Sampling Methods for Action Selection in Influence Diagrams

Luis E. Ortiz
Computer Science Department
Brown University
Box 1910
Providence, RI 02912 USA
leo@cs.brown.edu

Leslie Pack Kaelbling
Artificial Intelligence Laboratory
Massachusetts Institute of Technology
545 Technology Square
Cambridge, MA 02139 USA
lpk@ai.mit.edu

Abstract

Sampling has become an important strategy for inference in belief networks. It can also be applied to the problem of selecting actions in influence diagrams. In this paper, we present methods with probabilistic guarantees of selecting a near-optimal action. We establish bounds on the number of samples required for the traditional method of estimating the utilities of the actions, then go on to extend the traditional method based on ideas from sequential analysis, generating a method requiring fewer samples. Finally, we exploit the intuition that equally good value estimates for each action are not required, to develop a heuristic method that achieves major reductions in required sample size. The heuristic method is validated empirically.

Introduction

The problem of decision-making involves the selection of an *optimal strategy*. A strategy determines how we should act based on observations or available information about the variables of the system relevant to the decision problem. Posed in the framework of decision theory, an optimal strategy is one that maximizes our utility. The utility defines our notion of value associated with the execution of actions and the states of the system. The states result from the combination of the state of the individual variables in the system. In the case of decision-making under uncertainty, we are uncertain about both the state of the system and the result of the actions we take. We express this uncertainty as probabilities. Therefore, in this context an *optimal strategy* is one that *maximizes our expected utility*.

In this paper our main interest is in decision problems under uncertainty formulated as *influence diagrams (ID)*. An *influence diagram* is a graphical model that provides a compact representation of (1) the probability distribution governing the states, (2) the structural strategy model representing how we make decisions, and (3) a utility model defining our notion of value associated with actions and states. We study the problem of selecting an optimal strategy in an influence diagram, concentrating on the case in which there is only one decision to be made. This is because we can decompose the problem of multiple decisions into many subproblems involving single decisions (i.e., by using the technique presented by Charnes & Shenoy (1999)). We note that we can apply methods developed to solve IDs of this kind to obtain methods to solve finite-horizon Markov decision processes (MDPs) and partially observable Markov decision processes (POMDPs) expressed as dynamic Bayesian networks (DBNs) (i.e., by modifying the technique presented by Kearns, Mansour, & Ng (1999)).

The problem of strategy selection involves the subproblem of selecting an *optimal action*, from the set of action choices available for that decision, *for each possible observation* available at the time of making the decision. Therefore, we want to select the action that maximizes the expected utility for each observation. One way to do action selection is to compute, exactly or approximately, the probabilities of the sub-states of the system directly relevant to our utility in order to evaluate the expected utility or *value* of each action. A sub-state is formed from the state of a subset of variables in the system. We believe this approach fails to take advantage of an important intuition: it only matters which action is best. Therefore, the problem of action selection is primarily one of comparing the values of the actions. We combine this with the intuition that actions that are close to optimal are also good. In this paper, we present methods for action selection in IDs that take advantage of these intuitions to make major gains in efficiency.

Notation

Before we present the definition of the ID model, we introduce some notation used throughout the paper. We denote one-dimensional random variables by capital letters and denote multi-dimensional random variables by bold capital letters. For instance, we denote a multi-dimensional random variable by $\boldsymbol{X}$ and denote all its components by $(X_1, \ldots, X_n)$ where X_i is the i^{th} one-dimensional random variable. We use small letters to denote assignments to random variables. For instance, $\boldsymbol{X} = \boldsymbol{x}$ means that for each component X_i of $\boldsymbol{X}$, $X_i = x_i$. We also denote by capital letters the nodes in a graph. We denote by $Pa(Y)$ the parents of node Y in a directed graph.

We now introduce notation that will become useful during the description of the methods presented in this paper. For any function h with variables $\boldsymbol{X}$ and $\boldsymbol{Z}$, the expression

$$h(\boldsymbol{X}, \boldsymbol{Z})|_{\boldsymbol{Z}=\boldsymbol{z}}$$

stands for a function f' over variables $\boldsymbol{X}$ that results from setting the values of $\boldsymbol{Z}$ in h with assignment $\boldsymbol{z}$ while letting the values for $\boldsymbol{X}$ remain unassigned. In other words,

$$f'(\boldsymbol{X}) = h(\boldsymbol{X}, \boldsymbol{Z})|_{\boldsymbol{Z}=\boldsymbol{z}} = h(\boldsymbol{X}, \boldsymbol{Z} = \boldsymbol{z}).$$

The notation $\boldsymbol{Z} = (\boldsymbol{S}, \boldsymbol{S'})$ means that the variable $\boldsymbol{Z}$ is formed by all the variables that form $\boldsymbol{S}$ and $\boldsymbol{S'}$. That is, $\boldsymbol{Z} = (Z_1, \ldots, Z_{n'}) = (S_1, \ldots, S_{n_1}, S'_1, \ldots, S'_{n_2}) = (\boldsymbol{S}, \boldsymbol{S'})$, where $n' = n_1 + n_2$. Note that we are assuming that the set of variables forming $\boldsymbol{S}$ and those forming $\boldsymbol{S'}$ are disjoint. The notation $\boldsymbol{Z} \sim f$ means that the random variable $\boldsymbol{Z}$ is distributed according to probability distribution f. We denote a sequence of samples from $\boldsymbol{Z}$ by $\boldsymbol{z}^{(1)}, \boldsymbol{z}^{(2)}, \ldots$, where $\boldsymbol{z}^{(i)}$ is the i^{th} sample. In this paper, we assume that the samples are *independent*.

Definitions

An influence diagram (ID) is a graphical model for decision-making (See Jensen (1996) for additional information and references). It consists of a directed acyclic graph along with a structural strategy model, a probabilistic model and a utility model. The graph represents the decomposition used to compactly define the different models. Figure 1 shows an example of a general graphical representation of an ID. The vertices of the graph consist of three types of nodes: decision nodes, chance nodes and utility nodes. Decision nodes are square and represent the decisions or action choices in the decision problem. Chance nodes are circular and represent the variables of the system relevant to the decision problem. Utility nodes are diamonds and represent the utility associated with actions and *states*. A *state* is an assignment to the variables associated with the chance nodes of the ID.

Structural strategy model The structural strategy model defines locally the form of a decision rule for each decision node A_i. This rule is a function of (a subset of) the information available at the time of making that decision, which is contained in its parents $\mathrm{Pa}(A_i)$ in the graph, the decision nodes that are predecessors of decision node A_i in the graph and their respective parents. The example ID of Figure 1 has only one decision node. Denote a strategy for our example model by π, the *state space* or set of possible assignments for the parents of the action node by $\Omega_{\mathrm{Pa}(A)}$ and the set of possible actions Ω_A. Then, a policy $\pi : \Omega_{\mathrm{Pa}(A)} \rightarrow \Omega_A$.

Probability model The probability model compactly defines the joint probability distribution of the relevant variables given the actions taken using a Bayesian network (BN) (See Jensen (1996) for additional information and references). The model defines locally a conditional probability distribution $P(X_i \mid \mathrm{Pa}(X_i))$ for each variable X_i given its parents $\mathrm{Pa}(X_i)$ in the graph. This defines the following joint probability distribution over the n variables of the system, given that a particular action $\boldsymbol{a}$ is taken:

$$P(X_1, \ldots, X_n \mid \boldsymbol{A} = \boldsymbol{a}) = \textstyle\prod_{i=1}^{n} P(X_i \mid \mathrm{Pa}(X_i))|_{\boldsymbol{A}=\boldsymbol{a}}\,.$$

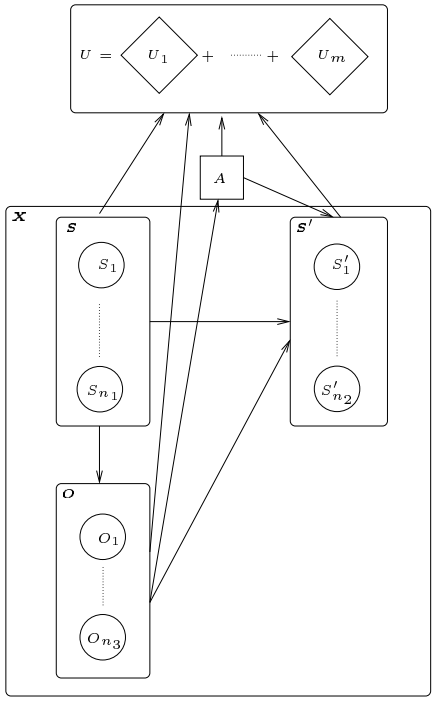

Figure 1: General structure of ID we consider.

In our example ID, $\boldsymbol{X} = (\boldsymbol{S}, \boldsymbol{S'}, \boldsymbol{O})$ and, since there is only one decision node, we can express $P(\boldsymbol{X} \mid \boldsymbol{A} = \boldsymbol{a})$ as

$$\begin{aligned} P(\boldsymbol{X} \mid A = a) &= P(\boldsymbol{S}, \boldsymbol{S'}, \boldsymbol{O} \mid A = a) \\ &= P(\boldsymbol{S})P(\boldsymbol{S'} \mid \boldsymbol{S}, \boldsymbol{O}, A = a)P(\boldsymbol{O} \mid \boldsymbol{S}), \end{aligned}$$

where

$$P(\boldsymbol{S}) = \textstyle\prod_{i=1}^{n_1} P(S_i \mid \mathrm{Pa}(S_i)), \quad (1)$$

$$P(\boldsymbol{S'} \mid \boldsymbol{S}, \boldsymbol{O}, A = a) = \textstyle\prod_{i=1}^{n_2} P(S'_i \mid \mathrm{Pa}(S'_i))|_{A=a}, \quad (2)$$

$$P(\boldsymbol{O} \mid \boldsymbol{S}) = \textstyle\prod_{i=1}^{n_3} P(O_i \mid \mathrm{Pa}(O_i)). \quad (3)$$

Utility model Finally, the utility model defines the utility associated with actions resulting from the decisions made and states of the variables in the system. The total utility function U is the sum of local utility functions associated with each utility node. For each utility node U_i, the utility function provides a utility value as a function of its parents $\mathrm{Pa}(U_i)$ in the graph. The total utility can be expressed as

$$U(\boldsymbol{X}, \boldsymbol{A}) = \textstyle\sum_{i=1}^{m} U_i(\mathrm{Pa}(U_i)). \quad (4)$$

Note that we are using the label of the utility node to also denote the utility function associated with it.

In this paper we assume that the variables and the decisions are discrete and the local utilities are bounded. In addition, we concentrate on IDs with one decision node and the general structure shown in Figure 1. The results in this paper are still valid for more general structural decompositions of the probability distribution. We use the structure given by the ID in the figure to simplify the presentation. Also, the results allow random utility functions.

Value of a strategy The *value V^π of a strategy π* is the expected utility of the strategy:

$$\begin{aligned} V^\pi &= \sum_{\boldsymbol{X}} P(\boldsymbol{X} \mid A = \pi(\boldsymbol{O}))U(\boldsymbol{X}, A = \pi(\boldsymbol{O})) \\ &= \sum_{\boldsymbol{O}} \sum_{\boldsymbol{S}} \sum_{\boldsymbol{S}'} P(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{O} \mid A = \pi(\boldsymbol{O})) \\ &\qquad U(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{O} \mid A = \pi(\boldsymbol{O})). \end{aligned}$$

The optimal strategy π^* is that which maximizes V^π over all π. We denote the value of the optimal strategy by V^*.

Note that we can decompose this maximization into maximizations over the set of actions for each observation. For each assignment to the observations $\boldsymbol{o}$, we define the *value of an action* a by

$$\begin{aligned} V_{\boldsymbol{o}}(a) = \sum_{\boldsymbol{S}} \sum_{\boldsymbol{S}'} P(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{O} = \boldsymbol{o} \mid A = a) \\ U(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{O} = \boldsymbol{o} \mid A = a). \quad (5) \end{aligned}$$

Hence, the value of a strategy is $V^\pi = \sum_{\boldsymbol{O}} V_{\boldsymbol{O}}(\pi(\boldsymbol{O}))$. Note that this is not the traditional definition of the value of an action. We discuss below why we do not use the traditional definition.

If we denote by $a^* = \pi^*(\boldsymbol{o})$ the action that maximizes $V_{\boldsymbol{o}}(a)$ over all actions a, then the value of the optimal strategy is $V^* = \sum_{\boldsymbol{O}} V_{\boldsymbol{O}}(\pi^*(\boldsymbol{O})) = \sum_{\boldsymbol{O}} \max_a V_{\boldsymbol{O}}(a)$. Hence, the problem of strategy selection reduces to that of action selection for each observation.

Exact methods exist for computing the optimal strategy in an ID (See Charnes & Shenoy (1999) and Jensen (1996) for short descriptions and a list of references). However, this problem is hard in general. In this paper, we concentrate on obtaining approximations to the optimal strategy with certain guarantees. Our objective is to find policies that are close to optimal with high probability. That is, for a given accuracy parameter ϵ^* and confidence parameter δ^*, we want to obtain a strategy $\hat{\pi}$ such that $V^* - V^{\hat{\pi}} < \epsilon^*$ with probability at least $1-\delta^*$. Note that given the decomposition described above, if we obtain actions for each observation such that their value is *sufficiently* close to optimal with *sufficiently* high probability, then we obtain a near-optimal strategy with high probability. That is, let l be the number of possible assignments to the observations. If for each observation $\boldsymbol{o}$ we select action $\hat{a}$ such that $V_{\boldsymbol{o}}(a^*) - V_{\boldsymbol{o}}(\hat{a}) < 2\epsilon$ with probability at least $1-\delta$, where $\epsilon = \epsilon^*/(2l)$ and $\delta = \delta^*/l$, then we obtain a strategy that is within ϵ^* of the optimal with probability at least $1-\delta^*$. Therefore, we concentrate on finding a *good* action for each observation.

Typically the value of an action is defined as the *conditional* expected utility of the action *given* an assignment of the observations. If we denote this value by $V(a \mid \boldsymbol{o})$, we can express the value of a policy as $V^\pi = \sum_{\boldsymbol{O}} P(\boldsymbol{O}) V(\pi(\boldsymbol{O}) \mid \boldsymbol{O})$. We do not use this definition because it is harder to obtain estimates for $V(a \mid \boldsymbol{o})$ with guaranteed confidence bounds than it is to obtain estimates for $V_{\boldsymbol{o}}(a)$.

Multiple Comparisons with the Best: Results

There are two important results from the field of *multiple comparisons* and in particular from the field of *multiple comparisons with the best* that we take advantage of in this paper. These results are based on the work of Hsu (1981) (See Hsu (1996) for more information). Before we present the results we introduce the following notation: denote $x^+ = \max(x, 0)$ and $-x^- = \min(0, x)$. The first result is known as *Hsu's single-bound lemma*, which is presented as Lemma 1 by Matejcik & Nelson (1995).

Lemma 1 *Let $\mu_{(1)} \leq \mu_{(2)} \leq \cdots \leq \mu_{(k)}$ be the (unknown) ordered performance parameters of k systems, and let $\hat{\mu}_{(1)}, \hat{\mu}_{(2)}, \ldots, \hat{\mu}_{(k)}$ be any estimators of the parameters. If*

$$\begin{aligned} Pr\{\hat{\mu}_{(k)} - \hat{\mu}_{(i)} - (\mu_{(k)} - \mu_{(i)}) > -w, i = 1, \ldots, k-1\} \\ = 1 - \alpha, \quad (6) \end{aligned}$$

then

$$\begin{aligned} Pr\{\mu_i - \max_{j \neq i} \mu_j \in [-(\hat{\mu}_i - \max_{j \neq i} \hat{\mu}_j - w)^-, \\ (\hat{\mu}_i - \max_{j \neq i} \hat{\mu}_j + w)^+], \text{ for all } i\} \geq 1 - \alpha. \quad (7) \end{aligned}$$

If we replace the $=$ in (6) *with $\geq$, then* (7) *still holds.*

In our context, we let for each action a, the true value $\mu_a = V_{\boldsymbol{o}}(a)$ and the estimate $\hat{\mu}_a = \hat{V}_{\boldsymbol{o}}(a)$. Also, the i^{th} smallest true value corresponds to $\mu_{(i)}$. That is, if $V_{\boldsymbol{o}}(a_1) \leq V_{\boldsymbol{o}}(a_2) \leq \cdots \leq V_{\boldsymbol{o}}(a_k)$, then for all i, $\mu_{(i)} = V_{\boldsymbol{o}}(a_i)$. Note that in practice, we do not know which action has the largest value. In order to apply Hsu's single-bound lemma, we obtain the bound $Pr\{\hat{\mu}_j - \hat{\mu}_i - (\mu_j - \mu_i) > -w, \text{ for all } i \neq j\} \geq 1 - \alpha$, for each action j, individually. This implies that $Pr\{\hat{\mu}_{(k)} - \hat{\mu}_{(i)} - (\mu_{(k)} - \mu_{(i)}) > -w, i = 1, \ldots, k-1\} \geq 1 - \alpha$, which allow us to apply the lemma. Figure 2 graphically describes this practical interpretation of the lemma. For each action i, individually, the upper bounds on the true differences, drawn on the left-hand side, $V_{\boldsymbol{o}}(i) - V_{\boldsymbol{o}}(j) < \hat{V}_{\boldsymbol{o}}(i) - \hat{V}_{\boldsymbol{o}}(j) + w$, for each $j \neq i$, hold simultaneously with probability at least $1 - \alpha$. The confidence intervals, drawn on the right-hand side, $V_{\boldsymbol{o}}(i) - \max_{j \neq i} V_{\boldsymbol{o}}(j) \in [-(\hat{V}_{\boldsymbol{o}}(i) - \max_{j \neq i} \hat{V}_{\boldsymbol{o}}(j) - w)^-, (\hat{V}_{\boldsymbol{o}}(i) - \max_{j \neq i} \hat{V}_{\boldsymbol{o}}(j) + w)^+]$, for each action i, hold simultaneously with probability at least $1 - \alpha$.

The second result allows us to assess joint confidence intervals on the difference between the value of each action from the value of the best action when we have estimates of the differences between value of each pair of actions with different degrees of accuracy. The result is known as *Hsu's multiple-bound lemma*. It is presented as Lemma 2 by Matejcik & Nelson (1995), and credited to Chang & Hsu (1992).

Lemma 2 *Let $\mu_{(1)} \leq \mu_{(2)} \leq \cdots \leq \mu_{(k)}$ be the (unknown) ordered performance parameters of k systems. Let T_{ij} be a point estimator of the parameter $\mu_i - \mu_j$. If for each i individually*

$$Pr\{T_{ij} - (\mu_i - \mu_j) > -w_{ij}, \text{for all } j \neq i\} = 1 - \alpha, \quad (8)$$

then we can make the joint probability statement

$$Pr\{\mu_i - \max_{j \neq i} \mu_j \in [D_i^-, D_i^+], \text{for all } i\} \geq 1 - \alpha, \quad (9)$$

where $D_i^+ = (\min_{j \neq i}[T_{ij} + w_{ij}])^+$, $\mathcal{G} = \{l : D_l^+ > 0\}$, and

$$D_i^- = \begin{cases} 0 & \text{if } \mathcal{G} = \{i\} \\ -(\min_{j \in \mathcal{G}, j \neq i}[-T_{ji} - w_{ji}])^- & \text{otherwise.} \end{cases}$$

If we replace the $=$ in (8) *with $\geq$, then* (9) *still holds.*

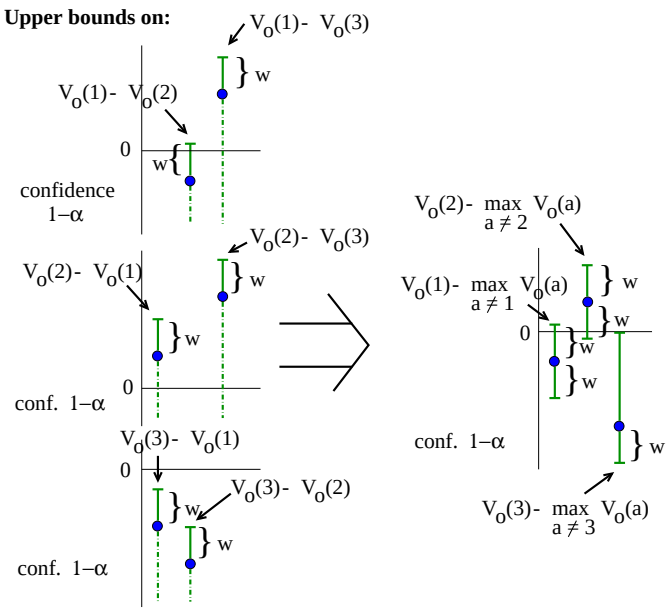

Figure 2: Graphical description for practical application of Hsu's single-bound lemma. Note that the "lower bounds" on the left-hand side are $-\infty$.

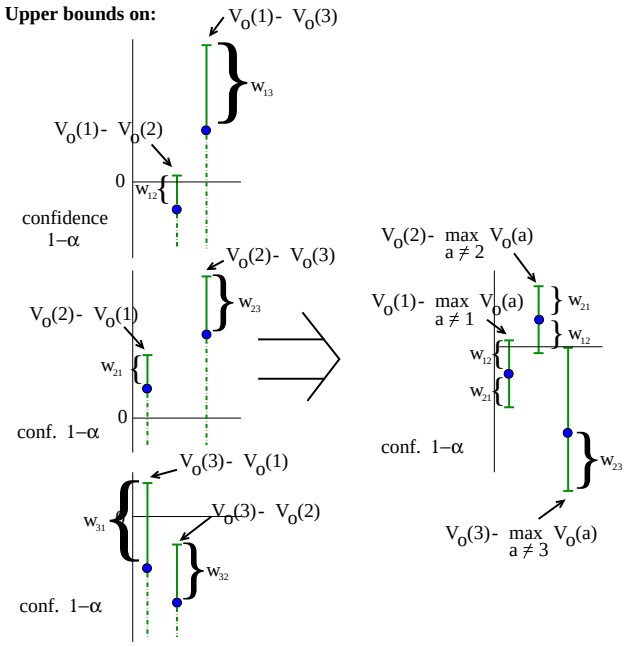

Figure 3: Graphical description of Hsu's multiple-bound lemma. Note that the "lower bounds" on the left-hand side are $-\infty$.

Figure 3 presents a graphical description of this lemma. Let, for all actions i, D_i^- and D_i^+, be as defined in Hsu's multiple-bound lemma, with $\mu_i = V_{\boldsymbol{o}}(i)$ and for all $j \neq i$, $T_{ij} = \hat{V}_{\boldsymbol{o}}(i) - \hat{V}_{\boldsymbol{o}}(j)$. For each action i, individually, the upper bounds on the true differences, drawn on the left-hand side, $V_{\boldsymbol{o}}(i) - V_{\boldsymbol{o}}(j) < T_{ij} + w_{ij}$, for each $j \neq i$, hold simultaneously with probability at least $1-\alpha$. The confidence intervals, drawn on the right-hand side, $V_{\boldsymbol{o}}(i) - \max_{j \neq i} V_{\boldsymbol{o}}(j) \in [D_i^-, D_i^+]$, for each action i, hold simultaneously with probability at least $1-\alpha$. Also, in this example, $\mathcal{G} = \{1, 2\}$. In our context, $\mathcal{G}$ is the set of all the actions that could potentially be the best with probability at least $1-\alpha$. That is, for each action a in $\mathcal{G}$, the upper bound D_a^+ on the difference of the true value of action a and the best of *all* the other actions, including those in $\mathcal{G}$, is positive.

Estimation-based methods

One approach to selecting the best action is to obtain estimates of $V_{\boldsymbol{o}}(a)$ for each a by sampling, using the probability model of the ID conditioned on a, then select the action with the largest estimated value.

We can apply the idea of *importance sampling* (See Geweke (1989) and the references therein) to this estimation problem by using the probability distribution defined by the ID as *the importance function* or *sampling distribution*. This is essentially the same idea as *likelihood-weighting* in the context of probabilistic inference in Bayesian networks (Shachter & Peot, 1989; Fung & Chang, 1989). We present this method in the context of our example ID.

First, we present definitions that will allow us to rewrite $V_{\boldsymbol{o}}(a)$ more clearly. First, let $\boldsymbol{Z} = (\boldsymbol{S}, \boldsymbol{S}')$. Define the *target function* (in our case, the *weighted utilities*)

$$\begin{aligned} g_{a,\boldsymbol{o}}(\boldsymbol{Z}) &= g_{a,\boldsymbol{o}}(\boldsymbol{S}, \boldsymbol{S}') \\ &= P(\boldsymbol{S})P(\boldsymbol{S}' \mid \boldsymbol{S}, \boldsymbol{O} = \boldsymbol{o}, A = a) \cdot \\ &\quad P(\boldsymbol{O} = \boldsymbol{o} \mid \boldsymbol{S})U(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{O} = \boldsymbol{o}, A = a). \end{aligned}$$

Note that $V_{\boldsymbol{o}}(a) = \sum_{\boldsymbol{Z}} g_{a,\boldsymbol{o}}(\boldsymbol{Z})$. Define the *importance function* as

$$f_{a,\boldsymbol{o}}(\boldsymbol{Z}) = P(\boldsymbol{S})P(\boldsymbol{S}' \mid \boldsymbol{S}, \boldsymbol{O} = \boldsymbol{o}, A = a). \quad (10)$$

Define the *weight function* $\omega_{a,\boldsymbol{o}}(\boldsymbol{Z}) = g_{a,\boldsymbol{o}}(\boldsymbol{Z}) / f_{a,\boldsymbol{o}}(\boldsymbol{Z})$. Note that in this case,

$$\omega_{a,\boldsymbol{o}}(\boldsymbol{Z}) = P(\boldsymbol{O} = \boldsymbol{o} \mid \boldsymbol{S})U(\boldsymbol{S}, \boldsymbol{S}', \boldsymbol{O} = \boldsymbol{o}, A = a). \quad (11)$$

Finally, note that $V_{\boldsymbol{o}}(a) = \sum_{\boldsymbol{Z}} f_{a,\boldsymbol{o}}(\boldsymbol{Z})(g_{a,\boldsymbol{o}}(\boldsymbol{Z}) / f_{a,\boldsymbol{o}}(\boldsymbol{Z}))$. The idea of the sampling methods described in this section is to obtain independent samples according to $f_{a,\boldsymbol{o}}$, use those samples to estimate the value of the actions, and finally select an approximately optimal action by taking the action with largest value estimate. Denote the *weight of a sample* $\boldsymbol{z}^{(i)}$ from $\boldsymbol{Z} \sim f_{a,\boldsymbol{o}}$ as $\omega_{a,\boldsymbol{o}}^{(i)} = \omega_{a,\boldsymbol{o}}(\boldsymbol{z}^{(i)})$. Then an unbiased estimate of $V_{\boldsymbol{o}}(a)$ is $\hat{V}_{\boldsymbol{o}}(a) = \frac{1}{N_{a,\boldsymbol{o}}} \sum_{i=1}^{N_{a,\boldsymbol{o}}} \omega_{a,\boldsymbol{o}}^{(i)}$.

Traditional Method

We can obtain an estimate of $V_{\boldsymbol{o}}(a)$ using the straightforward method presented in Algorithm 1; it requires parameters $N_{a,\boldsymbol{o}}$ that will be defined in Theorem 1.

This is the traditional sampling-based method used for action selection. However, we are unaware of any result regarding the number of samples needed to obtain a near-optimal strategy with high probability using this method.

Algorithm 1 Traditional Method

1. Obtain independent samples $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(N_{a,\boldsymbol{o}})}$ from $\boldsymbol{Z} \sim f_{a,\boldsymbol{o}}$.
2. Compute the weights $\omega_{a,\boldsymbol{o}}^{(1)}, \ldots, \omega_{a,\boldsymbol{o}}^{(N_{a,\boldsymbol{o}})}$.
3. Output $\hat{V}_{\boldsymbol{o}}(a) =$ average of the weights.

Theorem 1 *If for each possible action $i = 1, \ldots, k$, we estimate $V_{\boldsymbol{o}}(i)$ using the traditional method, the weight function satisfies $l_{i,\boldsymbol{o}} \leq \omega_{i,\boldsymbol{o}}(\boldsymbol{Z}) \leq u_{i,\boldsymbol{o}}$, and the estimate uses*

$$N_{i,\boldsymbol{o}} = \left\lceil \frac{(u_{i,\boldsymbol{o}} - l_{i,\boldsymbol{o}})^2}{2\epsilon^2} \ln \frac{k}{\delta} \right\rceil$$

samples, then the action with the largest value estimate has a true value that is within 2ϵ of the optimal with probability at least $1 - \delta$.

Proof sketch. The proof goes in three basic steps. First, we apply *Hoeffding bounds* (Hoeffding, 1963) to obtain a bound on the probability that each estimate deviates from its true mean by some amount ϵ. Then, we apply the *Bonferroni inequality (Union bound)* to obtain joint bounds on the probability that the difference of each estimate from all the others deviates from the true difference by 2ϵ. Finally, we apply Hsu's single bound lemma to obtain our result.

Note that we can compute $l_{i,\boldsymbol{o}}$ and $u_{i,\boldsymbol{o}}$ efficiently from information local to each node in the graph. Assuming that we have non-negative utilities, we can let

$$\begin{aligned} u_{i,\boldsymbol{o}} &= \left[\prod_{j=1}^{n_3} \max_{\mathrm{Pa}(O_j)} P(O_j \mid \mathrm{Pa}(O_j))|_{\boldsymbol{O}=\boldsymbol{o}}\right] \\ &\quad \bullet \left[\sum_{j=1}^{m} \max_{\mathrm{Pa}(U_j)} U_j(\mathrm{Pa}(U_j))|_{\boldsymbol{O}=\boldsymbol{o},A=i}\right], \quad (12) \\ l_{i,\boldsymbol{o}} &= \left[\prod_{j=1}^{n_3} \min_{\mathrm{Pa}(O_j)} P(O_j \mid \mathrm{Pa}(O_j))|_{\boldsymbol{O}=\boldsymbol{o}}\right] \\ &\quad \bullet \left[\sum_{j=1}^{m} \min_{\mathrm{Pa}(U_j)} U_j(\mathrm{Pa}(U_j))|_{\boldsymbol{O}=\boldsymbol{o},A=i}\right]. \quad (13) \end{aligned}$$

However, these bounds can be very loose.

Sequential Method

The sequential method tries to reduce the number of samples needed by the traditional method, using ideas from sequential analysis. The idea is to first obtain an estimate of the variance and then use it to compute the number of samples needed to estimate the mean. The method, presented in Algorithm 2, requires parameters $N'_{a,\boldsymbol{o}}$ and $N''_{a,\boldsymbol{o}}$ that will be defined in Theorem 2.

Note that given the sequential nature of the method, the total number of samples is now a random variable. We also note that while multi-stage procedures of this kind are commonly used in the statistical literature, we are only aware of results based on restricting assumptions on the distribution of the random variables (i.e., parametric families like normal and binomial distributions) (Bechhofer, Santner, & Goldsman, 1995).

Theorem 2 *If, for each possible action $i = 1, \ldots, k$, we estimate $V_{\boldsymbol{o}}(i)$ using the sequential method, the weight function satisfies $l_{i,\boldsymbol{o}} \leq \omega_{i,\boldsymbol{o}}(\boldsymbol{Z}) \leq u_{i,\boldsymbol{o}}$, $\sigma^2_{i,\boldsymbol{o}} = Var[\omega_{i,\boldsymbol{o}}(\boldsymbol{Z})]$,*

$$N'_{i,\boldsymbol{o}} = \left\lceil \frac{(u_{i,\boldsymbol{o}} - l_{i,\boldsymbol{o}})^{4/3}}{2\, 2^{2/3}\epsilon^{4/3}} \ln \frac{2k}{\delta} \right\rceil,$$

and

$$N''_{i,\boldsymbol{o}}(\hat{\sigma}^2_{i,\boldsymbol{o}}) = \left\lceil \left(\frac{2\hat{\sigma}^2_{i,\boldsymbol{o}} + 2(u_{i,\boldsymbol{o}} - l_{i,\boldsymbol{o}})\epsilon/3}{\epsilon^2} + 2^{1/3} \frac{(u_{i,\boldsymbol{o}} - l_{i,\boldsymbol{o}})^{4/3}}{\epsilon^{4/3}} \right) \ln \frac{2k}{\delta} \right\rceil,$$

then the action with the largest value estimate has a true value that is within 2ϵ of the optimal with probability at least $1 - \delta$. Also,

$$\begin{aligned} N_{i,\boldsymbol{o}} &< \left(\frac{2\sigma^2_{i,\boldsymbol{o}} + 2(u_{i,\boldsymbol{o}} - l_{i,\boldsymbol{o}})\epsilon/3}{\epsilon^2} + \frac{5}{2^{2/3}} \frac{(u_{i,\boldsymbol{o}} - l_{i,\boldsymbol{o}})^{4/3}}{\epsilon^{4/3}} \right) \ln \frac{2k}{\delta} + 1 \\ &= O\left(\max\left(\frac{\sigma^2_{i,\boldsymbol{o}}}{\epsilon^2}, \frac{(u_{i,\boldsymbol{o}} - l_{i,\boldsymbol{o}})^{4/3}}{\epsilon^{4/3}} \right) \ln \frac{k}{\delta} \right), \end{aligned}$$

with probability at least $1 - \delta/(2k)$, and

$$\begin{aligned} E[N_{i,\boldsymbol{o}}] &= 2N'_{i,\boldsymbol{o}} + N''_{i,\boldsymbol{o}}(\sigma^2_{i,\boldsymbol{o}}) \\ &= O\left(\max\left(\frac{\sigma^2_{i,\boldsymbol{o}}}{\epsilon^2}, \frac{(u_{i,\boldsymbol{o}} - l_{i,\boldsymbol{o}})^{4/3}}{\epsilon^{4/3}} \right) \ln \frac{k}{\delta} \right). \end{aligned}$$

Algorithm 2 Sequential Method

1. Obtain independent samples $\boldsymbol{z}^{(1)}, \ldots, \boldsymbol{z}^{(2N'_{a,\boldsymbol{o}})}$ from $\boldsymbol{Z} \sim f_{a,\boldsymbol{o}}$.
2. Compute the weights $\omega_{a,\boldsymbol{o}}^{(1)}, \ldots, \omega_{a,\boldsymbol{o}}^{(2N'_{a,\boldsymbol{o}})}$.
3. For $j = 1, \ldots, N'_{a,\boldsymbol{o}}$, let $y_j = (\omega_{a,\boldsymbol{o}}^{(2j-1)} - \omega_{a,\boldsymbol{o}}^{(2j)})^2/2$.
4. Compute $\hat{\sigma}^2_{a,\boldsymbol{o}} =$ average of y_j's.
5. Let $N_{a,\boldsymbol{o}} = 2N'_{a,\boldsymbol{o}} + N''_{a,\boldsymbol{o}}(\hat{\sigma}^2_{a,\boldsymbol{o}})$.
6. Obtain $N''_{a,\boldsymbol{o}}(\hat{\sigma}^2_{a,\boldsymbol{o}})$ new independent samples $\boldsymbol{z}^{(2N'_{a,\boldsymbol{o}}+1)}, \ldots, \boldsymbol{z}^{(N_{a,\boldsymbol{o}})}$ from $\boldsymbol{Z} \sim f_{a,\boldsymbol{o}}$.
7. Compute the new weights $\omega_{a,\boldsymbol{o}}^{(2N'_{a,\boldsymbol{o}}+1)}, \ldots, \omega_{a,\boldsymbol{o}}^{(N_{a,\boldsymbol{o}})}$.
8. Output $\hat{V}_{\boldsymbol{o}}(a) =$ average of the new weights.

Proof sketch. The only difference from the proof of Theorem 1 is the first step. Instead of using Hoeffding bounds to bound the probability that each estimate deviates from its true mean, we use a combination of *Bernstein's inequality* (as presented by Devroye, Györfi, & Lugosi (1996) and credited to Bernstein (1946)) and Hoeffding bounds as follows. We first use the Hoeffding bound to bound the probability that the estimate of the variance after taking some number of samples $2N'$ deviates from the true variance by some amount ϵ'. We then use Bernstein's inequality to bound the probability that the estimate we obtain after taking some number of samples N'' deviates from its true mean by

ϵ *given* that the true variance is no larger than our estimate of the variance plus ϵ'. We then find the value of ϵ' (in terms of ϵ) that minimizes the total number of samples $N'' + 2N'$. The results on the number of samples follow by substituting the minimizing ϵ' back into the expressions for N'' and N'. Steps 2 and 3 are as in Theorem 1.

The sequential method is particularly more effective than the traditional method when $\sigma^2_{i,\boldsymbol{o}} \ll (u_{i,\boldsymbol{o}} - l_{i,\boldsymbol{o}})^2$.

Comparison-based Method

Using the results from MCB, we can compute simultaneous or joint confidence intervals on the difference between the value of $V_{\boldsymbol{o}}(a)$ and the best of all the others for all actions a. Therefore, MCB allows us to select the best action choice or an action with value close to it, within a confidence level.

In the previous section we presented methods that require that we have estimates with the same precision in order to select a good action. Hsu's multiple-bound lemma applies when we do not have estimates of $V_{\boldsymbol{o}}(a)$ for each a with the same precision. Based on this result, we propose the method presented in Algorithm 3 for action selection.

Algorithm 3 Comparison-based Method

1. Obtain an *initial number of samples* for each action a.
2. Compute *MCB confidence intervals* on the difference in value of each action from the best of the other actions using those samples.

while *not able to select a good action with high certainty* **do**

3(a). Obtain *additional samples*.

3(b). Recompute MCB confidence intervals using total samples so far.

We compute the MCB confidence intervals heuristically. To do this, we approximate the precisions that satisfy the conditions required by Hsu's multiple-bound lemma (Equation 8) using Hoeffding bounds (Hoeffding, 1963). Using this approach, for each pair of actions i and j, and values $l_{ij,\boldsymbol{o}}$ and $u_{ij,\boldsymbol{o}}$ such that $l_{ij,\boldsymbol{o}} \leq \omega_{i,\boldsymbol{o}}(\boldsymbol{Z}) \leq u_{ij,\boldsymbol{o}}$ and $l_{ij,\boldsymbol{o}} \leq \omega_{j,\boldsymbol{o}}(\boldsymbol{Z}) \leq u_{ij,\boldsymbol{o}}$, we approximate w_{ij} as

$$w_{ij} = (u_{ij,\boldsymbol{o}} - l_{ij,\boldsymbol{o}}) \sqrt{\frac{1}{2}\left(\frac{1}{N_{i,\boldsymbol{o}}} + \frac{1}{N_{j,\boldsymbol{o}}}\right) \ln \frac{k-1}{\delta}}, \quad (14)$$

where $N_{i,\boldsymbol{o}}$ is the number of samples taken for action i thus far. We then use these approximate precisions and the value-difference estimates to compute the MCB confidence intervals (as specified by Equation 9). There are alternative ways of heuristically approximating the precisions but, in this paper, we use the one above for simplicity.

Once we compute the intervals, the stopping condition is as follows. If at least one of the lower bounds of the MCB confidence intervals is greater than -2ϵ, then we stop and select the action that attains this lower bound. Otherwise, we continue taking additional samples.

We define the value of *initial number of samples* in our experiments as 40. When taking additional samples, we use a sampling schedule that is somewhat selective in that it takes more samples from more promising actions as suggested by the MCB confidence intervals. We find the action whose corresponding MCB confidence interval has an upper bound greater than 0 (i.e., from the set $\mathcal{G}$ as defined in Hsu's multiple bound lemma) and whose lower bound is the largest. We take 40 additional samples from this action and 10 from all the others. We understand that these sample sizes are very arbitrary. Potentially, other setting of these sample sizes can be more effective but we did not try to optimize them for our experiments. Algorithm 4 presents a detailed description of the instance of the method we used in the experiments.

Algorithm 4 Algorithmic description of the instance of the comparison-based method used in the experiments.

for each observation $\boldsymbol{o}$ **do**
  $l \leftarrow 1$
  for each action $i = 1, \ldots, k$ **do**
    Compute $u_{i,\boldsymbol{o}}$ and $l_{i,\boldsymbol{o}}$ using equations 12 and 13, respectively.
    $D_i^- \leftarrow -\infty$; $N_{i,\boldsymbol{o}}^{(l)} \leftarrow 40$; $N_{i,\boldsymbol{o}} \leftarrow 0$; $\hat{V}_{\boldsymbol{o}}(i) \leftarrow 0$.
  for each pair of actions (i, j), $i \neq j$ **do**
    $u_{ij,\boldsymbol{o}} \leftarrow \max(u_{i,\boldsymbol{o}}, u_{j,\boldsymbol{o}})$; $l_{ij,\boldsymbol{o}} \leftarrow \max(l_{i,\boldsymbol{o}}, l_{j,\boldsymbol{o}})$.
  while there is no action i such that $D_i^- > -2\epsilon$ **do**
    for each action i **do**
      Obtain $N_{i,\boldsymbol{o}}^{(l)}$ samples $\boldsymbol{z}^{(N_{i,\boldsymbol{o}}+1)}, \ldots, \boldsymbol{z}^{(N_{i,\boldsymbol{o}}+N_{i,\boldsymbol{o}}^{(l)})}$ from $\boldsymbol{Z} \sim f_{i,\boldsymbol{o}}$, as in equation 10.
      Compute weights $\omega_{i,\boldsymbol{o}}^{(N_{i,\boldsymbol{o}}+1)}, \ldots, \omega_{i,\boldsymbol{o}}^{(N_{i,\boldsymbol{o}}+N_{i,\boldsymbol{o}}^{(l)})}$.
      $\hat{V}_{\boldsymbol{o}}(i) \leftarrow (N_{i,\boldsymbol{o}} \hat{V}_{\boldsymbol{o}}(i) + \sum_{j=1}^{N_{i,\boldsymbol{o}}^{(l)}} \omega_{i,\boldsymbol{o}}^{(N_{i,\boldsymbol{o}}+j)}) / (N_{i,\boldsymbol{o}} + N_{i,\boldsymbol{o}}^{(l)})$.
      $N_{i,\boldsymbol{o}} \leftarrow N_{i,\boldsymbol{o}} + N_{i,\boldsymbol{o}}^{(l)}$.
    for each pair of actions (i, j), $i \neq j$ **do**
      $T_{ij} \leftarrow \hat{V}_{\boldsymbol{o}}(i) - \hat{V}_{\boldsymbol{o}}(j)$; $T_{ji} \leftarrow -T_{ij}$.
      Compute w_{ij} using equation 14; $w_{ji} \leftarrow w_{ij}$.
    for each action i **do**
      Compute D_i^+, $\mathcal{G}$, and D_i^- using Hsu's multiple-bound lemma.
    for each action i **do**
      if $D_i^- == \max_{j \in \mathcal{G}} D_j^-$ **then** $N_{i,\boldsymbol{o}}^{(l+1)} \leftarrow 40$
      else $N_{i,\boldsymbol{o}}^{(l+1)} \leftarrow 10$.
    $l \leftarrow l + 1$.
  $\hat{\pi}(\boldsymbol{o}) \leftarrow \text{argmax}_i \, D_i^-$.

Although this method may seem well-grounded, we are not convinced that the bounds hold rigorously. The precisions are correct if the samples obtained so far for each action are independent. However, this might not be the case, since the number of samples gathered on each round depends on a property of the previous set of samples (that is, that the lower-bound condition did not hold). It is not yet clear to us whether the fact that the *number* of samples depends on the values of the samples implies that the samples must be considered dependent.

Related Work

Charnes & Shenoy (1999) present a Monte Carlo method similar to our "traditional method." One difference is that they use a heuristic stopping rule based on a normal approximation (i.e., the estimates have an *asymptotically* normal distribution). Their method takes samples until all the estimates achieve a required standard error to provide the correct confidence interval on each value under the assumption that the estimates are normally distributed and the estimate of the variance is equal to the true variance. They do not give bounds on the number of samples needed to obtain a near-optimal action with the required confidence. We refer the reader to Charnes & Shenoy (1999) for a short description and references on other similar Monte Carlo methods for IDs.

Bielza, Müller, & Insua (1999) present a method based on Markov-Chain Monte Carlo (MCMC) for solving IDs. Although their primary motivation is to handle continuous action spaces, their method also applies to discrete action spaces. Because of the typical complications in analyzing MCMC methods, they do not provide bounds on the number of samples needed. Instead, they use a heuristic stopping rule which does not guarantee the selection of a near-optimal action. Other MCMC-based methods have been proposed (See Bielza, Müller, & Insua (1999) for more information).

Empirical results

We tried the different methods on a simple made-up ID. Given space restrictions we only describe it briefly (See Ortiz (2000) for details). Figure 4 gives a graphical representation of the ID for the *computer mouse problem*. The idea is to select an optimal strategy of whether to *buy* a new mouse ($A = 1$), *upgrade* the operating system ($A = 2$), or take *no action* ($A = 3$). The observation is whether the mouse pointer is working ($MP_t = 1$) or not ($MP_t = 0$). The variables of the problem are the status of the operating system (OS), the status of the driver (D), the status of the mouse hardware (MH), and the status of the mouse pointer (MP), all at the current and future time (subscripted by t and $t+1$). The variables are all binary.

The probabilistic model encodes the following information about the system. The mouse is old and somewhat unreliable. The operating system is reliable. It is very likely that the mouse pointer will not work if either the driver or the mouse hardware has failed. Table 1 shows the utility function $U(MP_{t+1}, A)$ and the values of the actions and observations $V_{\boldsymbol{O}}(A)$ computed using an exact method. From Table 1 we conclude that the optimal strategy is: buy a new mouse ($A = 1$) if the mouse pointer is not working ($MP_t = 0$); take no action ($A = 3$) if the mouse pointer is working ($MP_t = 1$). This strategy has value 26.50.

Table 2 presents our results on the effectiveness of the sampling methods for this problem. We set our final desired accuracy for the output strategy to $\epsilon^* = 5$ and confidence level $\delta^* = 0.05$. This leads to the individual accuracy $2\epsilon = 2.5$ and confidence level $\delta = 0.025$ for each subproblem. We executed the sequential method and the comparison-based method 100 times. The comparison-based method produces major reductions in the number of samples. When we observe the mouse pointer not working, The comparison-based method always selects the optimal action of buying a new mouse. When we observe the mouse pointer working, The comparison-based method failed to select the optimal action of *taking no action* 4 times out of the 100. In those cases, it selected the next-to-optimal action of upgrading the operating system ($A = 2$). This action is within our accuracy requirements since the difference in value with respect to the optimal action is 0.91.

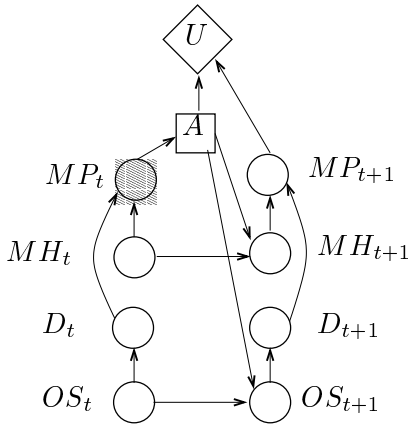

Figure 4: Graphical representation of the ID for the computer mouse problem.

	U		V	
	MP_{t+1}		MP_t	
A	0	1	0	1
1	0	40	**18.20**	6.60
2	5	45	7.54	7.39
3	10	50	10.57	**8.30**

Table 1: This table presents the utility function and the (exact) value of actions and observations for the computer mouse problem.

The comparison-based method is highly effective in cases where there is a clear optimal action to take. For instance, in the computer mouse problem, buying a new mouse when we observe the mouse not working is clearly the best option. The differences in value between the optimal action and the rest are not as large as when we observe the mouse working.

In this problem, the results for the sequential method should not fully discourage us from its use, because the variances are still relatively large. We have seen major reductions in problems where the variance is significantly smaller than the square of the range of the variable whose mean we are estimating.

Summary and Conclusion

The methods presented in this paper are an alternative to exact methods. While the running time of exact methods depends on aspects of the structural decomposition of the

A	MP_t	Method		
		Traditional	Sequential	Comp-based
1	0	2403	3802 (188)	335 (151)
2	0	3007	2266 (142)	115 (37)
3	0	3679	2426 (129)	118 (39)
1	1	2213	2508 (178)	521 (216)
2	1	2794	2969 (201)	695 (421)
3	1	3443	3468 (202)	1361 (560)
Total		17539	17438 (434)	**3145** (809)

Table 2: Number of samples taken by the different methods for each action and observation. For the sequential and the comparison-based methods, the table displays the average number of samples over 100 runs. The values in parenthesis are the sample standard deviations.

ID, the running time of the methods presented in this paper depends primarily on the range of the weight functions, the variance of the value estimators and the amount of separation between the value of the best action and that of the rest (in addition to the natural dependency on the number of action choices, and the precision and confidence parameters). In some cases, we can know in advance whether they will be faster or not. The methods presented in this paper can be a useful alternative in those cases where exact methods are intractable. How useful depends on the particular characteristics of the problem.

Sampling is a promising tool for action selection. Our empirical results on a small ID suggest that sampling methods for action selection are more effective when they take advantage of the intuition that action selection is primarily a comparison task. We look forward to experimenting with IDs large enough that sampling methods are the only potentially efficient alternative. Also, our work leads to the study of adaptive sampling as a way to improve the effectiveness of sampling methods (Ortiz & Kaelbling, 2000).

Acknowledgments We would like to thank Constantine Gatsonis for suggesting the MCB literature; Eli Upfal, Milos Hauskrecht, Thomas Hofmann, Thomas Dean and Kee Eung Kim for useful discussions and suggestions; and the anonymous reviewers for their useful comments. Our implementations use the *Bayes Net Toolbox for Matlab* (Murphy, 1999), for which we thank Kevin Murphy. Luis E. Ortiz was supported in part by an NSF Graduate Fellowship and by NSF IGERT award SBR 9870676. Leslie Pack Kaelbling was supported in part by a grant from NTT and by DARPA Contract #DABT 63-99-1-0012.

References

Bechhofer, R. E.; Santner, T. J.; and Goldsman, D. M. 1995. *Design and analysis of experiments for statistical selection, screening and multiple comparisons*. Wiley.

Bernstein, S. 1946. *The Theory of Probabilities*. Gastehizdat Publishing House, Moscow.

Bielza, C.; Müller, P.; and Insua, D. R. 1999. Monte Carlo methods for decision analysis with applications to influence diagrams. *Management Science.* Forthcoming.

Chang, J. Y., and Hsu, J. C. 1992. Optimal designs for multiple comparisons with the best. *Journal of Statistical Planning and Inference* 30:45–62.

Charnes, J. M., and Shenoy, P. P. 1999. A forward Monte Carlo method for solving Influence diagrams using local computation. School of Business, University of Kansas, Working Paper No. 273.

Devroye, L.; Györfi, L.; and Lugosi, G. 1996. *A Probabilistic Theory of Pattern Recognition*. Springer.

Fung, R., and Chang, K.-C. 1989. Weighting and integrating evidence for stochastic simulation in Bayesian networks. In *Proceedings of the Fifth Workshop on Uncertainty in Artificial Intelligence*, 112–117.

Geweke, J. 1989. Bayesian inference in econometric models using Monte Carlo integration. *Econometrica* 57(6):1317–1339.

Hoeffding, W. 1963. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association* 58(301):13–30.

Hsu, J. C. 1981. Simultaneous confidence intervals for all distances from the "best". *Annals of Statistics* 9(5):1026–1034.

Hsu, J. C. 1996. *Multiple Comparisons: Theory and Methods*. Chapman and Hall.

Jensen, F. V. 1996. *An Introduction to Bayesian Networks*. UCL Press.

Kearns, M.; Mansour, Y.; and Ng, A. Y. 1999. A sparse sampling algorithm for near-optimal planning in large Markov decision processes. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, 1324–1331. Menlo Park, Calif.: International Joint Conference on Artificial Intelligence, Inc.

Matejcik, F. J., and Nelson, B. L. 1995. Two-stage multiple comparisons with the best for computer simulation. *Operations Research* 43(4):633–640.

Murphy, K. P. 1999. Bayes net toolbox for Matlab. Available from `http://www.cs.berkeley.edu/~murphyk/Bayes/bnt.html`.

Ortiz, L. E., and Kaelbling, L. P. 2000. Adaptive importance sampling for estimation in structured domains. Under review.

Ortiz, L. E. 2000. Selecting approximately-optimal actions in complex structured domains. Technical Report CS-00-05, Computer Science Department, Brown University.

Shachter, R. D., and Peot, M. A. 1989. Simulation approaches to general probabilistic inference on belief networks. In *Proceedings of the Fifth Workshop on Uncertainty in Artificial Intelligence*, 311–318.