

Decision Analysis by Augmented Probability Simulation

Author(s): Concha Bielza, Peter Müller and David Ríos Insua

Source: *Management Science*, Jul., 1999, Vol. 45, No. 7 (Jul., 1999), pp. 995-1007

Published by: INFORMS

Stable URL: <https://www.jstor.org/stable/2661697>

REFERENCES

Linked references are available on JSTOR for this article:

https://www.jstor.org/stable/2661697?seq=1&cid=pdf-reference#references_tab_contents

You may need to log in to JSTOR to access the linked references.

JSTOR is a not-for-profit service that helps scholars, researchers, and students discover, use, and build upon a wide range of content in a trusted digital archive. We use information technology and tools to increase productivity and facilitate new forms of scholarship. For more information about JSTOR, please contact support@jstor.org.

Your use of the JSTOR archive indicates your acceptance of the Terms & Conditions of Use, available at <https://about.jstor.org/terms>



INFORMS is collaborating with JSTOR to digitize, preserve and extend access to *Management Science*

JSTOR

Decision Analysis by Augmented Probability Simulation

Concha Bielza • Peter Müller • David Ríos Insua

Decision Analysis Group, Madrid Technical University, Spain

Institute of Statistics and Decision Sciences, Duke University, Durham, North Carolina 27706

School of Engineering, Universidad Rey Juan Carlos, Spain

We provide a generic Monte Carlo method to find the alternative of maximum expected utility in a decision analysis. We define an artificial distribution on the product space of alternatives and states, and show that the optimal alternative is the mode of the implied marginal distribution on the alternatives. After drawing a sample from the artificial distribution, we may use exploratory data analysis tools to approximately identify the optimal alternative. We illustrate our method for some important types of influence diagrams. (*Decision Analysis; Influence Diagrams; Markov Chain Monte Carlo; Optimal Design; Simulation*)

1. Introduction

1.1. Decision Analysis by Simulation

Decision Analysis provides a framework for solving decision making problems under uncertainty, based on finding an alternative with maximum expected utility. While conceptually simple, the actual solution of the maximization problem may be extremely involved, e.g., when the probability model is complex, the set of alternatives is continuous, or when a sequence of decisions is included. Therefore, only particular probability models are studied, such as the multivariate Gaussian in Shachter and Kenley (1989). Inclusion of continuous variables in simple problems is carried out through discretization (Miller and Rice 1983, Smith 1991), through summaries of the first few moments and derivatives (Smith 1993), or through approximations by means of Gaussian mixtures (Poland 1994). In complicated problems, there may be no hope for an exact solution method and we may have to turn to approximate methods, specifically, simulation.

As observed in Pearl (1988, p. 311) and Cooper (1989), in principle, any simulation method to solve Bayesian networks (BN) may be used to solve decision problems represented by influence diagrams (ID) by

means of sequentially instantiating decision nodes and computing expected values. Cooper notes that, for a given instantiation of the decision nodes, the computation of the expected value at the value node can be reformulated as a computation of a posterior distribution in an artificially created additional random node. The problem of solving BNs is summarized, for example, in Shachter and Peot (1990). Exact algorithms, e.g., using clique join trees (Lauritzen and Spiegelhalter 1988), cutset conditioning (Pearl 1986), or arc reversal (Shachter 1986, 1988), proved to be intractable in many real-world networks, leading to approximate inference algorithms based on simulation methods. These include short-run algorithms, such as Logic Sampling (Henrion 1988), Likelihood Weighting (Shachter and Peot 1990) and its improved modifications, Bounded Variance, and *AA* algorithms (Pradhan and Dagum 1996); and long-run algorithms, using Markov chain Monte Carlo methods like Gibbs sampling (Pearl 1987, Hrycej 1990, York 1992) or hybrid strategies (Brewer et al. 1996).

However, as Matzkevich and Abramson (1995) note, we only have a couple of outlines of simulation methods specifically for IDs in Jenzarli (1995) and Charnes and Shenoy (1996). Whereas the first one

combines stochastic dynamic programming and Gibbs sampling, the latter simulates *iid* observations from only a small set of chance variables for each decision node instead of using the entire distribution. Both become intractable when continuous decision spaces are included.

In recent statistical literature the same problem, i.e., that of finding the optimal action in a decision problem, has been considered in Müller and Parmigiani (1996) and Carlin et al. (1998), among others. Again, all these approaches use Monte Carlo simulation to evaluate the expected utility of given instantiations of nodes.

1.2. Augmented Probability Simulation

In this paper we propose a scheme that differs in important ways from the above mentioned approaches. Since they use simulation to evaluate expected utilities (losses) for given instantiations of the decision nodes, they do not accommodate continuous variables, especially decision variables, unless a discretization is carried out or the probability distributions are in a conjugate framework. In contrast, we go a step further and define an artificial distribution on *all* nodes, including the decision nodes. We show that simulation from this artificial augmented probability model is equivalent to solving the original decision problem. The specific strength of the proposed method is its generality. The algorithm can, in principle, accommodate arbitrary probability models and utility functions, as long as it is possible to pointwise evaluate the probability density and the utility function for any chosen value of all involved nodes. Evaluation of the probability density up to a constant factor suffices. The idea of augmenting the probability model to transform the optimization problem into a simulation problem is not entirely new. For example, Shachter and Peot (1992) have proposed a similar approach that involves augmenting the probability model to include the decision nodes and thus transforms the original optimization problem into a simulation problem. But to the best of our knowledge the approach described here is the first to solve this simulation problem by systematically exploiting Markov chain Monte Carlo simulation methods recently developed in the statistical literature.

The method starts by considering an artificial distribution on the space of alternatives and states. The distribution is defined in such a way that its marginal on the space of alternatives is proportional to the expected utility of the alternative and, consequently, the optimal alternative coincides with the mode of the marginal. Then, the proposed simulation-based strategy follows these steps: (i) Draw a sample from the artificial distribution; (ii) Marginalise it to the space of alternatives; and, (iii) Find the mode of the sample as a way of approximating the optimal alternative. A key issue is how to sample from the artificial distribution. For that we introduce Markov chain Monte Carlo (MCMC) algorithms. See, for example, Smith and Roberts (1993), Tierney (1994), or Tanner (1994) for a review of MCMC methods.

Section 2 describes the basic strategy with a simple example. Section 3 is of a more technical nature and provides generic methods to sample approximately from the artificial distribution and identify the mode of the sample. Section 4 discusses application examples. Section 5 compares our method with alternative schemes and identifies situations that call for different approaches.

2. Basic Approach

Here we outline the basic approach. Assume we have to choose under uncertainty an alternative d from a set \mathcal{A} . The set of states θ is Θ . We propose as optimal the alternative d^* with maximum expected utility: $\max_{d \in \mathcal{A}} [V(d) = \int u(d, \theta) p_d(\theta) d\theta]$, where $u(d, \theta)$ is the utility function modeling preferences over consequences and $p_d(\theta)$ is the probability distribution modeling beliefs, possibly influenced by actions. When the problem is structurally complicated, say a heavily asymmetric and dense, large influence diagram with continuous non-Gaussian random variables, nonquadratic utility functions and/or continuous sets of alternatives at decision nodes, finding the exact solution might be analytically and computationally intractable, and we might need an approximate solution method. We will provide such an approximation based on simulation.

Assume that $p_d(\theta) > 0$, for all pairs (d, θ) and $u(d, \theta)$, is positive and integrable. Define an artificial distribution

over the product space $\mathcal{A} \times \Theta$ with density h proportional to the product of utility and probability, specifically $h(d, \theta) \propto u(d, \theta) \cdot p_d(\theta)$. Note that the artificial distribution h is chosen so that the marginal on the alternatives is $h(d) = \int h(d, \theta)d\theta \propto V(d)$. Hence, the optimal alternative d^* coincides with the mode of the marginal of the artificial distribution h in the space of alternatives. As a consequence, we can solve the expected utility maximization problem approximately with the following simulation based approach: (i) draw a random sample from the distribution $h(d, \theta)$; (ii) convert it to a random sample from the marginal $h(d)$; and (iii) find the mode of this sample.

This augmented probability model simulation is conceptually different from other simulation algorithms reviewed earlier. Simulation is not used to pointwise evaluate expected utilities for each decision alternative. Instead, simulation generates the artificial probability model $h(\cdot)$ on the augmented state vector (d, θ) .

The key steps are (i) and (iii). For (ii), since we use simulation to generate from $h(d, \theta)$, we can get a marginal sample from $h(d)$ by simply discarding the simulated θ values. For (iii) we rely mainly on tools from exploratory data analysis, as we describe in §3.3. For (i), we will introduce generic Markov chain simulation methods. Their underlying idea is simple. We wish to generate a sample from a distribution over a certain space, but cannot do this directly. Suppose, however, that we can construct a Markov chain with the same state space, which is straightforward to simulate from and whose equilibrium distribution is the desired distribution. If we simulate sufficiently many iterations, after dropping an initial transient phase, we may use the simulated values of the chain as an approximate sample from the desired distribution. We will provide several algorithms for constructing chains with the desired equilibrium distribution, in our case the artificial distribution h , in §3.2. In the rest of this section we will provide an algorithm and a simple example, so that readers may grasp the basic idea, without entering into technical details. Readers familiar with MCMC simulation may skip directly to §3.

The strategy we propose now is very simple, but may be only undertaken in limited cases. Suppose the

conditional distributions $h(d | \theta)$ and $h(\theta | d)$ are available for efficient random variate generation. Then, we suggest the following scheme, which is known as the Gibbs sampler in the statistical literature (Gelfand and Smith 1990): (i) Start at an arbitrary value $d^0 \in \mathcal{A}$, and set $i = 1$; (ii) Generate $\theta^i \sim h(\theta | d^{i-1})$; (iii) Generate $d^i \sim h(d | \theta^i)$; Set $i = i + 1$ and repeat steps (ii) and (iii) until convergence is judged.

As a consequence of results in Tierney (1994) and Roberts and Smith (1994) we have:

PROPOSITION 1. *If the utility function is positive and integrable, $p_d(\theta) > 0$, for all pairs (d, θ) , and \mathcal{A} and Θ are intervals in \mathbb{R}^n , the above scheme defines a Markov chain with stationary distribution h .*

It is impossible to give generally applicable results about when to terminate iterations in Markov chain Monte Carlo simulations. It is well known that this is a difficult theoretical problem, see, e.g., Robert (1995) and Polson (1996), who discuss approaches to find the number of iterations that will ensure convergence in total variation norm within a given distance to the true stationary distribution. However, practical convergence may be judged with a number of criteria, see, e.g., Cowles and Carlin (1996) or Brooks and Roberts (1999). Most of these methods have been implemented in CODA (Best et al. 1995), which we have used in our examples. Once practical convergence has been judged, say after k iterations, we may record the next N iterations of the simulation output $(d^1, \theta^1), \dots, (d^N, \theta^N)$, and use (d^1, \dots, d^N) as an approximate sample from $h(d)$. From that we may try to assess the mode.

We illustrate the above approach with an artificial example, adapted from Shenoy (1994).

EXAMPLE 1. A physician has to determine a policy for treating patients suspected of suffering from a disease D . D causes a pathological state P that, in turn, causes symptom S to be exhibited. The physician observes whether ($S = 1$) or not ($S = 0$) a patient is exhibiting the symptom. Based on this information, she either treats T ($T = 1$) the patient (for P and D) or does not ($T = 0$). The physician's utility function depends on T , P , and D , as shown in Table 1. The value 0.001 was changed from the original value (0) to adapt to the general result in Proposition 1. The

Table 1 The Probability Model $p(D, P, S)$, the Physician's Utility Function $u(T, P, D)$, and $h(d, D, P, S)$

D	P	S	$p(D, P, S)$	$u(T = d_s, D, P)$				$h(d, D, P, S) \propto u(\cdot)p(\cdot)$			
				$d = (0, 0)$	$(0, 1)$	$(1, 0)$	$(1, 1)$	$d = (0, 0)$	$(0, 1)$	$(1, 0)$	$(1, 1)$
0	0	0	0.612	8	4	8	4	4.896	2.448	4.896	2.448
0	0	1	0.153	8	8	4	4	1.224	1.224	0.612	0.612
0	1	0	0.0405	2	6	2	6	0.081	0.243	0.081	0.243
0	1	1	0.0945	2	2	6	6	0.189	0.189	0.567	0.567
1	0	0	0.016	1	8	1	8	0.016	0.128	0.016	0.128
1	0	1	0.004	1	1	8	8	0.004	0.004	0.032	0.032
1	1	0	0.024	0.001	10	0.001	10	0.000	0.240	0.000	0.240
1	1	1	0.056	0.001	0.001	10	10	0.000	0.000	0.560	0.560

Note. The probabilities used in steps (ii) and (iii) of the Markov chain Monte Carlo scheme described in the text are proportional to the entries in the appropriate column and row, respectively, of the h section at the right of the table.

probability of disease D ($D = 1$) is 0.1. For patients known to suffer from D , 80% suffer from P ($P = 1$). On the other hand, for patients known not to suffer from D ($D = 0$), 15% suffer from P . For patients known to suffer from P ($P = 1$), 70% exhibit symptom S , and for patients known not to suffer from P ($P = 0$), 20% exhibit S . We assume that D and S are probabilistically independent given P . To implement the proposed algorithm, we need to find the conditional distributions $h(\theta | d)$ and $h(d | \theta)$. In this case, $\theta = (D, P, S)$ and $d = (d_1, d_0)$, where d_1 is the decision taken if the symptom is exhibited, and d_0 , if it is not exhibited. $d_s = 1(0)$ means to treat (not to treat) the patient. Let $p(D, P, S) = p(D)p(P | D)p(S | P)$ denote the probabilities given in the above description. With $h(d, D, P, S) \propto u(T = d_s, P, D)p(D, P, S)$ we find $h(D, P, S | d_1, d_0) \propto u(T = d_s, P, D)p(D, P, S)$, and $h(d_1, d_0 | D, P, S) \propto u(T = d_s, P, D)$.

Our proposed method goes as follows: (i) Start at an arbitrary decision (d_1^0, d_0^0) and set $i = 1$; (ii) Generate $\theta^i \sim h(\theta | d^{i-1}) = h(D, P, S | d_1^{i-1}, d_0^{i-1})$; (iii) Generate $d^i \sim h(d | \theta^i) = h(d_1, d_0 | D^i, S^i, P^i)$; Set $i = i + 1$ and repeat steps (ii) and (iii) until convergence is judged. Once convergence is judged, we record the next N iterations of the algorithm $(\theta^1, d^1), \dots, (\theta^N, d^N)$, and use (d^1, \dots, d^N) as an approximate sample from the marginal in d of the artificial distribution. We leave out some values between those recorded to avoid serial correlation. Since alternatives are finite in number, we just need to inspect the histogram to approx-

imate the mode. From a simulated sample of size 1000, we find that the optimal decision is $d = (d_1 = 1, d_0 = 0)$, that is treat if symptom is present and not treat if symptom is absent.

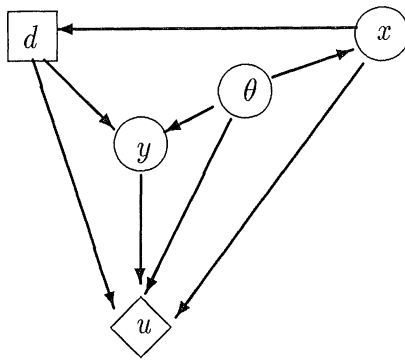
Note in the example how the proposed augmented probability model simulation differs from other simulation methods proposed for the solution of IDs. We use *one* simulation over the joint (θ, d) space to simulate from $h(\cdot)$ instead of many small simulations to evaluate expected utilities for each possible decision one at a time. Of course, the previous example is extremely simple in that we are able to sample from $h(d | \theta)$ and $h(\theta | d)$, and, by inspection of the histogram, we may approximate the modes. The following sections deal with more complex cases.

3. Sampling from the Artificial Distribution

We shall provide here a generic method to sample from the artificial distribution $h(\cdot)$. Typically, this distribution will not be straightforward to simulate from, requiring generation from possibly high dimensional models, including complex probability and utility functions, continuous decision and chance nodes, and possibly conditioning on observed data. MCMC simulation schemes are the most commonly used methods known to accommodate such generality; hence we choose them.

Given the enormous interest in IDs as a tool for

Figure 1 A generic influence diagram for our scheme.



structuring and solving decision problems, see, e.g., Matzkevich and Abramson (1995), we concentrate on such structures. An ID is a directed graph representation of a decision problem as a probability network with additional nodes representing decisions and values. For notational purposes, we will partition the set of nodes into five subsets, differentiating three types of chance nodes: (i) Decision nodes d , representing decisions to be made. (ii) Chance nodes, including random variables x observed prior to making the decision, i.e., data available at the time of decision making; not yet observed random variables y , i.e., data which will only be observed after making the decisions; and unobservable random variables θ , i.e., unknown parameters; (iii) One value node u representing the utility function $u(d, x, \theta, y)$. Figure 1 provides a simple generic ID for our scheme. An ID is solved by determining a decision d^* with maximum expected utility. This requires marginalizing over all chance nodes (y, θ), conditioning on x , and maximizing over d . See Shachter (1986) for a complete description and an algorithm to solve IDs.

The method we propose here is applicable to IDs with nonsequential structure, i.e., decision nodes must not have any chance nodes as predecessors that have distributions depending, in turn, on other decision nodes. Except for some technical conditions there will be no further requirements.

3.1. The Probability Model Defined by Influence Diagrams

An ID defines the conditional distributions $p(x|\theta)$, $p(\theta)$ and $p_a(y|\theta)$, a joint distribution on (x, θ, y) via

$p_a(x, \theta, y) = p(\theta)p(x|\theta)p_a(y|\theta)$, and a conditional distribution $p_a(\theta, y|x) \propto p(\theta)p(x|\theta)p_a(y|\theta)$, for (θ, y) given the observed nodes x . Typically, x and y are independent given θ , allowing the given factorization, and $p(\theta)$ does not depend on d . If a particular problem does not fit this setup, modifications of the proposed algorithm are straightforward.

In the context of this probability model, solving the ID amounts to maximizing the expected utility over d , where $p_a(\theta, y|x)$ is the relevant distribution to compute this expectation. In summary, solving the ID amounts to finding

$$\max_d \left[V(d) = \int u(d, x, \theta, y) dp_a(\theta, y|x) \right]. \quad (1)$$

We solve this problem approximately by simulation. Augment the probability measure $p_a(\theta, y|x)$ to a probability model for (θ, y, d) by defining a joint p.d.f.

$$h(d, \theta, y) \propto u(d, x, \theta, y)p_a(\theta, y|x).$$

The mode of the implied marginal distribution $h(d) \propto \int u(d, x, \theta, y)p_a(\theta, y|x)d\theta dy = V(d)$ corresponds to the optimal decision d^* . The underlying rationale of our method is to simulate a Markov chain in (θ, y, d) , defined to have $h(d, \theta, y)$ as its asymptotic distribution. For big enough t , the simulated values (θ^t, y^t, d^t) from successive states of the simulated process provide, approximately, a Monte Carlo sample from $h(d, \theta, y)$. Note that the simulation is defined on an augmented probability model $h(d, \theta, y)$ rather than on $p_a(\theta, y)$ for each possible instantiation of the actions d , as traditional methods do. By considering the marginal distribution of d^t in this Monte Carlo sample, we can infer the optimal decision using methods such as those discussed in §3.3.

The key issue is the definition of a Markov chain with the desired limiting distribution $h(\cdot)$. For that, we capitalise on recent work in numerical Bayesian inference concerning the application of Markov chain Monte Carlo methods to explore high dimensional distributions which do not allow analytic solutions for expectations, marginal distributions, etc.

3.2. Markov Chain Monte Carlo Simulation

We shall provide a general algorithm which will be valid for all IDs satisfying the structural conditions

specified above and some minor technical conditions discussed below. The algorithm we describe is of the Metropolis type (Tierney 1994): We generate a new candidate for the states from a probing distribution, and then move to that new state or stay at the old one according to certain probabilities. We do this transition in three steps, for d , θ , and y . We only require to be able to evaluate the utility function $u(d, x, \theta, y)$ and the probability distributions $p_a(y | \theta)$, $p(\theta)$, $p(x | \theta)$, for any relevant d , x , θ , y . This will typically be possible, since the definition of the ID includes explicit specification of these distributions, i.e., the modeler is likely to specify well-known distributions.

The scheme requires specification of probing distributions g_1 , g_2 , and g_3 . The choice of probing distributions $g_j(\cdot | \cdot)$ is conceptually arbitrary, with the only constraint that the resulting Markov chain should be irreducible and aperiodic. As we shall argue, whenever possible, we assume symmetric probing distributions, i.e., satisfying $g(a | b) = g(b | a)$. For example, $g(a | b)$ could be a multivariate normal distribution $N(b, \Sigma)$ for some Σ . Details about the choice of probing distribution are discussed in the appendix. We then have:

ALGORITHM 1.

1. Start at values (d^0, θ^0, y^0) for decisions, parameters and outcomes, and set $i = 1$.
2. Let $u_1 = u(d^{i-1}, x, \theta^{i-1}, y^{i-1})$;
Generate a "proposal" $\tilde{d} \sim g_1(\tilde{d} | d^{i-1})$ and evaluate $\tilde{u}_1 = u(\tilde{d}, x, \theta^{i-1}, y^{i-1})$;
Compute $a_1 = \min\{1, [h(\tilde{d}, \theta^{i-1}, y^{i-1})] / [h(d^{i-1}, \theta^{i-1}, y^{i-1})]\} = \min\{1, (\tilde{u}_1 / u_1) \cdot [p_a(y^{i-1} | \theta^{i-1})] / [p_{\tilde{a}}(y^{i-1} | \theta^{i-1})]\}$;
With probability a_1 , set $d^i = \tilde{d}$; otherwise, keep $d^i = d^{i-1}$.
3. Let $u_2 = u(d^i, x, \theta^{i-1}, y^{i-1})$;
Generate a "proposal" $\tilde{\theta} \sim g_2(\tilde{\theta} | \theta^{i-1})$ and evaluate $\tilde{u}_2 = u(d^i, x, \tilde{\theta}, y^{i-1})$;
Compute $a_2 = \min\{1, [h(d^i, \tilde{\theta}, y^{i-1})] / [h(d^i, \theta^{i-1}, y^{i-1})]\} = \min\{1, (\tilde{u}_2 / u_2) \cdot [p(\tilde{\theta})p(x | \tilde{\theta})p_a(y^{i-1} | \tilde{\theta})] / [p(\theta^{i-1})p(x | \theta^{i-1})p_a(y^{i-1} | \theta^{i-1})]\}$;
With probability a_2 , set $\theta^i = \tilde{\theta}$; otherwise, keep $\theta^i = \theta^{i-1}$.
4. Let $u_3 = u(d^i, x, \theta^i, y^{i-1})$.
Generate a proposal $\tilde{y} \sim g_3(\tilde{y} | y^{i-1})$ and evaluate

$$\tilde{u}_3 = u(d^i, x, \theta^i, \tilde{y}).$$

$$\text{Compute } a_3 = \min\{1, [h(d^i, \theta^i, \tilde{y})] / [h(d^i, \theta^i, y^{i-1})]\} = \min\{1, (\tilde{u}_3 / u_3) \cdot [p_{\tilde{a}}(\tilde{y} | \theta^i)] / [p_{\tilde{a}}(y^{i-1} | \theta^i)]\}.$$

With probability a_3 set $y^{i-1} = \tilde{y}$; otherwise keep $y^i = y^{i-1}$.

5. Set $i = i + 1$. Repeat steps 2 through 4 until chain is judged to have practically converged.

This algorithm defines a Markov chain, with $h(d, \theta, y)$ as stationary distribution. The generality of this algorithm comes at a price, namely possible slow convergence. Depending on the application, long simulation runs might be required to attain practical convergence. However, this fully general algorithm is rarely required.

Many problems allow simpler algorithms based on using $p(\theta | x)$ and $p_a(y | \theta)$ to generate proposals. Algorithm 2, given below, only requires a probing distribution $g(\tilde{d} | d)$ for d , evaluation of the utility function and algorithms to generate from $p(\theta | x)$ and $p_a(y | \theta)$. While simulating from $p_a(y | \theta)$ is typically straightforward, simulating from $p(\theta | x)$ is not. In general, this distribution will not be explicitly specified in the ID, but needs to be computed through repeated applications of Bayes formula, or several arc reversals in the language of IDs. However, note that simulating from $p(\theta | x)$ amounts to solving the statistical inference problem of generating from the posterior distribution on θ given the data x . Hence, we can appeal to versions of posterior simulation schemes appropriate for a variety of important inference problems recently discussed in the Bayesian literature (see, e.g., Smith and Roberts 1993, Tanner 1994, and Tierney 1994). Before starting the algorithm described below, we generate a sufficiently large Monte Carlo sample from $p(\theta | x)$ by whatever simulation method is most appropriate.

ALGORITHM 2.

1. Start at values (d^0, θ^0, y^0) and set $i = 1$;
2. Evaluate $u^i = u(d^{i-1}, x, \theta^{i-1}, y^{i-1})$;
3. Generate $(\tilde{d}, \tilde{\theta}, \tilde{y}) \sim g(\tilde{d} | d^{(i-1)})p_a(\tilde{\theta}, \tilde{y} | x) = g(\tilde{d} | d^{(i-1)})p(\tilde{\theta} | x)p_{\tilde{a}}(\tilde{y} | \tilde{\theta})$;
4. Evaluate $\tilde{u} = u(\tilde{d}, x, \tilde{\theta}, \tilde{y})$;
5. Compute $a = \min\{1, [h(\tilde{d}, \tilde{\theta}, \tilde{y})] / [h(d^{i-1}, \theta^{i-1}, y^{i-1})] \cdot [p_{\tilde{a}}(\theta^{i-1}, y^{i-1} | x)] / [p_{\tilde{a}}(\tilde{\theta}, \tilde{y} | x)]\} = \min(1, \tilde{u} / u^i)$;

6. With probability a , set $(d^i, \theta^i, y^i) = (\tilde{d}, \tilde{\theta}, \tilde{y})$; otherwise, keep $(d^i, \theta^i, y^i) = (d^{i-1}, \theta^{i-1}, y^{i-1})$.
7. Set $i = i + 1$. Repeat steps 2 through 6 until convergence is practically judged.

In step 3, generation of $\theta \sim p(\theta | x)$ is done using the simulated Monte Carlo sample generated earlier.

ALGORITHM 3. The algorithm simplifies if x is missing in the ID, i.e., if no data is given at the time of the decision. The associated Algorithm 3 would be stated as Algorithm 2, with the proposal distribution in step 3 replaced by $(\tilde{d}, \tilde{\theta}, \tilde{y}) \sim g(\tilde{d} | d) p_{\tilde{a}}(\tilde{\theta}, \tilde{y}) = g(\tilde{d} | d) p(\tilde{\theta}) p_{\tilde{a}}(\tilde{y} | \tilde{\theta})$. Sampling from $p_a(\theta, y) = p(\theta) p_a(y | \theta)$ will be feasible in general, since these distributions are defined explicitly in the ID.

3.3. Finding the Optimal Solution

The MCMC simulation provides us with an approximate simulated sample $\{(d^1, \theta^1, y^1), \dots, (d^N, \theta^N, y^N)\}$ from $h(d, \theta, y)$, from which we deduce an approximate sample (d^1, \dots, d^N) from the marginal $h(d)$. The mode of $h(d)$ is an approximation of the optimal alternative.

In the case of discrete alternatives, the problem is simple since we only have to count the number of times each element has appeared, and choose the one with the highest frequency. It may be worthwhile retaining not one but several of the most frequent decisions, and study them in further detail, as a way of conducting sensitivity analysis.

In the case of continuous alternatives, as a first approach we may use graphical exploratory data analysis tools, especially with low dimensional decision vectors. When the decision vector d is one- or two-dimensional, we may produce a histogram (or a smooth version) and inspect it to identify modes. For higher dimensional decision vectors d , we propose to consider the problem as one of cluster analysis. Modes of $h(d)$ correspond to d s with higher density, which suggests looking for regions with higher concentration of sampled d s. This leads us to compute a hierarchical cluster tree for the simulated points d^t . Since we are assuming h to be a density with respect to Lebesgue measure in \mathbb{R}^n , and we are interested in identifying regions where the optimal alternative might lie, we suggest using complete linkage with Euclidean dis-

tance. Once we have a classification tree, we cut at a certain height and obtain the corresponding clusters. The location of the largest cluster indicates the area of the best decision. Again, as before, it may be useful to keep several larger clusters and explore the corresponding regions. The result of course would depend on the cutting height, but by exploring several heights we may be able to identify several decisions of interest. We illustrate the approach in §4.2.

4. Examples

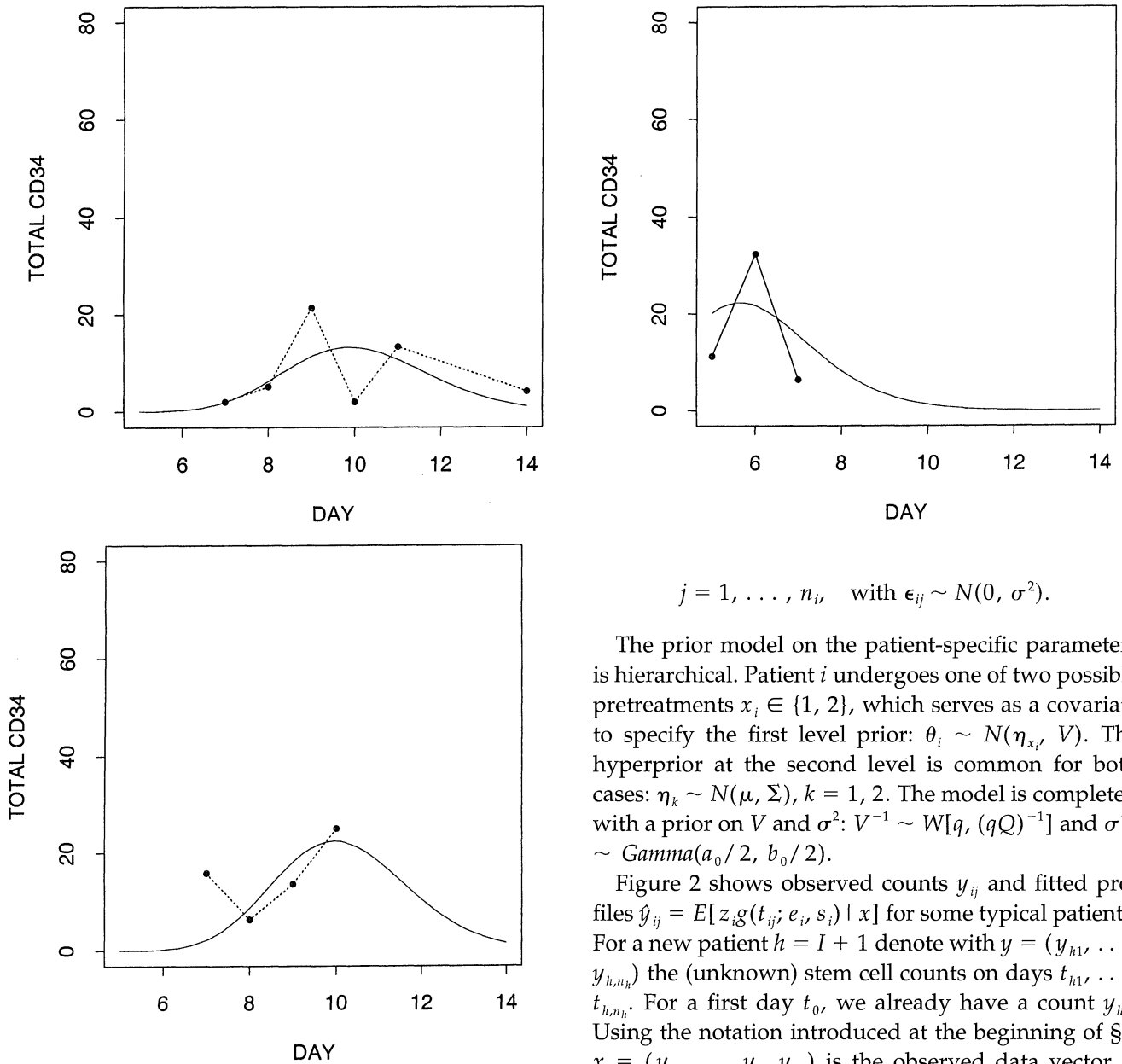
4.1. Example 2: A Medical Decision Making Problem

We illustrate the algorithm with a case study concerning the determination of optimal apheresis designs for cancer patients undergoing chemotherapy. Palmer and Müller (1998) describe the clinical background and solve the problem by large scale Monte Carlo integration.

Between a pretreatment and start of chemotherapy, stem cells (CD34) are collected to allow later reconstitution of white blood cell components. Depending on the pretreatment, the first stem cell collection process (apheresis) is scheduled on the fifth or seventh day after pretreatment. A decision is to be made on which days between pretreatment and treatment we should schedule stem cell collections to (i) collect some target number of cells; and (ii) minimize the number of aphereses. We have data on $I = 22$ past patients, and for the first day of the new patient.

Let y_{ij} , $i = 1, \dots, I$, and $j = 1, \dots, n_i$, denote the observed CD34 count for patient i on day t_{ij} . Also, $y_i = (y_{i1}, \dots, y_{i,n_i})$ shall designate the i -th's patient data and $x = (y_1, \dots, y_I)$ the combined data vector. Palmer and Müller (1998) specify the following probability model for this process. The likelihood is based on the observation that the typical profile of stem cell counts over days shows first a rise after pretreatment, reaches a maximum, and then slowly declines back towards the base level, as shown in Figure 2. To model such shapes we use a nonlinear regression model. Let $g(t; e, s) = 1/c\Gamma(t; a, b)$ denote a Gamma probability density function with parameters $b = e/s^2$, $a = e \cdot b$ chosen to imply mean and variance matching e and s^2

Figure 2 Three typical patients. The dashed lines connect the data points. The solid curve plots the fitted profile, using the described probability model.



$$j = 1, \dots, n_i, \quad \text{with } \epsilon_{ij} \sim N(0, \sigma^2).$$

The prior model on the patient-specific parameters is hierarchical. Patient i undergoes one of two possible pretreatments $x_i \in \{1, 2\}$, which serves as a covariate to specify the first level prior: $\theta_i \sim N(\eta_{x_i}, V)$. The hyperprior at the second level is common for both cases: $\eta_k \sim N(\mu, \Sigma), k = 1, 2$. The model is completed with a prior on V and σ^2 : $V^{-1} \sim W[q, (qQ)^{-1}]$ and $\sigma^{-2} \sim \text{Gamma}(a_0/2, b_0/2)$.

Figure 2 shows observed counts y_{ij} and fitted profiles $\hat{y}_{ij} = E[z_i g(t_{ij}; e_i, s_i) | x]$ for some typical patients. For a new patient $h = I + 1$ denote with $y = (y_{h1}, \dots, y_{h, n_h})$ the (unknown) stem cell counts on days $t_{h1}, \dots, t_{h, n_h}$. For a first day t_0 , we already have a count y_{h0} . Using the notation introduced at the beginning of §3, $x = (y_1, \dots, y_I, y_{h0})$ is the observed data vector, $y = y_h$ is the future data vector, and $\theta = (\theta_1, \dots, \theta_I, \theta_{I+1}, \eta_1, \eta_2, V, \sigma)$ are the unobservable parameters in the model. Given the typical profile, the optimal decision will schedule aphereses for all days between some initial day d_0 and a final day d_1 , i.e., the decision parameter is $d = (d_0, d_1)$.

Let A be the event of failing to collect a target number y^* of stem cells,

and rescaled by $c = [(a - 1)/b]^{a-1} \exp(-(a - 1))$, so that $\text{sup}(g) = 1$. We use $g(\cdot; e, s)$ to parametrize a nonlinear regression for the profiles through time of each patient:

$$y_{ij} = z_i g(t_{ij}; e_i, s_i) + \epsilon_{ij},$$

$$i = 1, \dots, I,$$

$$A = \left\{ \sum_{j=d_0}^{d_1} y_{hj} L_h < y^* \right\},$$

where L_h is the volume of blood processed at each stem cell collection for the new patient. Let $n_a = d_1 - d_0 + 1$ denote the number of scheduled stem cell collections. The utility function is $u(d, x, \theta, y) = -cn_a - p \Pr(A | d, \theta)$, where c is the sampling cost and p a penalty for underachievement of the target. We need to maximize over d the expected utility $V(d) = \int u(d, x, \theta, y) dp(y | \theta) dp(\theta | x)$. Note that the probability model $p(\theta)p(x, y | \theta)$ does not depend on the decision nodes, but there is data x influencing the belief model. Since $p(\theta | x)$ may be actually sampled with a Markov chain Monte Carlo method described in Palmer and Müller (1998), we use Algorithm 2 to solve the problem. To ensure a positive utility function we add a constant offset to $u(\cdot)$.

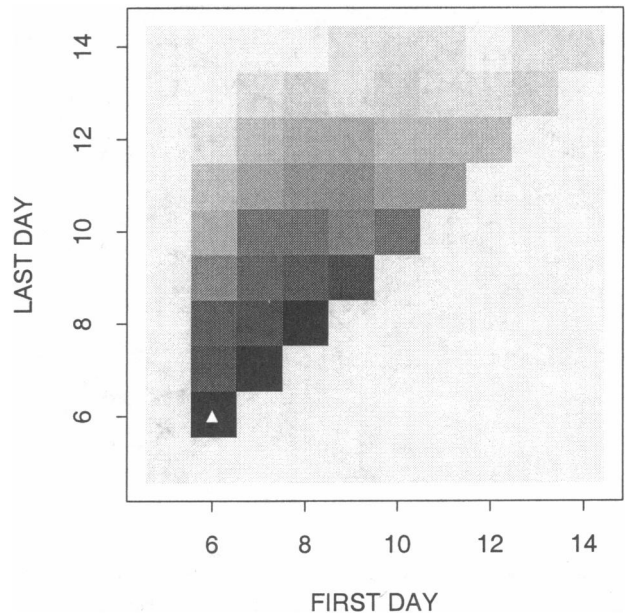
We found the optimal design d^* for a future patient with the above belief and preference model when $p/c = 10.0$. For a patient undergoing treatment $x_h = 1$ with a first observation $y_0 = 20.0$ on day $t_{h,0} = 5$, the optimal apheresis schedule for the remaining six days was found to be given by $d_0 = 6, d_1 = 6$. Since the decision space is two dimensional, we can do this by a simple inspection of the histogram. Figure 3 plots the estimated distribution $h(d) \propto V(d)$.

4.2. Example 3: A Water Reservoir Management Problem

In Ríos Insua et al. (1997), we describe a complex multi-period decision analysis problem concerning the management of two reservoirs: Lake Kariba (K) and Cahora Bassa (C). Here we solve a simplified version using the proposed MCMC approach to simulate from the augmented probability model.

We want to find, for a given month, optimal values to be announced for release from K and C through turbines and spillgates, $d_1^k, d_2^k, d_1^c, d_2^c$, respectively. The actual amounts of water released depend on the water available, which is uncertain, since there is uncertainty about inflows i^k and i^c to the reservoirs. There is a forecasting model for both i^k and i^c , the latter being dependent on the water released from K and the incremental inflows (inc), which, in turn, depend on a parameter β . The preference model combines utilities

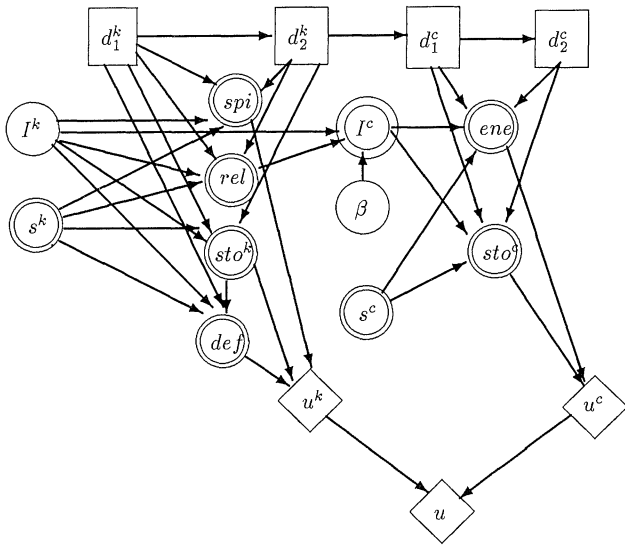
Figure 3 The grey shades show a histogram of the simulated d^i for the medical problem. Inspection of $h(d)$ reveals the optimal decision at $d = (6, 6)$ (white triangle).



for both K and C . Those for K depend on the energy deficit (def), the final storage (sto^k) and the amount of water spilled (spi). Those for C depend on the energy produced (ene) and the final storage (sto^c). Initial storages s^k and s^c have influence as well over actual releases. Figure 4 shows the influence diagram representing the problem. Nodes with double border are either known values or deterministic functions of their predecessors. They are required to compute the value node u , but will not show up in the probability model. In terms of our notation, the problem includes four decision nodes $d = (d_1^k, d_2^k, d_1^c, d_2^c)$ and two chance nodes i^k and β .

Figure 5 shows some profiles of the histogram of the simulated $d^i \sim h(d)$, generated by Algorithm 3. The decision parameter is four-dimensional. Hence we used a four-dimensional grid (with $10 \times 10 \times 10 \times 10$ cells) to record a four-dimensional histogram of the simulated states. Simple inspection of the empirical distribution allows to read off the optimal release at $d^* = \{d_1^{k*} = 3247, d_2^{k*} = 1000, d_1^{c*} = 4121, d_2^{c*} = 200\}$. The solution is based on 100,000 simulated values from the Markov chain Monte Carlo scheme. Figure 5 illustrates also another feature of our method which is

Figure 4 Influence diagram for the reservoir problem.



a simple sensitivity analysis procedure at no extra cost. Darkness in Figure 5b suggests that expected utility is rather flat when releases through turbines are fixed at their optimal values, hence suggesting insensitivity with respect to changes in spill. On the other hand, Figure 5a, with just one dark area where the estimated optimum is, suggests that expected utility is fairly peaked in release through turbines, and hence very sensitive to changes in energy releases.

Alternatively, as discussed in §3.3, we consider a hierarchical cluster tree of the simulation output. The dots in Figure 5 show the solution based on cutting a hierarchical cluster tree of 1000 simulated values $d \sim h(d)$ at height 2000 and finding the cluster with the most members. The optimum is found at $d^* = \{d_1^{k*} = 3353, d_2^{k*} = 742, d_1^{c*} = 3616, d_2^{c*} = 476\}$. This comes reasonably close to the optimum estimated earlier.

5. Discussion

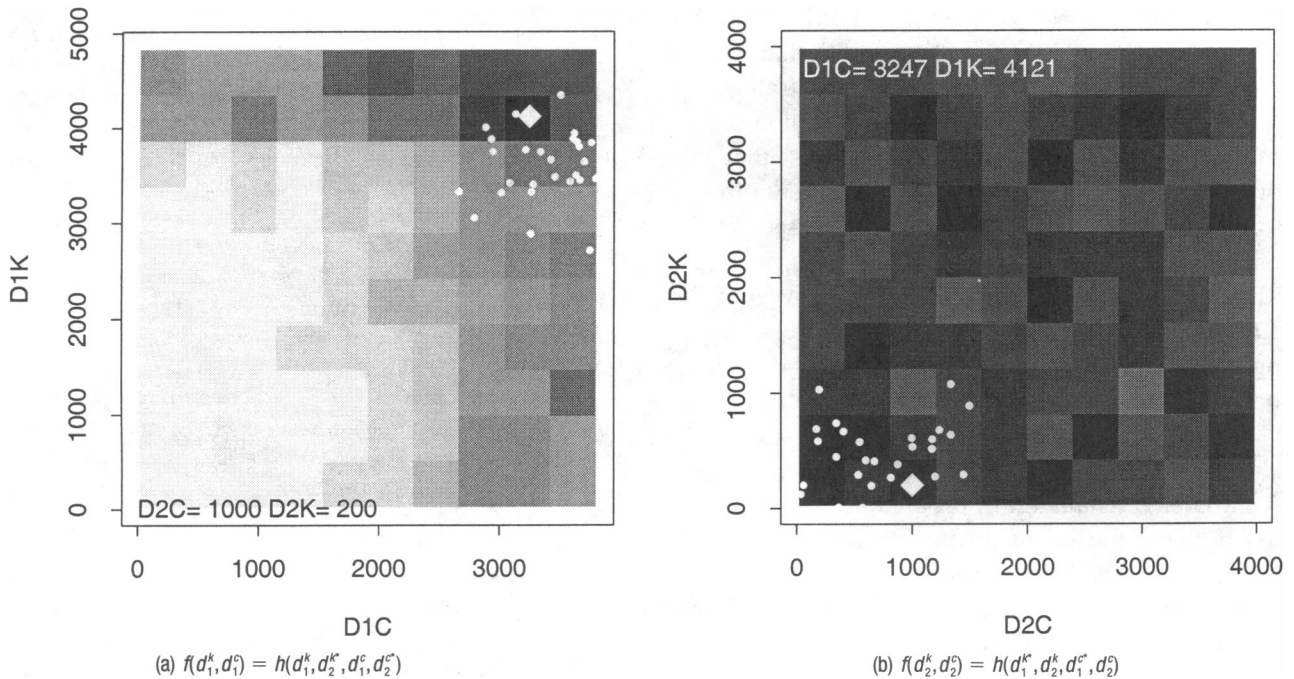
5.1. Comparison with Alternative Schemes

The scheme described in Algorithms 1, 2, and 3 transforms the original expected utility maximization problem (1) into a simulation problem. Our scheme is very generic, in the sense of accommodating arbitrary probability models, be they discrete or continuous, and utility functions, as long as the probability density

(or probability mass function) and the utility function are pointwise evaluable. The main difference with other simulation schemes earlier considered in the literature is that instead of using simulation to evaluate expected utilities (losses) for each possible instantiation of decisions, we use simulation from an artificial auxiliary model which augments the original probability model to include an artificial distribution over decision nodes. Whether one or the other approach is more efficient depends on the specifics of the considered decision problem. No general comparisons are possible. Even in specific examples, performance will depend heavily on arbitrary choices like the amount of discretization, which is necessary for many methods; run length of the involved Monte Carlo simulations; chosen MCMC scheme, etc. However, some general observations about the relative efficiency of the methods are possible.

In problems with few alternatives, analytic solutions using methods like arc reversal (Shachter 1986), and simulation methods which use simulation to pointwise evaluate expected utilities, like Likelihood Weighting (Shachter and Peot 1990), are typically more efficient than simulation over the auxiliary probability model. Bielza and Shenoy (1998) discuss a decision problem (the “reactor problem”) with six possible actions, and chance nodes with less than 10 possible joint outcomes. An exact solution using Shachter’s (1986) algorithm requires one arc reversal and the largest state space used during the solution phase contains four variables. By comparison, we implemented the same example using augmented probability simulation, following Algorithm 3. We used 100,000 iterations in the MCMC simulation. The computational effort of one iteration is approximately comparable to one arc reversal. Thus the exact solution is clearly far more efficient in terms of computing time. Alternatively, consider simulation to compute the expected utility of each of the six possible actions, using, for example, Likelihood Weighting. Considering the involved numerical standard errors, Monte Carlo simulation sizes of around 1000 simulations for each alternative decision would be adequate. Thus, also

Figure 5 (a) Expected utility as a function of release for energy (d_1^k, d_1^c) with spill fixed at the optimal levels; and (b) as a function of spill (d_2^k, d_2^c) with release through turbines fixed at the optima. The diamond indicates the optimal decision. The dots indicate the simulations in the largest cluster of the hierarchical clustering tree cut at height 2000.



Likelihood Weighting dominates simulation from the augmented probability model.

In problems where the optimal decision is to be computed conditional on some already available data x the comparison changes, especially if the posterior distribution of the unknown parameters is significantly different from the initial prior distribution, i.e., under low prior probability for the evidence x . Consider, for example, the application reported in §4.1, which is not amenable to exact methods. Using Monte Carlo simulation to compute expected utilities for alternative decisions, we can no longer generate independent samples. Following Jenzarli's (1995) proposal we could use Gibbs sampling to compute expected utilities. Depending on the specific choices of the implemented MCMC scheme and termination criteria, one would typically use on the order of 10,000 iterations (Palmer and Müller 1998). Discretizing the sample space, one could, in principle, also use Logic Sampling (Henrion, 1988). However, Logic Sampling would not be advisable for this problem since the fraction of simulated experiments which gen-

erate variables corresponding to the actual observations would be close to zero (i.e., $p(x_i^*) \approx 0$, in the notation of Shachter and Peot 1990). For similar reasons, Likelihood Weighting (Shachter and Peot 1990) would fail. Since only leaf nodes are observed, the sample scores would be proportional to the likelihood function, i.e., the scheme would amount to importance sampling using the prior probability model as importance sampling function. This can, however, be addressed using bounded variance type algorithms as discussed, for example, in Pradhan and Dagum (1996).

Finally, many decision problems involve continuous decision variables, like the example considered in §4.2. Continuous decision parameters create no problem for simulation from the augmented probability model, but would not allow a straightforward application of any scheme based on evaluating expected utilities for one decision at a time. Even if discretization was used, say on a $10 \times 10 \times 10 \times 10$ grid, the resulting number of alternative actions renders such schemes difficult to use.

5.2. Conclusion

Complex decision problems may render impossible the application of exact methods to obtain optimal decisions. As a consequence, we should look for approximation methods, including simulation.

We have proposed a simulation-based strategy for approximating optimal decisions in a decision analysis. Our experiments and examples suggest that this approach may be very powerful. Implementation of the algorithms is fairly straightforward based on the schemes provided. Specific cases may require simple modifications such as the ones suggested in §3.2. The exploration of the sample in search for modes may be done with standard statistical software. As we mentioned in the discussion of Example 3, one feature of our method is the provision of simple sensitivity analysis features, at no extra cost.

A number of challenging problems remain, particularly perhaps, the extension of our scheme to sequential decisions. The straightforward approach of expanding the model to non-sequential normal form may only be applied when the number of decision nodes is small. Another challenging problem would be to develop a computational environment based on our approach. It would be also interesting to develop further methods to look for modes in multivariate settings.

Similar ideas may be pursued to solve traditional statistical optimal design problems. From a formal point of view, an optimal design problem can be described as a stochastic optimization problem (1). This is explored in Clyde et al. (1995) for the special case of Algorithm 3 with continuous sample spaces and nonsequential setup.¹

¹ Research supported by grants from the National Science Foundation, CAM, CICYT, CICYT-HID and the Iberdrola Foundation. Parts of it took place while Peter Müller was visiting UPM and David Ríos Insua was visiting CNR-IAMI. We are grateful for discussions with Mike Wiper.

Appendix: Implementation

The choice of the probing distributions $g_j(\cdot | \cdot)$ in Algorithm 1 is conceptually arbitrary, with the only constraint that the resulting Markov chain be irreducible and aperiodic.

In the statement and proofs of the proposed algorithms, we assumed g to be symmetric in its arguments, i.e., $g(a|b) = g(b|a)$.

If $d \in \mathbb{R}^p$ is a continuous parameter, we propose to use a normal kernel $g(\tilde{d}|d) = N(d, \Sigma)$ with some appropriately chosen covariance matrix Σ , for example, a diagonal matrix with diagonal entries corresponding to reasonable step sizes in each of the decision parameters. Good values for the step size can be found by trial and error with a few values. In a particular setup, Gelman, Roberts and Gilks (1996) show that the optimal choice of step size should result in average acceptance probabilities around 25%, and similarly, for other parameters.

If d is discrete, a simple choice for $g(\tilde{d}|d)$ could generate $d - 1$ and $d + 1$ with probability 0.5. Of course, many other problem specific choices are possible. In Example 2, e.g., we define $d = (d_0, d_1)$ by choosing with probability $\frac{1}{6}$ one of six possible moves: (i) increase d_0 and d_1 by 1 day; (ii) decrease d_0 and d_1 by 1; (iii) increase d_0 by 1; (iv) decrease d_0 by 1; etc.

Should symmetry of g be violated, an additional factor $g(d|\tilde{d})/g(\tilde{d}|d)$ would be added in the expressions for acceptance probabilities. This would correspond to Metropolis-Hastings steps, rather than Metropolis steps. Convergence proofs for the proposed scheme are simple, based on results in Tierney (1994) and Roberts and Smith (1994).

References

- Best, N., M., Cowles, K. Vines. 1995. *CODA: Convergence Diagnosis and Output Analysis Software for Gibbs Sampling Output, Version 0.30*, MRC Biostatistics Unit, University of Cambridge, Cambridge, England.
- Bielza, C., P. Shenoy. 1998. A comparison of graphical techniques for asymmetric decision problems. To appear in *Management Sci.*
- Brewer, M., C. Aitken, M. Talbot. 1996. A comparison of hybrid strategies for Gibbs sampling in mixed graphical models. *Comput. Statist. Data Anal.* **21** 343–365.
- Brooks, S., G. Roberts. 1999. Assessing convergence of Markov chain Monte Carlo algorithms. To appear in *Statist. Comput.*
- Carlin, B., J. Kadane, A. Gelfand. 1998. Approaches for optimal sequential decision analysis in clinical trials. *Biometrics* **54** 964–975.
- Charnes, J., P. Shenoy. 1996. A forward Monte Carlo method for solving influence diagrams using local computation. Working paper, School of Business, University of Kansas.
- Clyde, M., P. Müller, G. Parmigiani. 1995. Exploring expected utility surfaces by Markov changes. Working paper, Duke University, Durham, NC.
- Cooper, G. 1989. A method for using belief networks as influence diagrams. *Uncertainty in Artificial Intelligence, Volume 4*. Association for Uncertainty in AI, Inc., Rome, N.Y., 55–63.
- Cowles, M., B. Carlin. 1996. Markov chain Monte Carlo convergence diagnostics: a comparative review. *J. Amer. Statist. Assoc.* **91** 883–904.
- Gelfand, A., A. Smith. 1990. Sampling based approaches to calculating marginal densities. *J. Amer. Statist. Assoc.* **85** 398–409.
- Gelman, A., G. Roberts, W. Gilks. 1996. Efficient Metropolis jumping

- rules. J. M. Bernardo, J. O. Berger, A. P. Dawid, A. F. M. Smith, eds., *Bayesian Statistics 5*, Oxford University Press.
- Henrion, M. 1988. Propagating uncertainty in bayesian networks by probabilistic logic sampling. J. Lemmer, L. Kanal, eds., *Uncertainty in Artificial Intelligence, Volume 2*. North Holland, 149–163.
- Hrycej, T. 1990. Gibbs sampling in Bayesian networks. *Artificial Intelligence* 46 351–364.
- Jenzarli, A. 1995. Solving influence diagrams using Gibbs sampling. Working paper, College of Business, University of Tampa, FL.
- Lauritzen, S., D. Spiegelhalter. 1988. Local computations with probabilities on graphical structures and their application to expert systems. *J. Royal Statist. Soc. B*, 50 157–224.
- Matzkevich, I., B. Abramson. 1995. Decision analytic networks in Artificial Intelligence. *Management Sci.* 41 1–22.
- Miller, A. C., T. R. Rice. 1983. Discrete approximations of probability distributions. *Management Sci.* 29 352–363.
- Müller, P., G. Parmigiani. 1996. Optimal design via curve fitting of Monte Carlo experiments. *J. Amer. Statist. Assoc.* 90 1322–1330.
- Palmer, J. L., P. Müller. 1998. Bayesian optimal design in population models of hematologic data. *Statist. in Medicine* 17 1613–1622.
- Pearl, J. 1986. Fusion, propagation and structuring in belief networks. *Artificial Intelligence* 29 241–288.
- Pearl, J. 1987. Evidential reasoning using stochastic simulation of causal models. *Artificial Intelligence* 32 245–257.
- . 1988. *Probabilistic reasoning in intelligent systems*. Morgan-Kaufmann, San Francisco, CA.
- Poland, W. B. 1994. *Decision Analysis with continuous and discrete variables: A mixture distribution approach*. PhD thesis, Stanford University, Dept. of Engineering-Economic Systems, Stanford, CA.
- Polson, N. G. 1996. Convergence of Markov chain Monte Carlo algorithms. J. M. Bernardo, J. O. Berger, A. P. Dawid, A. F. M. Smith, eds., *Bayesian Statistics 5*. Oxford University Press.
- Pradhan, M., P. Dagum. 1996. Optimal Monte Carlo estimation of belief network inference. E. Horvitz, F. Jensen, eds., *Uncertainty in Artificial Intelligence, Volume 12*. Morgan-Kaufmann, San Francisco, 446–453.
- Ríos Insua, D., K. Salewicz, P. Müller, C. Bielza. 1997. Bayesian methods in reservoir operations: the Zambezi river case. S. French, J. Q. Smith, eds., *The Practice of Bayesian Analysis*. Arnold, London, UK 107–130.
- Robert, C. P. 1995. Convergence control methods for Markov chain Monte Carlo algorithms. *Statist. Sci.* 10 473–511.
- Roberts, G., A. Smith. 1994. Simple conditions for the convergence of the Gibbs sampler and Metropolis-Hastings algorithms. *Stochastic Processes and Their Appl.* 49 207–216.
- Shachter, R. 1986. Evaluating influence diagrams. *Oper. Res.* 34 871–882.
- . 1988. Probabilistic inference and influence diagrams. *Oper. Res.* 36 589–604.
- , R. Kenley. 1989. Gaussian influence diagrams. *Management Sci.* 35 527–550.
- , M. Peot. 1990. Simulation approaches to general probabilistic inference on belief networks. M. Henrion, R. Shachter, L. Kanal, J. Lemmer, eds., *Uncertainty in Artificial Intelligence, Volume 5*. Elsevier, New York, 221–231.
- , ———. 1992. Decision making using probabilistic inference methods. *Uncertainty in Artificial Intelligence, Volume 8*. Morgan Kaufmann, San Francisco, CA. 276–283.
- Shenoy, P. 1994. A comparison of graphical techniques for decision analysis. *European J. Oper. Res.* 78 1–21.
- Smith, A. 1991. Bayesian computational methods. *Philosophical Trans. Royal Soc. London, Series A.* 337 369–386.
- Smith, J. E. 1993. Moment methods for decision analysis. *Management Sci.* 39 340–358.
- Smith, A., G. Roberts. 1993. Bayesian computation via the Gibbs sampler and related Markov chain Monte Carlo methods. *J. Royal Statist. Soc., B* 55 3–24.
- Tanner, M. 1994. *Tools for Statistical Inference*. Springer-Verlag, New York.
- Tierney, L. 1994. Markov chains for exploring posterior distributions (with discussion). *Ann. Statist.* 22 1701–1762.
- York, J. 1992. Use of the Gibbs sampler in expert systems. *Artificial Intelligence* 56 115–130.

Accepted by Robert Nau; received April 2, 1996. This paper has been with the authors 11 months for 4 revisions.