

# A Comparison on the Effectiveness of Two Heuristics for Importance Sampling

Changhe Yuan and Marek J. Druzdzel

Decision Systems Laboratory

School of Information Sciences and Intelligent Systems Program

University of Pittsburgh, Pittsburgh, PA 15260

{cyuan,marek}@sis.pitt.edu

## Abstract

Given a good *importance function*, *importance sampling* is able to achieve satisfactory precisions within a reasonable time. In addition to the well known requirement that the importance function should have a similar shape to the target density (Rubinstein, 1981), it is also highly recommended that the importance function possess *heavy tails* (Geweke, 1989; MacKay, 1998; Yuan and Druzdzel, 2004). To achieve this, the  $\epsilon$ -*cutoff* heuristic (Cheng and Druzdzel, 2000; Yuan and Druzdzel, 2003) was used to cut off extremely small probabilities in the importance function (Yuan and Druzdzel, 2003). However,  $\epsilon$ -cutoff demonstrates inconsistent performance on different networks. In this paper, we analyze the underlying reasons and propose another heuristic, *if-tempering*, based on *simulated tempering*. We test the new heuristic on three large real Bayesian networks and observe that if-tempering consistently helps the EPIS-BN algorithm (Yuan and Druzdzel, 2003) achieve better precisions than  $\epsilon$ -cutoff.

## 1 Introduction

*Importance sampling* is used in many aspects of modern statistics and econometrics to approximate unsolvable integrals. It has also become the basis for several state of the art Monte Carlo sampling-based inference algorithms for Bayesian networks, for which inference is known to be NP-hard (Cooper, 1990; Dagum and Luby, 1993). The accuracy of importance sampling is very sensitive to the form of the *importance function*. A good importance function can lead importance sampling to yield excellent results in a reasonable time. It has been shown that, in addition to the well known requirement that the importance function should have a similar shape to the target density (Rubinstein, 1981), it is also highly recommended that the importance function possess heavy tails (Geweke, 1989; MacKay, 1998; Yuan and Druzdzel, 2004). To this end, the  $\epsilon$ -*cutoff* heuristic (Cheng and Druzdzel, 2000; Yuan and Druzdzel, 2003) was used to cut off extremely small proba-

bilities in the importance function (Yuan and Druzdzel, 2003). Overall, the heuristic does help in achieving better precisions. However, it also demonstrates inconsistent performance on different networks. In this paper, we analyze the underlying reasons for this instability and propose to use another heuristic, *if-tempering*, based on *simulated tempering*. The if-tempering heuristic tempers an importance function to a degree that depends on a rough estimation on the performance of the importance function. We test the new heuristic on three large real Bayesian networks and observe that if-tempering consistently helps the EPIS-BN algorithm to achieve better precisions than  $\epsilon$ -cutoff.

The outline of the paper is as follows. Section 2 and Section 3 provide a brief review of most existing importance sampling algorithms for Bayesian networks, including the EPIS-BN algorithm (Yuan and Druzdzel, 2003). Section 4 proposes the new if-tempering heuristic.

First, we review the main results in (Yuan and Druzdzel, 2004) about why heavy tails are desirable for importance functions. Second, we discuss the  $\epsilon$ -cutoff heuristic (Cheng and Druzdzel, 2000; Yuan and Druzdzel, 2003) and its drawback. Last, we propose the if-tempering heuristic. Section 5 presents the results of experimental tests of the if-tempering heuristic on three large real Bayesian networks.

## 2 Importance Sampling Algorithms for Bayesian Networks

Importance sampling has become the basis for several state of the art Monte Carlo sampling-based inference algorithms for Bayesian networks. These algorithms inherit the characteristic that their accuracy largely depends on the quality of the importance functions. The farther the importance function is from the target distribution, the more samples importance sampling needs to converge. Since the theoretical convergence rate is in the order of  $\frac{1}{\sqrt{m}}$ , where  $m$  is the number of samples, for essentially all Monte Carlo methods, the number of samples needed increases at least at a quadratic speed. Hence, given limited sources, any effort of making the importance function closer to the target distribution will directly influence the precision and efficiency of importance sampling. Based on the different methods that they use to get importance function, we classify some existing importance sampling algorithms for Bayesian networks into three families. The first family uses the prior distribution of a Bayesian network as the importance function, such as *Probabilistic logic sampling* (Henrion, 1988) and *likelihood weighting* (Fung and Chang, 1989; Shachter and Peot, 1989). The second family resorts to learning methods to learn an importance function, such as *Self-importance sampling* (SIS) (Shachter and Peot, 1989), *adaptive importance sampling* (Ortiz and Kaelbling, 2000), and AIS-BN (Cheng and Druzdzel, 2000). The third family directly computes an importance function in the light of both the prior distribution and the evidence, such as The *backward sampling* (Fung and del

Favero, 1994), IS (Hernandez et al., 1998), *annealed importance sampling* (Neal, 1998), and EPIS-BN algorithms.

## 3 The EPIS-BN Algorithm

Based on the observation that *loopy belief propagation* (LBP) provides surprisingly good results for many networks with loops (Murphy et al., 1999), Yuan and Druzdzel propose to use LBP to compute the importance function in the EPIS-BN algorithm (Yuan and Druzdzel, 2003). The importance function in the EPIS-BN algorithm is defined as:

$$\rho(\mathbf{X} \setminus \mathbf{E}) = \prod_{i=1}^n P(X_i | PA(X_i), \mathbf{E}), \quad (1)$$

where each  $P(X_i | PA(X_i), \mathbf{E})$  is an *importance conditional probability table* (ICPT) (Cheng and Druzdzel, 2000). The following theorem shows that we can calculate the ICPTs exactly (Yuan and Druzdzel, 2003).

**Theorem 1** *Let  $X_i$  be a variable in a polytree, and  $\mathbf{E}$  be the set of evidence. The exact ICPT  $P(X_i | PA(X_i), E)$  for  $X_i$  is*

$$\alpha(PA(X_i))P(X_i | PA(X_i))\lambda(X_i), \quad (2)$$

where  $\alpha(PA(X_i))$  is a normalizing constant dependent on  $PA(X_i)$ , and  $\lambda(X_i)$  is the message to  $X_i$  sent from its descendants.

In networks with loops, getting the exact  $\lambda$  messages for all variables is equivalent to performing an exact inference. Since our goal is to obtain a good and not necessarily the optimal importance function, we can accept good approximations of the  $\lambda$  messages. Given the surprisingly good performance of LBP, we believe that it can also provide us with good approximations of the  $\lambda$  messages. After applying LBP to calculating an importance function, the EPIS-BN algorithm also uses the  $\epsilon$ -cutoff heuristic to modify the function in order to possess heavy tails. We will discuss the  $\epsilon$ -cutoff heuristic in Section 4.2 in more detail.

Experimental results in (Yuan and Druzdzel, 2003) shows that the EPIS-BN algorithm achieves a considerable improvement over the

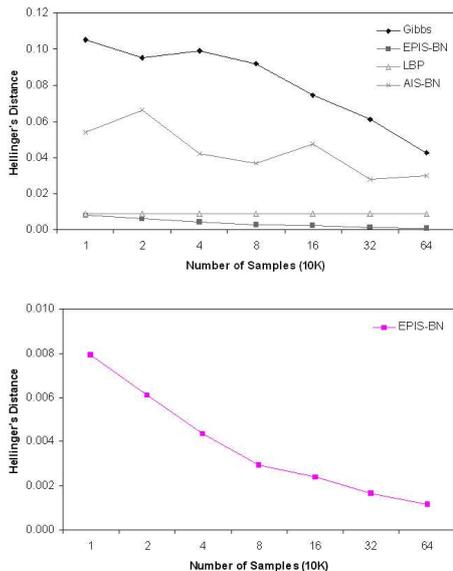


Figure 1: Convergence rates of the Gibbs sampling, AIS-BN, LBP, and EPIS-BN algorithms on the ANDES network. The bottom plot show important fragments of the top plot on a finer scale.

state of the art algorithm, the AIS-BN algorithm (Cheng and Druzdzel, 2000), which in turn has been shown to achieve precisions orders of magnitude better than likelihood weighting and SIS. Figure 1 shows a typical plot of the convergence rates of several inference algorithms on the ANDES network. Furthermore, the results in (Yuan and Druzdzel, 2003) also show that the EPIS-BN algorithm in some cases already approaches the limit that sampling algorithms can do, because the precisions that it achieves on some networks are already in the same order as those of probabilistic logic sampling on the same networks without evidence; in the latter case, since there is no evidence in the networks, logic sampling samples from the optimal importance function, the prior distribution. We believe that precision so achieved is the limit of sampling algorithms.

## 4 Heuristics for Generating Heavy-Tail Importance Functions

The last two sections review and classify the main Monte Carlo sampling-based inference algorithms for Bayesian networks based on their methods of obtaining importance functions. Most of these algorithms look for importance functions that have shapes close to the target density. They neglect that it is also highly recommended that importance functions possess heavy tails (Geweke, 1989; MacKay, 1998; Yuan and Druzdzel, 2004). In this section, we first review the main results in (Yuan and Druzdzel, 2004) about why heavy tails are desirable, then discuss the  $\epsilon$ -cutoff heuristic and its drawback, and finally propose another heuristic, the if-tempering heuristic.

### 4.1 Why Heavy Tails?

Let  $f$  be the joint probability distribution of a Bayesian network. Druzdzel (Druzdzel, 1994) shows that  $f$  follows the lognormal distribution. Therefore, we can look at any importance sampling algorithm for Bayesian networks as using one lognormal distribution as the importance function to compute the expectation of another lognormal distribution. Let  $f(X)$  be the original density of a Bayesian network and let  $f(\ln X) \propto N(\mu, \sigma_0^2)$ . We assume that we cannot sample from  $f(X)$  but we can only evaluate it at any point. Let the importance function be  $g(X)$ , which satisfies  $g(\ln X) \propto N(\mu', \sigma_1^2)$ . We obtain the variance of the importance sampling estimator as

$$Var_{g(X)}(w(X)) = \frac{(\frac{\sigma_1}{\sigma_0})^2}{\sqrt{2(\frac{\sigma_1}{\sigma_0})^2 - 1}} e^{\frac{(\frac{\mu' - \mu}{\sigma_0})^2}{2(\frac{\sigma_1}{\sigma_0})^2 - 1}} - 1, \quad (3)$$

where  $w(X) = \frac{f(X)}{g(X)}$ . The necessary condition for the variance in Equation 3 to exist is that  $2(\frac{\sigma_1}{\sigma_0})^2 - 1 > 0$ , which means that the variance of  $g(\ln X)$  should at least be greater than half of the variance of  $f(\ln X)$ . Note that  $|\frac{\mu' - \mu}{\sigma_0}|$  can be looked on as the standardized distance between  $\mu'$  and  $\mu$  with regard to  $f(\ln X)$ . For different values of  $|\frac{\mu' - \mu}{\sigma_0}|$ , we plot the variance against

$\frac{\sigma_1}{\sigma_0}$  in Figure 2. We observe that as the tails of the importance function become lighter, the variance increases rapidly and suddenly goes to infinity. However, when the tails become heavier, the variance increases slowly. Therefore, we want to avoid light tails and err on the heavy tail side in order to be safe.

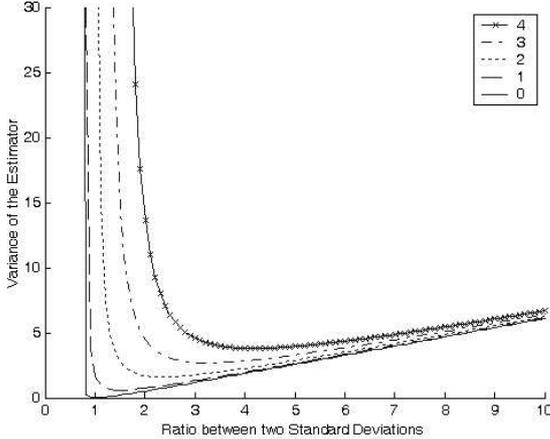
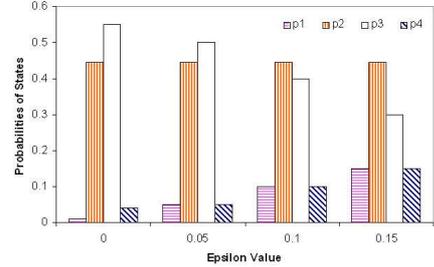


Figure 2: A plot of  $\frac{\sigma_1}{\sigma_0}$  against the variance when using the importance function  $g(\ln X) \propto N(\mu', \sigma_1^2)$  with different  $\mu'$ 's to integrate the density  $f(\ln X) \propto N(\mu, \sigma_0^2)$ . The legend shows different values of  $|\frac{\mu' - \mu}{\sigma_0}|$ .

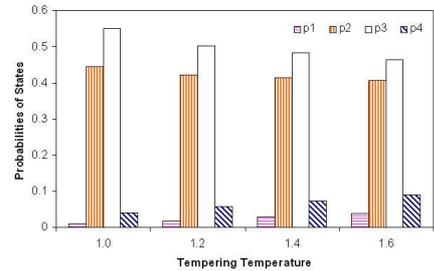
## 4.2 The $\epsilon$ -cutoff heuristic

Because of the importance of heavy tails, we apply the  $\epsilon$ -cutoff heuristic (Cheng and Druzdzal, 2000; Yuan and Druzdzal, 2003) to modify the importance function as in Equation 1 in the EPIS-BN algorithm (Yuan and Druzdzal, 2003). The main idea of the  $\epsilon$ -cutoff heuristic is setting a threshold  $\epsilon$  and replacing any smaller probability in the importance function by  $\epsilon$ . At the same time, we compensate for this change by subtracting it from the largest probability in the same conditional probability distribution. After we apply LBP to calculate an importance function in EPIS-BN, the importance function will not be a precise estimate of the target density and is likely to possess light tails. The tails in the context of Bayesian networks can be defined as the states with extremely small probabilities and extremely large

probabilities, which actually locate in the tails of the approximate lognormal distributions of the Bayesian networks. Therefore, arguably, the  $\epsilon$ -cutoff heuristic makes the tails heavier.



(a1)  $\epsilon$ -cutoff



(a2) if-tempering

Figure 3: The effect of applying  $\epsilon$ -cutoff and if-tempering to a discrete distribution of four states with different  $\epsilon$  values and temperatures.

However, the  $\epsilon$ -cutoff shows inconsistent behavior for different networks (Yuan and Druzdzal, 2003): It makes the results better in some networks but worse for others. After we examine  $\epsilon$ -cutoff more carefully, we believe that there are two main reasons for this inconsistency. First, regardless of the relative value of the small probabilities,  $\epsilon$ -cutoff cuts them using a uniform standard defined by a single  $\epsilon$ . However, light tails are defined relative to the target distribution. Its criteria for different parts of the importance function should definitely be different. Furthermore,  $\epsilon$ -cutoff only subtracts the discrepancy from the largest probability in the same conditional probability distribution. We believe that the negative consequence of this strategy is that it changes the shape of the importance function a lot. The left plot in Figure 3 is such an example. In the example, we apply

$\epsilon$ -cutoff with different  $\epsilon$  values. The resulting shapes become much different from the original importance function. However, as we said in the beginning, the shape of an importance function matters a lot: it should be close to the target density. After we apply LBP to calculating an importance function, we believe the importance function is already close to the target density. We need to take a gentler approach to obtain heavy tails.

Second,  $\epsilon$ -cutoff sets the  $\epsilon$  value based on the structure of the importance function. More specifically, it sets different  $\epsilon$  values for nodes with different number of states. The larger the number, the smaller the  $\epsilon$ . However, we believe a more appropriate strategy is to take into account the performance of an importance function.

### 4.3 The if-tempering Heuristic

The analysis in the previous section delivers the message that abrupt interference with an importance function may make its shape change a lot in an unknown direction. We want to largely keep its original shape and take gentler modifications. To this end, we propose the *if-tempering* heuristic based on simulated tempering. If the original importance function is  $I(X)$ , the if-tempering heuristic proposes to use the following tempered importance function

$$I'(X) \propto I(X)^{1/T}, \quad (4)$$

where  $T$  ( $T > 1$ ) is the tempering temperature. The right plot in Figure 3 shows the results when we apply if-tempering to the same distribution as in the left plot. The figure shows that by tempering the distribution, we keep the shape of the original distribution and also achieve the goal of making states with small probabilities more likely and states with large probabilities less likely. The states with moderate probabilities are left almost intact. These properties successfully avoid the main drawback of  $\epsilon$ -cutoff that we discussed in the last section.

One problem of applying the if-tempering heuristic to EPIS-BN algorithm is that tempering and normalizing the importance function

in Equation 1 is itself an NP-hard problem. To avoid this difficulty, we temper and normalize each ICPT separately. The tempered importance function thus will be

$$I'(X) \propto \prod_{i=1}^n P^{1/T}(X_i | \text{PA}(X_i), \mathbf{E}). \quad (5)$$

Choosing temperature is not an easy task. If the temperature is too low, we will not be able to obtain heavy tails; if the temperature is too high, the tails of the importance function may become too heavy. The optimal tempering temperature  $T$  will be different for different networks and problem cases. We recommend to select  $T$  based on the *coefficient of variation* of the unnormalized weights. Suppose that we have drawn  $m$  independent samples from the importance function  $I(\theta)$ ; then, the coefficient of variation is defined as

$$cv^2(w) = \frac{\sum_{j=1}^m (w^{(j)} - \bar{w})^2}{(m-1)\bar{w}^2}, \quad (6)$$

where  $\bar{w}$  is the sample average of the  $w^{(j)}$ .  $cv^2(w)$  is a good indicator how close the shape of an importance function is to the target density. Small  $cv^2(w)$  means that the shape of the importance function is close to the target distribution. In such circumstances, if-tempering with higher temperatures has more power in correcting light tails. On the contrary, large  $cv^2(w)$  indicates that there is a mismatch between the shapes of the importance function and the target distribution. It is better to use lower temperatures for if-tempering, because tempering the importance function too much only makes it worse. We recommend some criteria in Section 5.1 based on some experimental results.

Notice that there is a close relation between if-tempering and annealed importance sampling (Neal, 1998). In order to draw samples more freely from a possibly isolated sample space, the annealed importance sampling algorithm anneals each sample using Markov chains defined by a sequence of distributions defined as

$$f_i(x) = f_0(x)^{\beta_i} f_n(x)^{1-\beta_i}, \quad (7)$$

where  $1 = \beta_0 > \beta_1 > \dots > \beta_n = 0$ ,  $f_n(x)$  is the importance function, and  $f_0(x)$  is the target density.  $f_n(x)$  is usually a distribution flatter than the target density. The main drawback of this algorithm is that we need to draw many samples in order to get a single sample. In our approach, since we already get a good importance function by applying LBP, we propose to use only a single tempered importance function.

## 5 Experimental Results

To compare the effectiveness of the  $\epsilon$ -cutoff and if-tempering heuristics, we compare the performances of three algorithms: the EPIS-BN algorithm without  $\epsilon$ -cutoff (EP), the EPIS-BN algorithm with  $\epsilon$ -cutoff (EPIS), and the EP algorithm with if-tempering heuristic (EP+A). We calculate their departure from the exact solutions, which we obtain using the clustering algorithm (Lauritzen and Spiegelhalter, 1988). The distance metric that we use is Hellinger’s distance (Kokolakis and Nanopoulos, 2001). We applied these three algorithms to three large real Bayesian networks: ANDES (Conati et al., 1997), CPCS (Pradhan et al., 1994), and PATHFINDER (Heckerman, 1990). We implemented our algorithm in C++ and performed our tests on a 2.5 GHz Pentium IV Windows XP computer with 1 GB memory.

### 5.1 Parameter Selection

We did some experiments to choose the tempering temperature based on coefficient of variation. In the experiments, we randomly selected 20 evidence nodes for each network. After we use LBP to calculate the importance function, we apply the if-tempering heuristic with different temperatures and generated 320K samples. The tempering temperature of course depends on different networks. Based on the analysis of the results, we recommend to use the following parameters.

$$T = \begin{cases} 1.05, & \text{if } cv^2(w) > 5.0 ; \\ 1.10, & \text{if } 1.0 < cv^2(w) \leq 5.0 ; \\ 1.17, & \text{otherwise .} \end{cases}$$

For the  $\epsilon$ -cutoff heuristic in EPIS, we use the default settings:  $\epsilon = 0.006$  for nodes with the number of outcomes fewer than 5,  $\epsilon = 0.001$  for nodes with the number of outcomes between 5 and 8, and otherwise  $\epsilon = 0.0005$ . These parameters have already been tuned to optimize the performance of  $\epsilon$ -cutoff.

### 5.2 Results of Batch Experiments

We generated a total of 75 test cases for each of the three networks. These cases consisted of five sequences of 15 cases each. For each sequence, we randomly chose a different number of evidence nodes: 15, 20, 25, 30, 35 respectively. The evidence nodes were chosen from a predefined list of potential evidence nodes. The prior probability of evidence was extremely small: between  $10^{-4}$  and  $10^{-20}$  in ANDES, between  $10^{-8}$  and  $10^{-38}$  in CPCS, between  $10^{-4}$  and  $10^{-38}$  in PATHFINDER, and with the average around  $10^{-16}$ . We believe that these cases represent difficult real problems.

For each of the test cases, we ran the EP, EPIS, and EP+A algorithms for 320K samples. Figure 4 shows the box plots of the results. We also plot the results against the probability of the evidence in Figure 5. The results show that EP+A performed consistently better than EP and EPIS for all three networks. Although EPIS yielded better results for CPCS, it generates worse results for ANDES. Furthermore, we can see from Figure 5 that  $\epsilon$ -cutoff heuristic brings many oscillations to the results of PATHFINDER. The results of paired one-tailed t-test for the results are shown in Table 1. We can see that the improvement of EP+A over EP and EPIS are significant.

	E vs EC	E vs EA	EC vs EA
A	$1.2 \times 10^{-4}$	$4.6 \times 10^{-6}$	$6.6 \times 10^{-7}$
C	$4.3 \times 10^{-5}$	$2.0 \times 10^{-10}$	$3.0 \times 10^{-6}$
P	$5.7 \times 10^{-4}$	$7.7 \times 10^{-10}$	$4.3 \times 10^{-5}$

Table 1: Results of paired one-tail t-test for comparisons between the EP (E), EPIS (EC), and EP+A (EA) algorithms for ANDES (A), CPCS (C), and PATHFINDER (P).

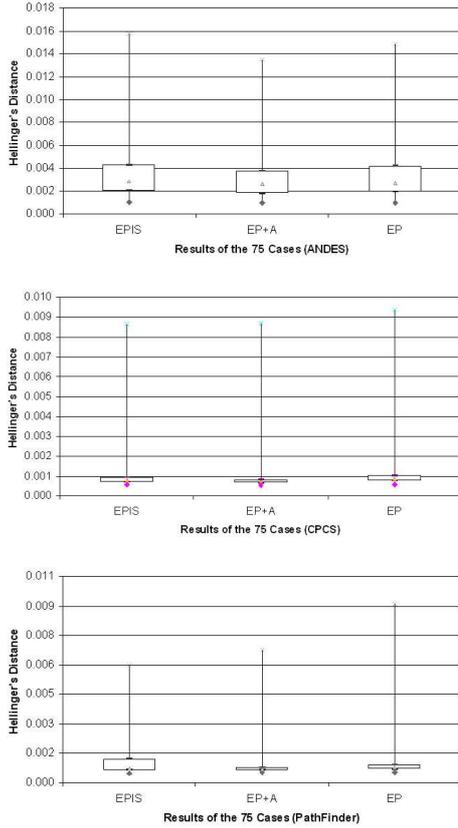


Figure 4: Summary boxplots of the results of the EP, EPIS, and EP+A algorithms for all three Bayesian networks.

## 6 Conclusion

The accuracy of importance sampling is very sensitive to the form of the importance function. In addition to the well known requirement that the importance function should have a similar shape to the target density (Andrieu et al., 2003; Rubinstein, 1981), it is also highly recommended that the importance function possesses heavy tails (Geweke, 1989; MacKay, 1998; Yuan and Druzdzal, 2004). The  $\epsilon$ -cutoff heuristic in the EPIS-BN algorithm (Yuan and Druzdzal, 2003) achieves this by cutting off small probabilities. However, it demonstrates inconsistent performances on different networks. In this paper, we analyze the underlying reasons behind the inconsistency and propose to use another heuristic, if-tempering, based on simulated tempering. By tempering a probability density, we

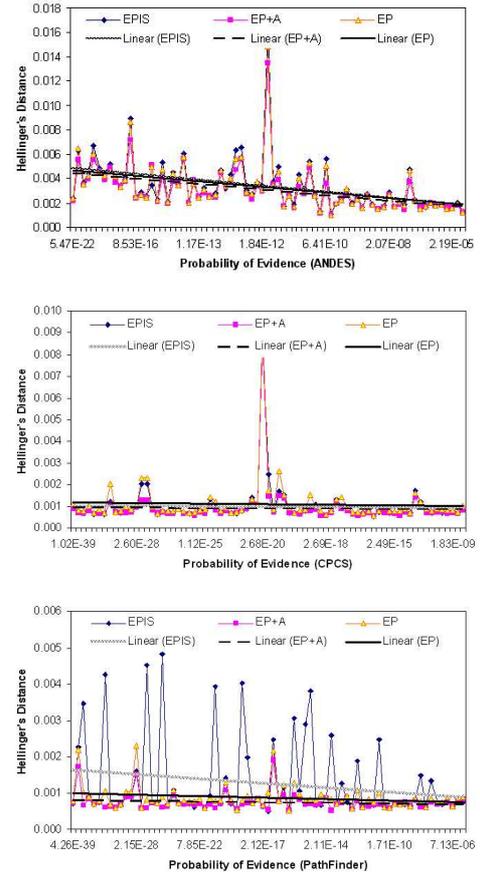


Figure 5: Performance of the EP, EPIS, and EP+A algorithms: Hellinger's distance for each test case plotted against the probability of evidence on all three Bayesian networks.

make it more flatter while largely keep the original shape. We tested the new heuristic on three large real Bayesian networks and observe that if-tempering helps the EPIS-BN algorithm to achieve considerable better precisions than  $\epsilon$ -cutoff consistently.

## References

- C. Andrieu, N. de Freitas, A. Doucet, and M. Jordan. 2003. An introduction to MCMC for machine learning. *Machine Learning*, 350:5–43.
- J. Cheng and M. J. Druzdzal. 2000. BN-AIS: An adaptive importance sampling algorithm for evidential reasoning in large Bayesian networks. *Journal of Artificial Intelligence Research*, 13:155–188.

- C. Conati, A. S. Gertner, K. VanLehn, and M. J. Druzdzel. 1997. On-line student modeling for coached problem solving using Bayesian networks. In *Proceedings of the Sixth International Conference on User Modeling (UM-96)*, pages 231–242, Vienna, New York. Springer Verlag.
- G. F. Cooper. 1990. The computational complexity of probabilistic inference using Bayesian belief networks. *Artificial Intelligence*, 42(2–3):393–405, March.
- P. Dagum and M. Luby. 1993. Approximating probabilistic inference in Bayesian belief networks is NP-hard. *Artificial Intelligence*, 60(1):141–153.
- M. J. Druzdzel. 1994. Some properties of joint probability distributions. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence (UAI-94)*, pages 187–194, Morgan Kaufmann Publishers San Francisco, California.
- R. Fung and K.-C. Chang. 1989. Weighing and integrating evidence for stochastic simulation in Bayesian networks. In M. Henrion, R.D. Shachter, L.N. Kanal, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence 5*, pages 209–219, New York, N. Y. Elsevier Science Publishing Company, Inc.
- R. Fung and B. del Favero. 1994. Backward simulation in Bayesian networks. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*, pages 227–234, San Mateo, CA. Morgan Kaufmann Publishers, Inc.
- J. Geweke. 1989. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica*, 57(6):1317–1339.
- D. Heckerman. 1990. Probabilistic similarity networks. *Networks*, 20(5):607–636, August.
- M. Henrion. 1988. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In *Uncertainty in Artificial Intelligence 2*, pages 149–163, New York, N.Y. Elsevier Science Publishing Company, Inc.
- L. D. Hernandez, S. Moral, and A. Salmeron. 1998. A Monte Carlo algorithm for probabilistic propagation in belief networks based on importance sampling and stratified simulation techniques. *International Journal of Approximate Reasoning*, 18:53–91.
- G. Kokolakis and P.H. Nanopoulos. 2001. Bayesian multivariate micro-aggregation under the Hellinger’s distance criterion. *Research in official statistics*, 4(1):117–126.
- S. L. Lauritzen and D. J. Spiegelhalter. 1988. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society, Series B (Methodological)*, 50(2):157–224.
- D. MacKay. 1998. *Introduction to Monte Carlo methods*. The MIT Press, Cambridge, Massachusetts.
- K. Murphy, Y. Weiss, and M. Jordan. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Proceedings of the Fifteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, pages 467–475, San Francisco, CA. Morgan Kaufmann Publishers.
- R. M. Neal. 1998. Annealed importance sampling. Technical report no. 9805, Dept. of Statistics, University of Toronto.
- L. Ortiz and L. Kaelbling. 2000. Adaptive importance sampling for estimation in structured domains. In *Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence (UAI-00)*, pages 446–454, Morgan Kaufmann Publishers San Francisco, California.
- M. Pradhan, G. Provan, B. Middleton, and M. Henrion. 1994. Knowledge engineering for large belief networks. In *Proceedings of the Tenth Annual Conference on Uncertainty in Artificial Intelligence (UAI-94)*, pages 484–490, San Mateo, CA. Morgan Kaufmann Publishers, Inc.
- R. Y. Rubinstein. 1981. *Simulation and the Monte Carlo Method*. John Wiley & Sons.
- R. D. Shachter and M. A. Peot. 1989. Simulation approaches to general probabilistic inference on belief networks. In M. Henrion, R.D. Shachter, L.N. Kanal, and J.F. Lemmer, editors, *Uncertainty in Artificial Intelligence 5*, pages 221–231, New York, N. Y. Elsevier Science Publishing Company, Inc.
- C. Yuan and M. J. Druzdzel. 2003. An importance sampling algorithm based on evidence pre-propagation. In *Proceedings of the 19th Conference on Uncertainty in Artificial Intelligence (UAI-03)*, pages 624–631, Morgan Kaufmann Publishers San Francisco, California.
- C. Yuan and M.J. Druzdzel. 2004. How heavy should the tails be? Under review.