A FORWARD MONTE CARLO METHOD FOR SOLVING INFLUENCE DIAGRAMS USING LOCAL COMPUTATION

John M. Charnes*
jcharnes@ukans.edu

Prakash P. Shenoy
pshenoy@ukans.edu

School of Business
University of Kansas
Summerfield Hall
Lawrence, KS 66045–2003 USA

Abstract

The main goal of this paper is to describe a Monte Carlo method for solving influence diagrams using *local computation*. The forward Monte Carlo sampling technique draws independent and identically distributed observations. Methods that have been proposed in this spirit sample from the entire distribution. However, when the number of variables is large, the state space of all variables is exponentially large, and the sample size required for good estimates is too large to be practical. In the forward Monte Carlo method we generate observations from a subset of chance variables for each decision node in the influence diagram. We use methods developed for exact solution of influence diagrams to limit the number of chance variables sampled at any time. Because influence diagrams model each chance variable with a conditional probability distribution, the forward Monte Carlo solution method lends itself very well to influence-diagram representations.

Key Words: Influence Diagrams, Monte Carlo Methods, Local Computation

1 Introduction

The main goal of this paper is to investigate a forward Monte Carlo method for solving influence diagrams using local computation. Influence diagrams are a compact representation of Bayesian decision problems. They were proposed initially as a front-end for decision trees [Miller et al. 1976, Howard and Matheson 1984]. Subsequently, Olmsted [1983] and Shachter [1986] devised a method for solving influence diagrams directly, i.e., without having to transform them to decision trees. The influence diagram solution technique, called arc-reversal, uses local computation both for the computation of the conditionals and for computation of an optimal strategy.

In the last decade, there have been many improvements to the influence diagram representation and solution technique. Ezawa [1986] has examined efficient deletion sequences for solving influence diagrams. Tatman [1986] and Tatman and Shachter [1990] describe an extension of the influence diagram technique for a decomposition of the joint utility function. Shenoy [1992] has proposed a generalization of influence diagrams, called valuation networks, which allow for representation of any probability model (whereas influence diagrams allow for representation of only conditional probability models). Ndilikilikesha [1992, 1994] has adapted Shenoy's valuation network technique to influence diagrams for the case of an undecomposed utility function. Smith [1989], Shachter and Peot [1992], Zhang et al. [1994], Jensen et al. [1994], Cowell [1994], Goutis [1995], and Qi and Poole [1995] have proposed modifications to the symmetric influence diagram technique to make the representation and solution more efficient. Call and Miller [1990], Smith et al. [1993], Fung and Shachter [1990], Covaliu and Oliver [1996], and Qi et al. [1994] have proposed modifications to the influence diagram technique for representing and solving asymmetric decision problems. Finally, Shenoy [1993, 1996] has proposed a generalization of the symmetric valuation network technique for asymmetric decision problems.

All research described in the previous paragraph deals with discrete decision problems, i.e., problems in which all chance and all decision variables have a discrete state space. For problems in which some of the decision and/or chance variables are continuous, several approximate approaches have been proposed. The traditional approach is to discretize the continuous variables to a few states [Howard and

*Comments and suggestions for improvement are welcome and will be gratefully appreciated.

Matheson, 1983, 1984, Miller and Rice 1983, Keefer 1994, Smith 1991]. A related approach is to summarize continuous distributions by their first few moments, summarize continuous utility functions by their first few derivatives, and then use either the moments and derivatives directly [Howard 1971], or discretize the continuous variables based on the moments and derivatives [Smith 1993]. A more recent approach is to deal with continuous variables directly without discretization. For example, Kenley and Shachter have studied influence diagram methodology for decision problems in which the probability model is multivariate Gaussian [Kenley 1986, Shachter and Kenley 1989], and Poland [1994] has developed influence diagrams that use Gaussian mixtures to approximate arbitrary continuous distributions.

In this paper, we describe a forward Monte Carlo sampling technique that draws independent and identically distributed observations. Although our long term goal is to develop methods for solving decision problems that have a mixture of discrete and continuous variables, in this paper we restrict ourselves to a problem in which all variables are discrete. Monte Carlo methods that have been proposed in this spirit sample from the entire distribution (see, e.g., Hertz [1964], Hertz and Thomas [1983, 1984] for decision problems, and Henrion [1988], Pearl [1987], and Shachter and Peot [1990] for Bayesian networks, which are influence diagrams without decision and value nodes). However, when the number of variables is large, the state space of all variables is exponentially large, and the sample size required for good estimates is too large to be practical. In the forward Monte Carlo method described here observations are generated from only a small set of variables for each decision node in the influence diagram. We use methods developed for exact solution of influence diagrams to limit the number of variables sampled at any time. Since influence diagrams model each chance variable with a conditional probability distribution, the forward Monte Carlo solution method lends itself very well to influence-diagram representations.

2 Used Car Buyer's Problem

In this section, we give a complete statement of the used car buyer's (UCB) problem [Howard 1962]. This problem is highly asymmetric. Howard [1962] describes a decision tree representation and solution of this problem. Smith, Holtzman and Matheson [1993] describe a representation and solution of this problem based on a generalization of the symmetric influence diagram technique for asymmetric decision problems.

A statement of the UCB problem is as follows. Joe is considering buying a used car from a dealer for \$1,000. The market price of similar cars with no defects is \$1,100. Joe is uncertain whether the particular car he is considering is a "peach" or a "lemon." Of the ten major subsystems in the car, a peach has a serious defect in only one subsystem, whereas a lemon has a serious defect in six subsystems. The probability that the used car under consideration is a lemon is 0.2. The cost of repairing one defect is \$40, and the cost of repairing six defects is \$200.

For an additional \$60, Joe can buy the car from the dealer with an "anti-lemon guarantee." The anti-lemon guarantee will normally pay for 50% of the repair cost, but if the car is a lemon, then the guarantee will pay 100% of the repair cost.

Before buying the car, Joe has the option of having the car examined by a mechanic for an hour. In this time period, the mechanic offers three alternatives: t_1—Test the steering subsystem alone at a cost of \$9; t_2—Test the fuel and electrical subsystems for a total cost of \$13; and t_3—Perform a two-test sequence in which Joe can authorize a second test after the result of the first test is known. In this alternative, the mechanic will first test the transmission subsystem at a cost of \$10 and report the results to Joe. If Joe approves, the mechanic will then proceed to test the differential subsystem at an additional cost of \$4. All tests are guaranteed to find a defect in the subsystems if a defect exists. We assume that Joe's utility for profit is linear in dollars.

A decision tree representation and solution of this problem is given by Charnes and Shenoy [1996]. The optimal strategy is to choose test t_2; if both systems are non-defective then buy with no guarantee, else buy with guarantee. The maximum expected utility is \$32.87.

3 Influence Diagram Representation

A mathematical representation of a Bayesian decision problem can be broken into four parts: (*i*) *Alternatives*—the sets of alternatives available to the decision maker; (*ii*) *Uncertainty Model*—a probability model of the uncertainties faced by the decision maker; (*iii*) *Preferences for Outcomes*—a utility function model of the preferences of the decision

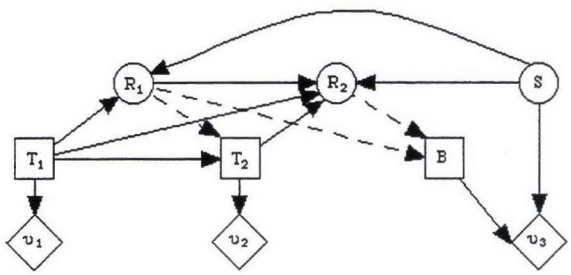

Figure 1: An influence diagram representation of the UCB problem at the graphical level.

Ω_{T_1}	v_1
nt	0
t_1	-9
t_2	-13
t_3	-10

Ω_{T_2}	v_2
nc	0
c	-4
s	0

Ω_{T_2}	v_2
(b,p)	60
(b,l)	-100
(g,p)	20
(g,l)	40
$(\sim b,p)$	0
$(\sim b,l)$	0

Table 1: The additive factors of the utility function in the UCB problem.

maker for all possible outcomes; and (*iv*) *Information Constraints*—specifications of what information the decision maker has available at times when the decision maker must choose an alternative.

An influence diagram representation of a decision problem is specified at two levels, graphical, and numerical. Figure 1 shows an influence diagram representation of the UCB at the graphical level. Figure 2 and Table 1 show the influence diagram representation at the numerical level.

In the UCB problem influence diagram, T_1 (first test), T_2 (second test), and B (buy) are rectangular decision nodes representing decision variables; R_1 (first test results), R_2 (second test results), and S (state of car), are circular chance nodes representing chance variables; and v_1, v_2, and v_3 are diamond-shaped value nodes representing additive factors of the joint utility function.

The solid arrows pointing to decision variables indicate the domain of the conditionals for the decision variables. By *domain*, we mean the subset of variables that are included in the conditional. In the UCB problem the domain for the conditional for T_1 is denoted as $\mathrm{Dom}(T_1) = \{T_1\}$. Also, $\mathrm{Dom}(T_2) = \{T_1, T_2\}$, and $\mathrm{Dom}(B) = \{B\}$.

A conditional for a decision variable can be thought of as a constraint on the set of possible alternatives available to the decision maker at that variable. Since $\mathrm{Dom}(T_1) = \{T_1\}$ and $\mathrm{Dom}(B) = \{B\}$, there are no constraints on the choices at these decision variables. $\mathrm{Dom}(T_2) = \{T_1, T_2\}$ implies that the choices available at T_2 (the variable at the head of the solid arrows) depends on the choice made at $\{T_1\}$ (the set of variables at the tails of the solid arrows pointing to T_2).

The solid arrows pointing to decision nodes can also be interpreted as information constraints. If there is a solid arrow from node V (either chance or decision) to decision node D, then this is interpreted as the true state of V is known to the decision maker at the time (s)he has to choose an alternative at D.

The solid arrows pointing to chance nodes specify the domains of the conditionals for the chance variables in the same sense as for decision variables. In the UCB problem, we have $\mathrm{Dom}(S) = \{S\}$, $\mathrm{Dom}(R_1) = \{T_1, S, R_1\}$, and $\mathrm{Dom}(R_2) = \{T_1, T_2, R_1, S, R_2\}$. This means we have the following conditionals: $P(S)$, $P(R_1|T_1, S)$, and $P(R_2|T_1, T_2, R_1, S)$.

The solid arrows pointing to value nodes indicate the domain of the corresponding utility function in the sense that the domain of the utility function at a value node is the set of variables at the tails of the arrows pointing to it. In the UCB problem, $\mathrm{Dom}(v_1) = \{T_1\}$, $\mathrm{Dom}(v_2) = \{T_2\}$, and the $\mathrm{Dom}(v_3) = \{B, S\}$.

Finally, the dashed arrows pointing to decision nodes indicate information constraints such that if there is a dashed arrow from a node V to a decision node D, then it means that the decision maker knows the true value of V at the time (s)he has to make a choice at D. A solid arrow pointing to a decision node indicates conditional information and information constraints, whereas a dashed arrow indicates only information constraints. The absence of an arrow from chance node C to decision node D means that the decision maker must make a choice at D without knowing the true value of C. In the UCB problem, at T_1, Joe has observed nothing; at T_2, Joe has observed the true state of R_1 but knows neither the true state of R_2 nor the true state of S; and at B, Joe has observed the true states of R_1 and R_2, but not S.

The influence diagram literature usually assumes a "no-forgetting" condition that specifies that if there is a directed path from a decision node D_i to decision node D_j, then this implies that the decision maker remembers her/his choice at D_i at the time (s)he has to make a choice at D_j. This constraint can be (optionally) represented in the influence diagram as a dashed arrow from D_i to D_j if

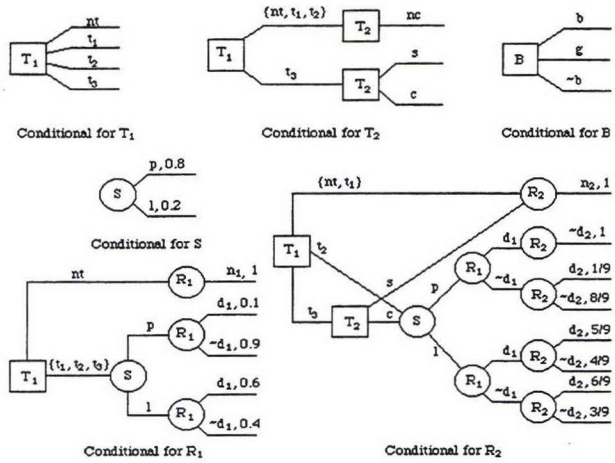

Figure 2: The conditionals for the decision and chance nodes in the UCB problem.

there does not exist an arrow from D_i to D_j already, but since such constraints are deduced, we can safely leave them out of the graphical representation. In the UCB problem, we already have a solid arrow from T_1 to T_2, and we can deduce dashed arrows from T_1 to B, and from T_2 to B.

At the numerical level, we specify the details of the conditionals for each decision and chance nodes and the details of the utility functions. In the UCB problem, the state spaces for the variables are as follows: $\Omega_{T_1} = \{nt, t_1, t_2, t_3\}$ (i.e., no test, test 1, test 2, test 3), $\Omega_{R_1} = \{n_1, d_1, \sim d_1\}$ (i.e., no results, defective, not defective), $\Omega_{T_2} = \{nc, s, c\}$ (i.e., no choice, stop, continue), $\Omega_{R_2} = \{n_2, d_2, \sim d_2\}$ (i.e., no results, defective, not defective), $\Omega_B = \{b, g, \sim b\}$ (i.e., buy with no guarantee, buy with guarantee, not buy), and $\Omega_S = \{p, l\}$ (i.e., peach, lemon).

The conditionals for the decision and chance nodes are shown in Figure ?? using Smith, Holtzman and Mathesons [1993] "distribution trees" notation. A distribution tree consists of paths, called "conditioning scenarios," that lead to "atomic distributions." For decision nodes, the atomic distributions are the sets of alternatives available to the decision maker in each conditioning scenario. For chance nodes, the atomic distributions are conditional probability distributions conditioned on the paths leading to the atomic distributions.

In the UCB problem, the distribution tree for T_1 has no conditioning scenarios and one atomic distribution Ω_{T_1}; the distribution tree for T_2 has two atomic distributions $\{nc\}$ and $\{s, c\}$ depending on the conditioning scenarios; and the distribution tree for B has one atomic distribution Ω_B. The distribution tree for S has one atomic distribution; the distribution tree for R_1 has three atomic distributions; and the distribution tree for R_2 has five atomic distributions.

4 Forward Monte Carlo Method

In this section, we describe a forward Monte Carlo method for solving influence diagrams using local computation and discuss the solution of the Used Car Buyers (UCB) problem.

4.1 Forward Monte Carlo Method

In the forward Monte Carlo method, in each stage, we determine a decision function for a decision variable. To do so, we iteratively sample the state of each variable in a subset of variables in a sequence called a "sampling sequence." After we have completed a specified number of iterations, we determine a decision function for the decision variable based on maximum expected utility.

There are at least five related questions:

1. Which decision variable? Suppose Γ denotes the set of chance variables and D denotes the set of decision variables. Each decision variable D partitions the set Γ of chance variables into two sets: a set $O_D = \{C \in \Gamma |$ there is an arrow (either solid or dashed) from C to $D\}$ of "observed" chance variables, and a set $U_D = \Gamma \backslash O_D$ of "unobserved" chance variables. Define a partial order $<$ on the set $\Gamma \cup \Delta$ of chance and decision variables recursively as follows: (a) If C is a chance variable, D is a decision variable, and $C \in O_D$ ($C \in U_D$), then $C < D$ ($D < C$); (b) If C_1 and C_2 are chance variables, and there exists a decision variable D such that $C_1 < D$ and $D < C_2$, then $C_1 < C_2$; and (c) If D_1 and D_2 are decision variables, and there exists a chance variable C such that $D_1 < C$ and $C < D_2$, then $D_1 < D_2$. We choose a decision variable that is maximal with respect to the partial order, $<$. Let D^* denote such a variable. (Since $<$ is a partial order, there is at least one.)

In the UCB problem, we have a complete order $T_1 < R_1$ (using Rule (b) above), $R_1 < T_2$ (using Rule (a)), $T_2 < R_2$ (using Rule (a)), $R_2 < B$ (using Rule (a)), and $B < S$ (using Rule (a)). Thus a decision variable that is maximal with respect to the partial order $<$ is B. Once we have a decision function for B, we pick T_2 next, and finally T_1.

2. Which additive factors of the utility function are relevant? Let Ψ denote the set of

all utility nodes in the influence diagram. It follows from the method proposed by Tatman and Shachter [Tatman 1986, Tatman and Shachter 1990] that only utility functions whose domain includes a variable in $\{D^\star\} \cup U_{D^\star}$ are relevant for the determination of the decision function for $D^\star$, i.e. the set of relevant utility functions is $\mathrm{RU}(D^\star) = \{\upsilon \in \Psi | \mathrm{Dom}(\upsilon) \cap (\{D^\star\} \cup U_{D^\star}) \neq \phi\}$.

Let $J(D^\star) = \cup\{\mathrm{Dom}(\upsilon) | \upsilon \in \mathrm{RU}(D^\star)\}$ denote the union of the domains of the relevant utility functions.

In the UCB problem, for determining a decision function for B, $U_B = \{S\}$, $\mathrm{RU}(B) = \{\upsilon_3\}$ (since υ_3 is the only utility function that includes B and S in its domain), and $J(B) = \{B, S\}$.

3. What is the smallest relevant domain of the decision function for decision variable $D^\star$? Consider all conditionals and additive factors of the utility function that have some variable in $J(D^\star)$ in their domains. Let $H(D^\star) = \cup\{\mathrm{Dom}(V) | V \in \Gamma \cup D \cup \Psi \ni \mathrm{Dom}(V) \cap J(D^\star) \neq \phi\}$ denote the union of their domains.

It follows from the arc reversal method of Olmsted [1983] and Shachter [1986] that the smallest relevant domain of the decision function for $D^\star$, denoted by $\mathrm{RD}(D^\star)$, is $H(D^\star) \backslash U_{D^\star}$.

In the UCB problem, for decision variable B, $H(B) = \{S, B, T_1, R_1, T_2, R_2\}$, and $\mathrm{RD}(B) = \{T_1, R_1, T_2, R_2, B\}$.

4. What subset of variables do we sample to determine the decision function for the decision variable chosen in 1 and in what sequence? Let $\prec$ denote a partial order on $\Gamma \cup \Delta$ defined as follows. Suppose $V_1, V_2 \in \Gamma \cup \Delta$. Then $V_1 \prec V_2$ if and only if there is a directed path from V_1 to V_2 in the influence diagram representation. Since the influence diagram is acyclic, it follows that the binary order $\prec$ is a partial order. Notice that the partial order $\prec$ is different from the partial order $<$ defined earlier. In the UCB problem we have, for example, $S \prec B$ (since there is a directed path from S to B), and $B < S$ (since the true state of S is in general not known when Joe has to pick an alternative from Ω_B).

Suppose $D^\star$ is the decision variable selected in 1. Recall that $J(D^\star)$ denotes the union of domains of all relevant utility functions. Clearly, we need to sample all variables in $J(D^\star)$. Also, we need to sample all unobserved chance variables that influence the variables in $J(D^\star)$. First consider the subset of variables that precede the variables in $J(D^\star)$, $\{V \in \Gamma \cup \Delta | V \prec W$ for some $W \in J(D^\star)\}$. Next, from this subset we select those that are unobserved at $D^\star$ and consider these along with $J(D^\star)$, i.e., $(\{V \in \Gamma \cup \Delta | V \prec W$ for some $W \in J(D^\star)\} \cap U_{D^\star}) \in J(D^\star) = L(D^\star)$, say. Next, for ease of forward sampling, we consider all variables that lie between the variables in $L(D^\star)$ with respect to the partial order $\prec$, and denote this set by $G(D^\star) = [L(D^\star) \cup \{V \in \Gamma \cup \Delta | W \prec V$ for some $W \in L(D^\star)\}] \backslash \{V \in \Gamma \cup \Delta | W \prec V$ for all $W \in L(D^\star)\}$.

To determine a decision function for $D^\star$ by forward Monte Carlo sampling, we need to sample each variable $V \in G(D^\star)$ using its conditional. Notice that to sample variable V using its conditional, we will need to have fixed the states of the variables in $\mathrm{Dom}(V) \backslash \{V\}$. If these variables are not in $G(D^\star)$, we sample for these using the equiprobable distribution. Thus we sample all variables in the set $K(D^\star) = \cup\{\mathrm{Dom}(V) | V \in G(D^\star)\}$ using either the conditional distribution specified in the influence diagram representation (for variables in $G(D^\star)$) or the equiprobable conditional (for variables not in $G(D^\star)$).

In the UCB problem, to determine a decision function for B, $G(B) = \{S, R_1, T_2, R_2, B\}$, and $K(B) = \{T_1, S, R_1, T_2, R_2, B\}$.

We sample the variables in $K(D^\star)$ in a sequence such that if $V_1 \prec V_2$ then V_1 must precede V_2 in the sequence. We call such a sequence a *sampling sequence*. The sampling sequence is motivated by the fact that in an influence diagram there is a conditional probability distribution for each chance and decision variable given its direct predecessors. In a given sampling sequence, all direct predecessors of V will have been sampled when it is time to sample V, and thus the conditional distribution for V can be used at that point in the sequence. Working our way through the sampling sequence for the variables in $K(D^\star)$ constitutes one iteration of the forward Monte Carlo simulation.

In the UCB problem, to determine a decision function for B, there are two sampling sequences $ST_1R_1T_2R_2B$, and $T_1SR_1T_2R_2B$.

5. How many iterations? There is a large body of results in the simulation output analysis literature that is relevant here [e.g., Law 1983]. The general problem is to determine the number of iterations as a function of the pre-specified precision in the computation of an approximately optimal strategy. The number of iterations will depend on the variances of the distributions that are being sampled. In each stage, we sample from a family of conditional distributions for the variables being sampled, one for each possible decision function. If we know the maximum variance of the family of conditional distributions, then we can conservatively

estimate the number of iterations as a function of the precision that is pre-specified by the decision maker. In the algorithm below, the number of iterations required for the specified precision is denoted $n(D^\star)$.

In each stage of the simulation, we determine a decision function for the maximal decision variable $D^\star$. Before we start the next stage, we replace the decision variable $D^\star$ by a chance variable whose conditional distribution is the decision function, and we replace the set of utility functions $\mathrm{RU}(D^\star)$ by the utility function whose domain is $\mathrm{RD}(D^\star)\backslash\{D^\star\}$ and whose values are the corresponding maximum mean utility values found in the determination of the optimal decision function for $D^\star$. We repeat this process recursively as indicated below.

4.2 Forward Monte Carlo Algorithm

Given an influence diagram representation of a decision problem:

1. Pick a maximal decision variable $D^\star$. If there are no decision variables, then stop;
2. Pick the subset of variables $K(D^\star)$ to be sampled, and a sampling sequence;
3. Run $n(D^\star)$ iterations of the model by generating observations on each variable in the order given by the sampling sequence. For variables in $G(D^\star)$, use the conditional distribution specified in the influence diagram. For variables not in $G(D^\star)$, use the equiprobable conditional distribution. After each iteration, store the vector of observations on $\mathrm{RD}(D^\star)$ and the utility obtained from the relevant utility functions based on the sampled values of each variable in $J(D^\star)$;
4. Pick an optimal decision function whose domain is $\mathrm{RD}(D^\star)$ based on maximizing expected utility, which is estimated by averaging the utility values stored in Step 3;
5. Make $D^\star$ a chance variable whose conditional distribution is given by the decision function from Step 4;
6. Replace the set of utility function $\mathrm{RU}(D^\star)$ by the utility function whose domain is $\mathrm{RD}(D^\star)\backslash\{D^\star\}$, and whose values are the corresponding maximum utility values found in Step 4;
7. Go to Step 1.

T_1			
nt	t_1	t_2	t_3
27.6 (0.00)	32.6 (0.02)	32.9 (0.03)	32.7 (0.02)

Table 2: Mean utilities and standard errors for $D^\star = T_1$ in the third stage of the UCB problem.

4.3 Results

A detailed presentation of the stage-by-stage results appears in Charnes and Shenoy [1996]. Table 2 contains the final results, which indicate that choosing $T_1 = t_2$ is optimal. The stage-by-stage results not shown here indicate optimal choices consistent with the decision-tree solution. However, note that the magnitudes of the standard errors in Table 2 are not small enough to allow us to choose unequivocally the best alternative among t_1, t_2, and t_3. This is an area for further research.

5 Implementation Issues and Further Research

In carrying out the forward Monte Carlo simulation, the main implementation issue is determining the number of iterations, $n(D^\star)$, necessary to obtain sufficient precision of the expected utilities. Note that our method will only compute an approximately optimal strategy. If the precision specified is sufficiently high to distinguish between optimal and non-optimal strategies, then the method will guarantee an optimal strategy. Of course, high precision comes at computational cost. Notice that our method will be unable to detect alternative optimal strategy (if any exist).

The standard errors are readily calculated for the forward Monte Carlo method because each iteration through a sampling path is stochastically independent of the other iterations by the nature of the random numbers generated for the simulation. For the results reported in Charnes and Shenoy [1996], one way to decrease the standard errors is simply to increase the number of iterations. However, this is often much less efficient than using other methods of variance reduction available [Bratley et al. 1987; Ripley 1987], and even with today's fast computers, the savings in computation time may be significant for some problems. Part of future research is to explore the use of variance reduction techniques for the type of problems considered here.

References

Bratley, P., B. L. Fox, and L. E. Schrage (1987), *A Guide to Simulation*, 2nd edition, Springer-Verlag, NY.

Call, H. J. and W. A. Miller (1990), "A comparison of approaches and implementations for automating decision analysis," *Reliability Engineering and System Safety*, 30, 115–162.

Charnes, J. M., and P. P. Shenoy (1996). "A forward Monte Carlo method for solving influence diagrams using local computation," School of Business, University of Kansas Working Paper No. 273 (available by anonymous ftp from `ftp.bschool.ukans.edu/data/pub/pshenoy/wp273.ps`).

Covaliu, Z. and R. M. Oliver (1996), "Representation and solution of decision problems using sequential decision diagrams," *Management Science*, to appear.

Cowell, R. G. (1994), "Decision networks: A new formulation for multistage decision problems," Research Report No. 132, University College, London.

Ezawa, K. J. (1986), "Efficient evaluation of influence diagrams," Ph.D. dissertation, Department of Engineering-Economics Systems, Stanford University, Stanford, CA.

Fung, R. M. and R. D. Shachter (1990), "Contingent influence diagrams," unpublished manuscript, Department of Engineering-Economic Systems, Stanford University, Stanford, CA.

Goutis, C. (1995), "A graphical method for solving a decision analysis problem," *IEEE Transactions on Systems, Man, & Cybernetics*, 25(8), 1181–1193.

Henrion, M. (1988), "Propagating uncertainty in Bayesian networks by probabilistic logic sampling," in J. F. Lemmer and L. Kanal, N. (eds.), *Uncertainty in Artificial Intelligence*, 2, 149–164, North-Holland, Amsterdam.

Hertz, D. B. (1964), "Risk analysis in capital investment," *Harvard Business Review.* Reprinted in *Harvard Business Review*, 1979, 169–181.

Hertz, D. B. and H. Thomas (1983), *Risk Analysis and Its Applications*, John Wiley & Sons, New York, NY.

Hertz, D. B. and H. Thomas (1984), *Practical Risk Analysis*, John Wiley & Sons, New York, NY.

Hillier, F. S. And G. J. Lieberman (1990), *Introduction to Operations Research*, fifth edition, McGraw-Hill, NY.

Howard, R. A. (1962) "The used car buyer," in R. A. Howard and J. E. Matheson (eds.) (1984), *The Principles and Applications of Decision Analysis, 2*, 719–762, Strategic Decisions Group, Menlo Park, CA.

Howard, R. A. (1971), "Proximal decision analysis," *Management Science*, 19(9), 507–541.

Howard, R. A. and J. E. Matheson (1983), *Readings on the Principles and Applications of Decision Analysis, 1*, Strategic Decisions Group, Menlo Park, CA.

Howard, R. A. and J. E. Matheson (1984), "Influence diagrams," in R. A. Howard and J. E. Matheson (eds.), *Readings on the Principles and Applications of Decision Analysis, 2*, 719–762, Strategic Decisions Group, Menlo Park, CA.

Jensen, F., F. V. Jensen and S. L. Dittmer (1994), "From influence diagrams to junction trees," in R. L. Mantaras and D. Poole (eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Tenth Conference*, 367–373, Morgan Kaufmann, San Francisco, CA.

Keefer, D. L. (1994), "Certainty equivalents for three-point discrete-distribution approximations," *Management Science*, 40(6), 760–773.

Kenley, C. R. (1986), "Influence diagram models with continous variables," PhD dissertation, Department of Engineering-Economic Systems, Stanford University, Stanford, CA.

Law, A. M. (1983), "Statistical analysis of simulation output data," *Operations Research*, 31, 983–1029.

Miller III, A. C., M. W. Merkhofer, R. A. Howard, J. E. Matheson and T. R. Rice (1976), "Development of decision aids for decision analysis," Final Technical Report DO # 27742 Stanford Research Institute, Menlo Park, CA.

Miller III, A. C. and T. R. Rice (1983), "Discrete approximations of probability distributions," *Management Science*, 29, 352–362.

Ndilikilikesha, P. C. (1992), "A study of influence diagrams and their generalizations," PhD thesis, School of Business, University of Kansas, Lawrence, KS.

Ndilikilikesha, P. C. (1994), "Potential influence diagrams," *International Journal of Approximate Reasoning*, 10(3), 251–285.

Olmsted, S. M. (1983), "On representing and solving decision problems," Ph.D. thesis, Department of Engineering-Economic Systems, Stanford University, Stanford, CA.

Pearl, J. (1987), "Evidential reasoning using stochastic simulation of causal models," *Artificial Intelligence*, 33, 245–257.

Poland III, W. B. (1994), "Decision analysis with

continuous and discrete variables: A mixture distribution approach," PhD thesis, Stanford University, Department of Engineering-Economic Systems, Stanford, CA.

Qi, R. and D. Poole (1995), "A new method for influence diagram evaluation," *Computational Intelligence*, 11(3), 498528.

Qi, R., L. Zhang and D. Poole (1994), "Solving asymmetric decision problems with influence diagrams," in R. L. Mantaras and D. Poole (eds.), *Uncertainty in Artificial Intelligence: Proceedings of the Tenth Conference*, 491–497, Morgan Kaufmann, San Francisco, CA.

Ripley, B. D. (1987), *Stochastic Simulation*, Wiley, NY.

Shachter, R. D. (1986), "Evaluating influence diagrams," *Operations Research*, 34(6), 871–882.

Shachter, R. D. and C. R. Kenley (1989), "Gaussian influence diagrams," *Management Science*, 35(5), 527–550.

Shachter, R. D. and M. A. Peot (1990), "Simulation approaches to general probabilistic inference on belief networks," in M. Henrion, R. D. Shachter, L. N. Kanal and J. F. Lemmer (eds.), *Uncertainty in Artificial Intelligence*, 5, 221–231, North-Holland, Amsterdam.

Shachter, R. D. and M. A. Peot (1992), "Decision making using probabilistic inference methods," in D. Dubois, M. P. Wellman, B. D'Ambrosio and P. Smets (eds.), *Uncertainty in Artificial Intelligence*, 276–283, Morgan Kaufmann, San Mateo, CA.

Shenoy, P. P. (1992), "Valuation-based systems for Bayesian decision analysis," *Operations Research*, 40(3), 463–484.

Shenoy, P. P. (1993), "Valuation network representation and solution of asymmetric decision problems," Working Paper No. 246, School of Business, University of Kansas, Lawrence, KS.

Shenoy, P. P. (1996), "Representing and solving asymmetric decision problems using valuation networks," in D. Fisher and H.-J. Lenz (eds.), *Learning from Data: Artificial Intelligence and Statistics V, Lecture Notes in Statistics*, 99–108, Springer-Verlag, Berlin.

Smith, A. F. M. (1991), "Bayesian computational methods," *Philosophical Transactions of the Royal Society of London*, series A, 337, 337–386.

Smith, J. E. (1993), "Moment methods for decision analysis," *Management Science*, 39(3), 340–358.

Smith, J. E., S. Holtzman and J. E. Matheson (1993), "Structuring conditional relationships in influence diagrams," *Operations Research*, 41(2), 280–297.

Smith, J. Q. (1989), "Influence diagrams for Bayesian decision analysis," *European Journal of Operational Research*, 40, 363–376.

Tatman, J. A. (1986), "Decision processes in influence diagrams: Formulation and analysis," Ph.D. Thesis, Department of Engineering-Economic Systems, Stanford University, Stanford, CA.

Tatman, J. A. and R. D. Shachter (1990), "Dynamic programming and influence diagrams," *IEEE Transactions on Systems, Man, and Cybernetics*, 20(2), 365–379.

Zhang, N. L., R. Qi and D. Poole (1994), "A computational theory of decision networks," *International Journal of Approximate Reasoning*, 11(2), 83–158.