delta'(d_2, b_2) = (1, 2P(b_2|d_2)\varphi_{14,3}(d_2))$ and $\delta'(d_2, b_3) = (1, 2(1 - P(b_2|d_2))\varphi_{14,3}(d_2))$ by mixtures of Gaussians. For that, first we need to decide how many Gaussian components are needed for an acceptable approximation. Increasing the number of components used, might improve the approximation however with more components in use the computational load will also increase. Keeping that in mind three Gaussian distributions will be used for the approximation of δ'. That means we will approximate δ' by the mixture pdf $g = p_1\ \varphi_{\mu_1, \sigma_1} + p_2\ \varphi_{\mu_2, \sigma_2} + p_3\ \varphi_{\mu_3, \sigma_3}$, where $\varphi_{\mu_i, \sigma_i}$ denote the pdf of a uni-variate Gaussian distribution with mean μ_i and standard deviation $\sigma_i > 0$, $p_1, \ldots, p_3 \geq 0$, and $p_1 + p_2 + p_3 = 1$. For the estimation of each conditional density function eight free parameters, e.g., $p_1, p_2, \mu_1, \ldots, \mu_3, \sigma_1, \ldots, \sigma_3$ will be used which will be estimated through the solution of the following non-linear optimization problem.

Find $p_1, p_2, \mu_1, \ldots, \mu_3, \sigma_1, \ldots, \sigma_3$ so as to minimize $\delta_{KL}(\delta', g)$
Subject to: $p_1 \geq 0, p_2 \geq 0, p_1 + p_2 \leq 1, \sigma_1 \geq 0, \ldots, \sigma_3 \geq 0$.

In order to solve the problem we are going to discretize both δ' and g using a large number of bins. Let δ'_i and g_i denote the probability masses for the i^{th} bin corresponding the PDFs δ' and g, respectively. Then the discrete version of the nonlinear programming problem can be stated as follows:

$$\text{Minimize} \sum_{i=1}^{n} \delta'_i \, ln(\delta'_i/g_i)$$

Subject to: $p_1 \geq 0, p_2 \geq 0, p_1 + p_2 \leq 1, \sigma_1 \geq 0, \ldots, \sigma_3 \geq 0$.

For this optimization problem, we are going to use the solver in Excel. Notice that, in the solution of the problem we should take care to avoid local optimal solutions. The optimal solutions computed in Excel are as follows:

$$2\, P(b_2|d_2)\, \varphi_{14,\,3}(d_2) \approx 0.484\, \varphi_{14.820,\,2.352} + 0.083\, \varphi_{15.647,\,2.393} + 0.433\, \varphi_{16.298,\,2.631}$$

$$2(1 - P(b_2|d_2))\, \varphi_{14,\,3}(d_2) \approx 0.433\, \varphi_{11.702,\,2.631} + 0.083\, \varphi_{13.180,\,2.393} + 0.484\, \varphi_{12.186,\,2.352}$$

A graph of the conditional density $f_{b_2|d_2}$ overlaid with the MoG approximation is shown in Figure 10.

We have completed approximating the mixed Bayesian network by a mixture of Gaussians (MoG) Bayesian network. The original Bayesian network and its MoG approximation are shown in Figure 9.

5 Converting the PERT Bayes net to a MoG Bayes Net

Notice that the PERT Bayes net in Figure 2 is not a MoG BN. The variables C_{23} and F are conditionally deterministic variables that are not linear, and D_5 has a non-Gaussian distribution. Using the techniques described in 4.1 and 4.2 above, these variables need to be approximated by MoG distributions. For that reason, as shown in top left panel of Figure 11, we add three discrete variables, B_5, B_F, and B_{23}, with the appropriate discrete distributions.

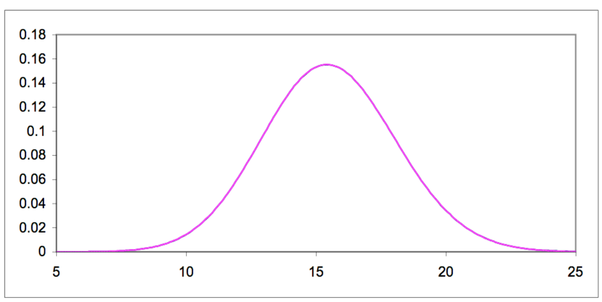

Figure 10: A graph of the conditional density $f_{d_2|b_2}$ overlaid on its MoG approximation

As mentioned before, in MoG Bayes nets, discrete nodes cannot have continuous parents. B_{23} has two continuous parents C_3 and D_2, and B_F has two continuous parents C_4 and C_5. We use arc reversals to address this situation.

First, notice that C_3 is a conditionally deterministic variable, i.e., there are no conditional density functions for C_3. So, before we can reverse arc

(C_3, B_{23}), we need to reverse (C_3, D_3) first. The BN that results from reversing arc (D_3, C_3) is shown in the top right panel of Figure 11. Notice that the natures of the nodes D_3 and C_3 have changed. C_3 is conditionally non-deterministic and D_3 is conditionally deterministic.

Now we reverse arc (C_3, B_{23}). When an arc between two nodes is reversed, the nodes inherit the each others parents. So B_{23} inherits D_1 as its parent, and C_3 inherits D_2 as its parent. The resulting BN is shown in bottom left panel of Figure 11.

Next, we reverse arc (D_2, B_{23}). After reversing this arc, D_2 inherits D_1 as its parent. The resulting BN is shown in the bottom right panel of Figure 11.

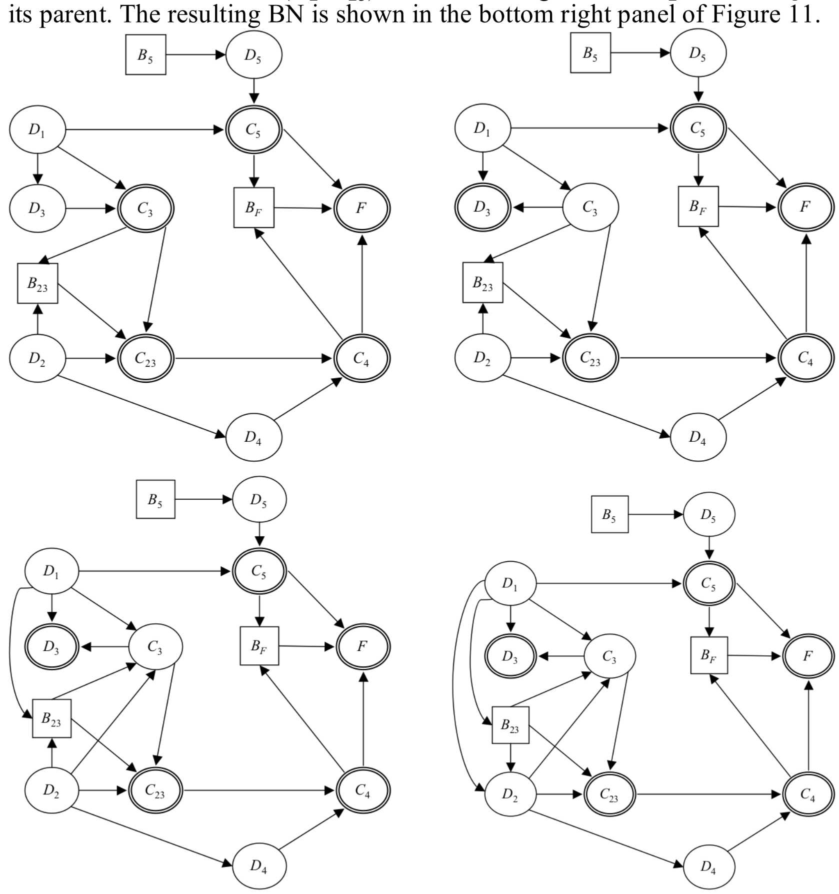

Figure 11: *Top Left*: PERT BN after adding discrete nodes. *Top Right*: After reversal of arc (D_3, C_3). *Bottom Left*: After reversal of arc (C_3, B_{23}). *Bottom Right*: After reversal of arc (D_2, B_{23}).

Next, we reverse arc (D_1, B_{23}). D_1 and B_{23} do not have any other parents, and consequently, no additional arcs will be added after reversing (D_1, B_{23}). The resulting BN is shown in the top left panel of Figure 12.

Next, B_F has continuous predecessors C_5, C_4, and C_{23}, all of which are conditionally deterministic. We will address this situation next. First, we reverse (D_5, C_5). C_5 inherits B_5 as its parent, and D_5 inherits D_1 as its parent.

The resulting BN is shown in the top right panel of Figure 12. Notice that the natures of the nodes D_5 and C_5 have interchanged. Second, we reverse arc (D_4, C_4). D_4 inherits C_{23}, and C_4 inherits D_2 as their parents. The resulting BN is shown in the bottom left panel of Figure 12. Notice that the natures of the nodes D_4 and C_4 have interchanged. Third, we reverse arc (C_3, C_{23}). C_{23} inherits D_1 as its parent. The resulting BN is shown in the bottom right panel of Figure 12. Notice that the natures of the nodes C_3 and C_{23} have interchanged. All the predecessors of B_F are now conditionally non-deterministic, and we can proceed to reverse the corresponding arcs in sequence.

Next, we reverse arc (C_4, B_F). B_F inherits D_2 and C_{23} as its parents, and C_4 inherits C_5 as its parent. The resulting BN is shown in the top left panel of Figure 13.

Next, we reverse arc (C_{23}, B_F). B_F inherits D_1 and B_{23} as its parents, and C_{23} inherits C_5 as its parent. The resulting BN is shown in the top right panel of Figure 13.

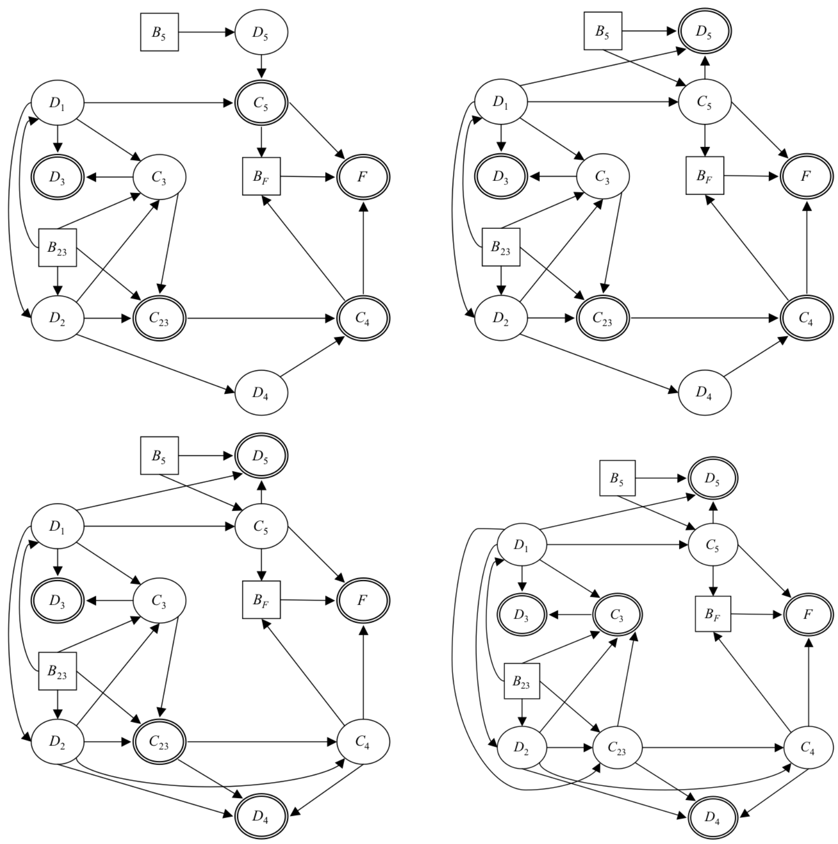

Figure 12: *Top Left*: After reversal of arc (D_1, B_{23}). *Top Right*: After reversal of arc (D_5, C_5). *Bottom Left*: After reversal of arc (D_4, C_4). *Bottom Right*: After reversal of arc (C_3, C_{23}).

Next, we reverse arc (D_2, B_F). D_2 inherits C_5 as its parent. The resulting BN is shown in the bottom left panel of Figure 13.

Next, we reverse arc (C_5, B_F). C_5 inherits B_{23} as its parent, and B_F inherits B_5 as its parent. The resulting BN is shown in the bottom right panel of Figure 13.

Finally, we reverse arc (D_1, B_F). D_1 inherits B_5 as its parent. The resulting BN is shown in Figure 14.

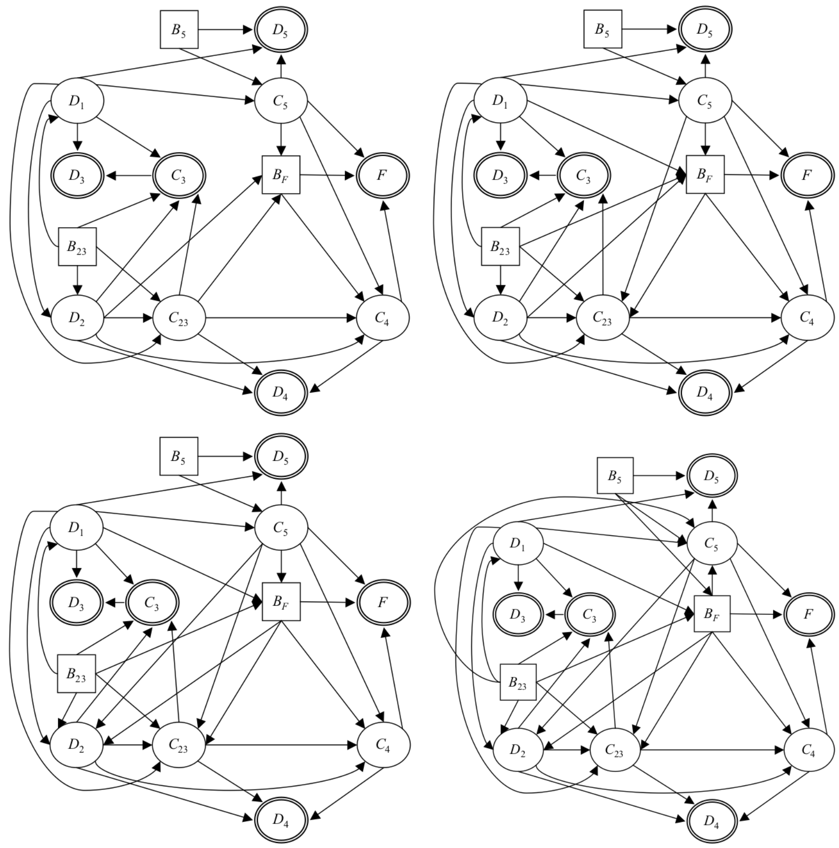

Figure 13: *Top Left*: After reversal of arc (C_4, B_F). *Top Right*: After reversal of arc (C_{23}, B_F). *Bottom Left*: After reversal of arc (D_2, B_F). *Bottom Right*: After reversal of arc (C_5, B_F).

We are now done with arc reversals. The discrete nodes B_5, B_{23}, and B_F, do not have continuous parents. In the process of arc reversals, the potentials at some of the continuous nodes are no longer conditional linear Gaussian. We can address this as discussed in sub-section 4.1. At the end of this process, the hybrid Bayes net is now a MoG Bayes net, and we can proceed to make inferences using the Lauritzen-Jensen algorithm.

6 Summary and Conclusions

The goal of this paper has been to make inferences in PERT Bayes nets. Our main strategy is to approximate a PERT Bayes net by a MoG Bayes net, and then use the Lauritzen-Jensen algorithm to make exact inferences in the MoG Bayes net. The Lauritzen-Jensen algorithm is implemented in Hugin, a commercially available software, and thus our strategy can be used in practice.

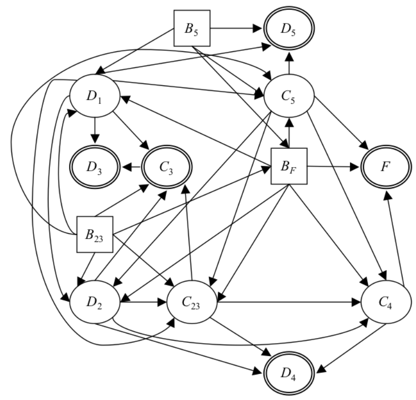

Figure 14: After reversal of arc (D_1, B_F).

Some disadvantages of our strategy are as follows. In the process of arc reversals, we increase the domains of the potentials. This increases the complexity of inference. In our toy example, the complexity is still manageable. For large examples, the resulting complexity may make this strategy impractical. Another disadvantage is arc reversals make Gaussian distributions non-Gaussian. We can approximate non-Gaussian distributions by mixtures of Gaussians. However, when the non-Gaussian distribution has many continuous variables in its domain, the task of finding a mixture of Gaussians approximation by solving an optimization problem can be difficult in practice.

References

[1] Anklesaria, K. P. and Z. Drezner (1986), "A multivariate approach to estimating the completion times for PERT networks," *Journal of the Operational Research Society*, **37**, 811–815.

[2] Burt, J. M., and M. B. Garman (1971), "Conditional Monte Carlo: A simulation technique for stochastic network analysis," *Management Science*, **18**, 207–217.

[3] Elmaghraby, S. E. (1967), "On the expected duration of PERT type networks," *Management Science*, **13**, 299–306.

[4] Elmaghraby, S. E. (1977), *Activity Networks: Project Planning and Control by Network Models*, John Wiley and Sons, NY.

[5] Fishman, G. S. (1985), "Estimating critical path and arc probabilities in stochastic activity networks," *Naval Research Logistics Quarterly*, **32**, 249–261.

[6] Fulkerson, D. R. (1962), "Expected critical path lengths in PERT networks," *Operations Research*, **10**, 808–817.

[7] Garman, M. B. (1972), "More on conditional sampling in the simulation of stochastic networks," *Management Science*, **19**, 90–95.

[8] Jenzarli, A. (1995), "Modeling dependence in project management," PhD dissertation, University of Kansas School of Business, Lawrence, KS.

[9] Kelley, J. E. and M. R. Walker (1959), "Critical path planning and scheduling," in *Proceedings of the Eastern Joint Computer Conference*, pp. 160–173, Institute of Radio Engineers, NY.

[10] Kulkarni, V. G. and V. G. Adlakha (1986), "Markov and Markov-regenerative PERT networks," *Operations Research*, **34**(5), 769–781.

[11] Lauritzen, S. L. (1992), "Propagation of probabilities, means and variances in mixed graphical association models," *Journal of American Statistical Association*, **87**(420), 1098–1108.

[12] Lauritzen, S. L. and F. Jensen (2001), "Stable local computation with conditional Gaussian distributions," *Statistics and Computing*, **11**, 191–203.

[13] Malcolm, D. G., J. H. Roseboom, C. E. Clark, and W. Fazar (1959), "Application of a technique for research and development program evaluation," *Operations Research*, **7**, 646–669.

[14] Robillard, P. (1976), "Expected completion time in PERT networks," *Operations Research*, **24**, 177–182.

[15] Sculli, D. (1983), "The completion time of PERT networks," *Journal of the Operational Research Society*, **34**(2), 155–158.

[16] Shenoy, P. P. (2006), "Exact inference in hybrid Bayesian networks using mixtures of Gaussians," in R. Dechter, and T. Richardson (eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Twenty-Second Conference* (UAI-06), 2006, pp. 428–436, AUAI Press, Corvallis, OR.

[17] Sigal, C. E, A. A. B. Pritsker, and J. J. Solberg (1979), The use of cutsets in Monte-Carlo analysis of stochastic networks," *Mathematics and Computers in Simulation*, **21**, 376–384.

[18] Van Slyke, R. M. (1963), "Monte Carlo methods and the PERT problem," *Operations Research*, **11**, 839–860.