Corresponding Author: Dr. wenhui liao, Ph.D.

Corresponding Author's Institution: Rensselaer Polytechnic Institute

First Author: wenhui liao, Ph.D.

Order of Authors: wenhui liao, Ph.D.; Qiang Ji, PHD

Manuscript Region of Origin:

Abstract: In an influence diagram (ID), value-of-information (VOI) is defined as the difference between the maximum expected utilities with and without knowing the outcome of an uncertainty variable prior to making a decision. It is widely used as a sensitivity analysis technique to rate the usefulness of various information sources, and to decide whether pieces of evidence are worth acquisition before actually using them. However, due to the exponential time complexity of exactly computing VOI of multiple information sources, decision analysts and expert-system designers
focus on the myopic VOI, which assumes observing only one information source, even though several information sources are available. In this paper, we present an approximate algorithm to compute non-myopic VOI efficiently by utilizing the central-limit theorem. The proposed method overcomes several limitations in the existing work. In addition, a partitioning procedure based on the d-separation concept is proposed to further improve the computational complexity of the proposed algorithm. Both the experiments with synthetic data and the experiments with real data from a real-world application demonstrate that the

proposed algorithm can approximate the true non-myopic VOI well even with a small number of observations. The accuracy and efficiency of the algorithm makes it feasible in various applications where efficiently evaluating a large amount of information sources is necessary.

Suggested Reviewers:

Opposed Reviewers:

Efficient Non-myopic Value-of-Information Computation For Influence Diagrams

Wenhui Liao and Qiang Ji

{liaow, jiq}@rpi.edu
Department of Electrical, Computer and Systems Engineering
Rensselaer Polytechnic Institute, Troy, NY 12180-3590, USA

Abstract

In an influence diagram (ID), value-of-information (VOI) is defined as the difference between the maximum expected utilities with and without knowing the outcome of an uncertainty variable prior to making a decision. It is widely used as a sensitivity analysis technique to rate the usefulness of various information sources, and to decide whether pieces of evidence are worth acquisition before actually using them. However, due to the exponential time complexity of exactly computing VOI of multiple information sources, decision analysts and expert-system designers focus on the myopic VOI, which assumes observing only one information source, even though several information sources are available. In this paper, we present an approximate algorithm to compute non-myopic VOI efficiently by utilizing the central-limit theorem. The proposed method overcomes several limitations in the existing work. In addition, a partitioning procedure based on the d-separation concept is proposed to further improve the computational complexity of the proposed algorithm. Both the experiments with synthetic data and the experiments with real data from a real-world application demonstrate that the proposed algorithm can approximate the true non-myopic VOI well even with a small number of observations. The accuracy and efficiency of the algorithm makes it feasible in various applications where efficiently evaluating a large amount of information sources is necessary.

Key words: Value-of-Information, Influence Diagrams, Decision Making, Central-Limit Theorem, Stress Modeling

1 Introduction

In a wide range of decision-making problems, a common scenario is that a decision maker must decide whether some information is worth collecting, and

what information should be acquired first given several information sources available. Each set of information sources is usually evaluated by value-of-information (VOI). VOI is a quantitative measure of the value of knowing the outcome of the information source(s) prior to making a decision. In other words, it is quantified as the difference in value achievable with or without knowing the information sources in a decision-making problem.

Generally, VOI analysis is one of the most useful sensitivity analysis techniques for decision analysis (Raiffa, 1968; Shachter, 1999). VOI analysis evaluates the benefit of collecting additional information in a specific decision-making context (Yokota and Thompson, 2004). General VOI analyses usually require three key elements: 1) A set of available actions and information collection strategies; 2) A model connecting the actions and the related uncertainty variables within the context of the decision; and 3) values for the decision outcomes. The methods of VOI analysis could be quite different when different models are used.

In this paper, we consider VOI analysis in decision problems modeled by Influence Diagrams. Influence diagrams were introduced by Howard and Matheson in 1981 (Howard and Matheson, 1981) and have been widely used as a knowledge representation framework to facilitate decision making and probability inference under uncertainty. An ID uses a graphical representation to capture the three diverse sources of knowledge in decision making: conditional relationships about how events influence each other in the decision domain; informational relationships about what action sequences are feasible in any given set of circumstances; and functional relationships about how desirable the consequences are (Pearl, 1988). An ID can systematically model all the relevant random variables and decision variables in a compact graphical model.

In the past several years, a few methods have been proposed to compute VOI in IDs. (Ezawa, 1998) introduces some basic concepts about VOI and evidence propagation in IDs. (Dittmer and Jensen, 1997) present a method for calculating myopic VOI in IDs based on the strong junction tree framework (Jensen et al., 1994). (Shachter, 1999) further improves this method by enhancing the strong junction tree as well as developing methods for reusing the original tree in order to perform multiple VOI calculations. (Zhang et al., 1993) present an algorithm to speed up the VOI computation by making use of the intermediate computation results, which are obtained when computing the optimal expected value of the original ID without the observations from the information sources. Instead of computing VOI directly, (Poh and Horvitz, 1996) describe a procedure to identify a partial order over variables in terms of their VOIs based on the topological relationships among variables in the ID. However, all these papers only focus on computing myopic VOI, which is based on two assumptions: 1) "No competition:" each information source is evaluated in isolation, as if it were the only source available for the entire decision; 2)

2

"One-step horizon:" the decision maker will act immediately after consulting the source (Pearl, 1988). These assumptions result in a myopic policy: every time, the decision maker evaluates the VOI of each information source one by one, and chooses the one with the largest VOI. Then the observations are collected from the selected information sources, the probabilities are updated, and all the remaining information sources are to be reevaluated again, and a similar procedure repeats.

Obviously, the assumptions are not always reasonable in some decision circumstances. Usually, the decision maker will not act after acquiring only one information source. Also, although a single information source may have low VOI and is not worth acquisition compared to its cost, several information sources used together may have high VOI compared to their combined cost. In this case, by only evaluating myopic VOI, the conclusion will be not to collect such information, which is not optimal since its usage together with other information sources can lead to high value for the decision maker. Therefore, given these limitations in myopic VOI, it is necessary to compute non-myopic VOI.

Non-myopic VOI respects the fact that the decision maker may observe more than one piece of information before acting, thus requires the consideration of any possible ordered sequence of observations given a set of information sources. Unfortunately, the number of the sequences grows exponentially as the number of available information sources increases, and thus it is usually too cumbersome to compute non-myopic VOI for any practical use, and this is why the before mentioned work only focuses on myopic VOI. Given these facts, an approximate computation of non-myopic VOI is necessary to make it feasible in practical applications. To the best of our knowledge, (Heckerman et al., 1993) are the only ones who proposed a solution to this problem. In their approach, the central-limit theorem is applied to approximately compute non-myopic VOI in a special type of ID for the diagnosis problem, where only one decision node exists. Certain assumptions are required in their method: 1) all the random nodes and decision nodes in the ID are required to be binary; 2) the information sources are conditionally independent from each other given the hypothesis node, which is the node associated with the decision node and utility node.

Motivated by the method of Heckerman et al, we extend this method to more general cases [1]: 1) all the random nodes can have multiple states and the decision node can have multiple rules (alternatives); 2) the information sources can be dependent given the hypothesis node; and 3) the ID can have a more general structure. **But same as Heckerman et al's method, we only discuss the VOI computation in terms of IDs that have only one**

---

[1] A brief version of this extension can be found in (Liao and Ji, 2006).

3

decision node. This decision node shares only one utility node with another chance node. With the proposed algorithm, non-myopic VOI can be efficiently approximated. In order to validate the performance of the proposed algorithm, we not only perform the experiments based on the synthetic data for various types of IDs, but also provide a real-world application with real data.

Because of the efficiency and accuracy of the proposed method, we believe that it can be widely used to choose the optimal set of available information sources for a wide range of applications. No matter what selection strategies people use to choose an optimal set, such as greedy approaches, heuristic searching algorithms, or brute-force methods, the proposed method can be utilized to evaluate any information set efficiently in order to speed up the selection procedure.

The following sections are organized as follows. Section 2 presents a brief introduction to influence diagrams. The detail of the algorithm is described in Section 3. Section 4 discusses the experimental results based on synthetic data. And a real application is demonstrated in Section 5. Finally, Section 6 gives the conclusion and some suggestions for future work.

2   Influence Diagrams

An influence diagram (ID) is a graphical representation of a decision-making problem under uncertainty. Its knowledge representation can be viewed through three hierarchical levels, namely, relational, functional, and numerical. At the relational level, an ID represents the relationships between different variables through an acyclic directed graph consisting of various node types and directed arcs. The functional level specifies the interrelationships between various node types and defines the corresponding conditional probability distributions. Finally, the numerical level specifies the actual numbers associated with the probability distributions and utility values (Diehl and Haimes, 2004).

Specifically, an ID includes three types of nodes: decision, chance (random), and value (utility) nodes. Decision nodes, usually drawn as rectangles, indicate the decisions to be made and their set of possible alternative values. Chance nodes, usually drawn as circles/ellipses, represent uncertain variables that are relevant to the decision problem. They are similar to the nodes in Bayesian networks (Jensen, 2001), and are associated with conditional probability tables (CPTs). Value nodes, usually drawn as diamonds, are associated with utility functions to represent the utility of each possible combination of the outcomes of the parent node. The arcs connecting different types of nodes have different meanings. An arc between two chance nodes represents probabilistic

4

dependence, while an arc from a decision node to a chance node represents functional dependence, which means the actions associated with the decision node affect the outcome of the chance node. An arc between two decision nodes implies time precedence, while an arc from a chance node to a decision node is informational, i.e., it shows which variable will be known to the decision maker before a decision is made (Pearl, 1988). An arc pointing to a utility node represents value influence, which indicates that the parents of the utility node are those that directly affect its utility. Figure 1 illustrates these arcs and gives corresponding interpretations.
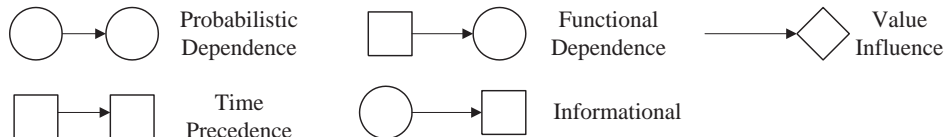

Fig. 1. Interpretations of arcs in an ID, where circles represent chance (random) nodes, rectangles for decision nodes, and diamonds for value (utility) nodes.

Most IDs assume a precedence ordering of the decision nodes. A regular ID assumes that there is a directed path containing all decision nodes; a no-forgetting ID assumes that each decision node and its parents are also parents of the successive decision nodes; and a stepwise decomposable ID assumes that the parents of each decision node divide the ID into two separate fractions. In this paper, we consider IDs that have only one decision node, i.e., ignoring all previous decisions. The goal of ID modeling is to choose an optimal policy that maximizes the overall expected utility. A policy is a sequence of decision rules where each rule corresponds to one decision node. Mathematically, if there is only one decision node in an ID and assuming additive decomposition of the utility functions, the expected utility under a decision rule d given any available evidence e, denoted by $EU(d|e)$, can be defined as follows:

$$EU(d|e) = \sum_{i=1}^{n} \sum_{X_i} p(X_i|e, d)u_i(X_i, d) \tag{1}$$

where u_i is the utility function over the domain $X_i \cup \{D\}$. For example, X_i could be the parents of the utility node that u_i is associated with. To evaluate an ID is to find an optimal policy as well as to compute its optimal expected utility (Shachter, 1986; Shachter and Ndilikilikesha, 1993). More detail about IDs can be found in (Korb and Nicholson, 2003; Jensen, 2001).

Generally, the advantages of an ID can be summarized by its compact and intuitive formulation, its easy numerical assessment, and its effective graphical representation of dependence between variables for modeling decision-making under uncertainty. These benefits make ID a widely used tool to model and solve complex decision problems in recent years.

5

3 Approximate VOI Computation

3.1 Value of Information

The VOI of a set of information sources is defined as the difference between the maximum expected utilities with and without the information sources (Korb and Nicholson, 2003). VOI can be used to rate the usefulness of various information sources and to decide whether pieces of evidence are worth acquisition before actually using the information sources (Pearl, 1988).
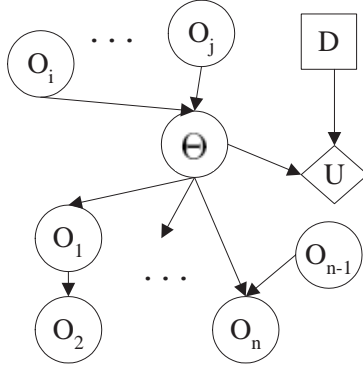

Fig. 2. An ID example for non-myopic VOI computation. Θ is the hypothesis node, D is the decision node, and U is the utility node. O_i represents possible observations from an information source. There could be hidden nodes between Θ and O_i.

We discuss the VOI computation in terms of IDs that have only one decision node. This decision node shares only one utility node with another chance node, as shown in Figure 2. **And the decision node and the chance node are assumed to be independent.** In the ID, the chance node Θ, named as hypothesis node, represents a mutually exclusive and exhaustive set of possible hypotheses θ_1, θ_2, ..., θ_h; the decision node D represents a set of possible alternatives d_1, d_2, ..., d_q; the utility node U represents the utility of the decision maker, which depends on the outcome of Θ and D; and the chance nodes O_1, ..., O_n represent possible observations from all kinds of information sources about the true state of Θ. And each O_i may have multiple states. Let $O = \{O_1, ..., O_n\}$, the VOI of O, $VOI(O)$, w.r.t. the decision node D, can be defined as follows:

$$VOI(O) = EU(O) - EU(\bar{O}) \tag{2}$$

$$EU(O) = \sum_{o \in O} p(o) \max_{d_j \in D} \sum_{\theta_i \in \Theta} p(\theta_i|o) u(\theta_i, d_j) \tag{3}$$

$$EU(\bar{O}) = \max_{d_j \in D} \sum_{\theta_i \in \Theta} p(\theta_i) u(\theta_i, d_j) \tag{4}$$

6

where $u()$ denotes the utility function associated with the utility node U, $EU(O)$ denotes the expected utility to the decision maker if O were observed, while $EU(\bar{O})$ denotes the expected utility to the decision maker without observing O. Here the cost of collecting information from the information sources is not included; thus, the VOI can also be called *perfect VOI* (Heckerman et al., 1993). The *net VOI* is the difference between the perfect VOI and the cost of collecting information (Howard, 1967). Since after calculating the perfect VOI, the computation of the net VOI is just a subtraction of cost, we focus on the perfect VOI in the subsequent sections.

As shown in Equation 2, to compute $VOI(O)$, it is necessary to compute $EU(O)$ and $EU(\bar{O})$ respectively. Obviously, $EU(\bar{O})$ is easier to compute, whereas directly computing $EU(O)$ could be cumbersome. If the decision maker has the option to observe a subset of observations $\{O_1, ..., O_n\}$ and each O_i has m possible values, then there are m^n possible instantiations of the observations in this set. Thus, to compute $EU(O)$, there are m^n inferences to be performed. In other words, the time complexity of computing VOI is exponential. It becomes infeasible to compute VOI(O) when n is not small.

The key to computing $VOI(O)$ efficiently is to compute $EU(O)$, which can be rewritten as follows:

$$
\begin{aligned}
EU(O) &= \sum_{o \in O} p(o) \max_{d_j \in D} \sum_{\theta_i \in \Theta} p(\theta_i | o) u(\theta_i, d_j) \\
&= \sum_{o \in O} \max_{d_j \in D} \sum_{\theta_i \in \Theta} p(o) p(\theta_i | o) u(\theta_i, d_j) \\
&= \sum_{o \in O} \max_{d_j \in D} \sum_{\theta_i \in \Theta} p(\theta_i) p(o | \theta_i) u(\theta_i, d_j)
\end{aligned} \tag{5}
$$

It is noticed that each instantiation of O corresponds to a specific optimal action for the decision node D. We define the decision function $\delta : O \rightarrow D$, which maps an instantiation of O into a decision in D. For example, $\delta(o) = d_k$ indicates when the observation is o, the corresponding optimal decision is d_k, $d_k = \arg\max_{d_j \in D} \sum_{\theta_i \in \Theta} p(\theta_i | o) u(\theta_i, d_j)$. Therefore we can divide all the instantiations of O into several subsets, where the optimal action is the same for those instantiations in the same subset. Specifically, if D has q decision rules, $\{d_1, ..., d_q\}$, all the instantiations of O can be divided into q subsets, o_{d_1}, o_{d_2}, ..., o_{d_q}, where $o_{d_k} = \{o \in O | \delta(o) = d_k\}$. Figure 3 illustrates the relationships between each instantiation and the q subsets. Thus, from Equation 5, EU(O) can be further derived as follows:

$$
EU(O) = \sum_{\theta_i \in \Theta} p(\theta_i) \sum_{k=1}^{q} \sum_{o \in o_{d_k}} p(o | \theta_i) u(\theta_i, d_k) \tag{6}
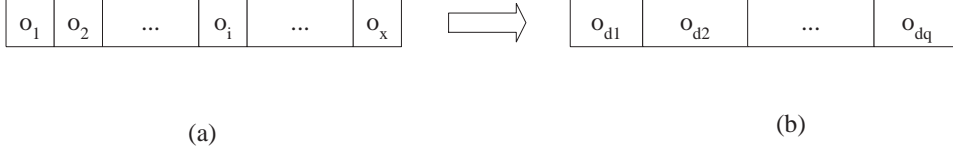$$

7

(a)

(b)

Fig. 3. (a) Each o_i corresponds to an instantiation; (b) All the instantiations can be divided into q subsets, where each instantiation in the set o_{d_i} corresponds to the optimal decision d_i.

In the next several sections, we show how to compute $EU(O)$ efficiently.

3.2 Decision Boundaries

In Equation 6, the difficult part is to compute $\sum_{o \in o_{d_k}} p(o|\theta_i)$ because the size of the set o_{d_k} could be very large based on the previous analysis. In order to compute it efficiently, it is necessary to know how to divide all the instantiations of O into the q subsets. We first focus on the case that Θ has only two states, θ_1, θ_2, and then extend it to the general case in Section 3.4.

Based on the definition, the expected utility of taking the action d_k is $EU(d_k) = p(\theta_1) * u_{1k} + p(\theta_2) * u_{2k}$, where $u_{1k} = u(\theta_1, d_k)$, and $u_{2k} = u(\theta_2, d_k)$. We can sort the index of all the decision rules based on the utility functions, such that $u_{1k} > u_{1j}$ and $u_{2k} < u_{2j}$ for $k < j$. Figure 4 gives an example of the utility function $u(\Theta, D)$. As shown in the figure, as k increases, u_{1k} decreases and u_{2k} increases. If there is an action d_i that cannot be sorted according to this criterion, it is either dominated by another action, or it dominates another action. (If $u(d_i, \Theta)$ is always larger than $u(d_j, \Theta)$, no matter what state of Θ is, we say d_i dominates d_j). Then the dominated action can be removed from the set of possible actions, without changing the optimal policy.
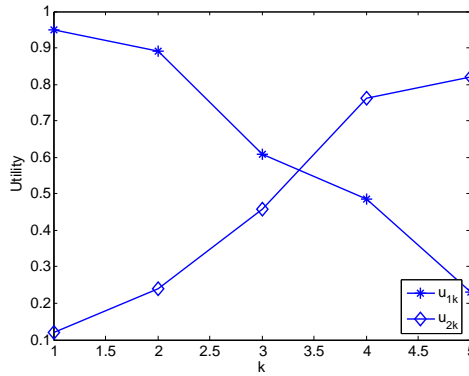

Fig. 4. An example of the utility function $U(\Theta, D)$.

Proposition 1: *Let* $r_{jk} = \frac{u_{2j} - u_{2k}}{u_{1k} - u_{1j} + u_{2j} - u_{2k}}$, $p^*_{kl} = \max_{k < j \leq q} r_{jk}$, *and* $p^*_{ku} = \min_{1 \leq j < k} r_{jk}$,

8

then d_k is the optimal action if and only if $p_{kl}^* \leq p(\theta_1) \leq p_{ku}^*$. In addition, $p_{ql}^* = 0$ and $p_{1u}^* = 1$.(Here k is the index of an action.)

Proof: see Appendix.

Proposition 1 presents that if the probability of Θ being θ_1 is between p_{kl}^* and p_{ku}^*, d_k is the optimal decision. From this, we can further derive Proposition 2.

Proposition 2:

$$\sum_{o \in o_{d_k}} p(o) = p(p_{kl}^* \leq p(\theta_1|o) \leq p_{ku}^*) \tag{7}$$

Proof: see Appendix.

The proof of Proposition 2 establishes Equation 7 by showing that both sides of this equation express the probability that d_k is the optimal decision for θ_1. Based on Proposition 2, we can get the following corollary.

Corollary 1:

$$\sum_{o \in o_{d_k}} p(o|\theta_1) = p(p_{kl}^* \leq p(\theta_1|o) \leq p_{ku}^*|\theta_1) \tag{8}$$

$$\sum_{o \in o_{d_k}} p(o|\theta_2) = p(p_{kl}^* \leq p(\theta_1|o) \leq p_{ku}^*|\theta_2) \tag{9}$$

The equations in Corollary 1 indicate the probability that the decision maker will take the optimal decision d_k after observing new evidence, given the situation that the state of Θ is θ_i before collecting the evidence.

Based on Corollary 1, the problem of computing $\sum_{o \in o_{d_k}} p(o|\theta_i), i = 1, 2$, (from Equation 6) transfers to the problem of computing $p(p_{kl}^* \leq p(\theta_1|o) \leq p_{ku}^*|\theta_i)$, which is the topic of the next section. We will focus on $p(p_{kl}^* \leq p(\theta_1|o) \leq p_{ku}^*|\theta_1)$ only because the procedure of computing $p(p_{kl}^* \leq p(\theta_1|o) \leq p_{ku}^*|\theta_2)$ is similar.

3.3   Approximation with Central-Limit Theorem

3.3.1   A Partitioning Procedure

To compute $p(p_{kl}^* \leq p(\theta_1|o) \leq p_{ku}^*|\theta_1)$, one way is to treat $p(\theta_1|o)$ as a random variable. If the probability density function of this variable is known, it will

be easy to compute $p(p_{kl}^* \leq p(\theta_1|o) \leq p_{ku}^*|\theta_1)$. However, it is hard to get such a probability density function directly. But we notice that $p(p_{kl}^* \leq p(\theta_1|o) \leq p_{ku}^*|\theta_1) = p(\frac{p_{kl}^*}{1-p_{kl}^*} \leq \frac{p(\theta_1|o)}{p(\theta_2|o)} \leq \frac{p_{ku}^*}{1-p_{ku}^*}|\theta_1)$. Based on the transformation property between a random variable and its function (Casella and Berger, 1990), it is straightforward that $p(p_{kl}^* \leq p(\theta_1|o) \leq p_{ku}^*|\theta_1) = p(\frac{p_{kl}^*}{1-p_{kl}^*} \leq \frac{p(\theta_1|o)}{p(\theta_2|o)} \leq \frac{p_{ku}^*}{1-p_{ku}^*}|\theta_1)$.

Let us take a closer look at $\frac{p(\theta_1|o)}{p(\theta_2|o)}$ because it is critical in the approximate algorithm.

If all the O_i nodes are conditionally independent from each other given Θ, based on the chain rule:

$$\frac{p(\theta_1|O)}{p(\theta_2|O)} = \frac{p(O_1|\theta_1)}{p(O_1|\theta_2)}...\frac{p(O_n|\theta_1)}{p(O_n|\theta_2)}\frac{p(\theta_1)}{p(\theta_2)} \tag{10}$$

Usually some O_is may not be conditionally independent given Θ. We will show that $\frac{p(\theta_1|o)}{p(\theta_2|o)}$ is approximately distributed as a log-normal random variable. However, in order to prove it, it is necessary to obtain a format similar to Equation 10 even when O_is are not conditionally independent. We thus propose a partitioning procedure to partition O into several groups based on the principle of d-separation(Pearl, 1988), where the nodes in one group are conditionally independent from the nodes in other groups. This procedure consists of three steps.

(1) Decide whether two nodes, O_i, O_j, are conditionally independent given Θ by exploring the ID structure based on four rules: i) if there is a **directed** path between O_i and O_j without passing Θ, O_i and O_j are dependent; ii) if both O_i and O_j are the ancestors of Θ, O_i and O_j are dependent given Θ; iii) after removing the links to and from Θ from the original ID, if O_i and O_j have common ancestors, or O_i is O_j's ancestor, or vice versa, then O_i and O_j are dependent; and iv) in all the other cases, O_i and O_j are conditionally independent given Θ.
(2) Build an undirected graph to model the relationships between the nodes. In such a graph, each vertex represents an O_i node, and each edge between two vertices indicates that the two corresponding nodes are dependent according to the rules in Step 1.
(3) Partition the graph into disjoint connected subgraphs. A depth first search (DFS) algorithm (Cormen et al., 2002) is used to partition the graph into several *connected components* (disjoint connected subgraphs) so that each component is disconnected from other components. The nodes in each connected component are conditionally independent from the nodes in any other connected components. Therefore, each connected component corresponds to one group.
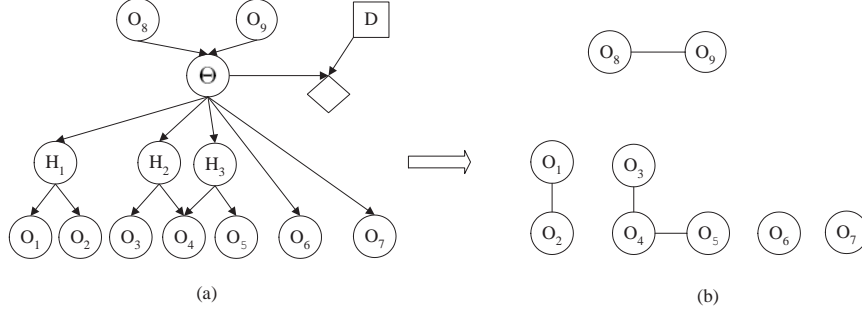
Fig. 5. (a) An ID example; (b) The graph built by the partitioning procedure.

For example, for the ID in Figure 5(a), with the partitioning procedure, the O_i nodes can be divided into 5 groups, $\{O_1, O_2\}$, $\{O_3, O_4, O_5\}$, $\{O_6\}$, $\{O_7\}$, and $\{O_8, O_9\}$. Figure 5(b) shows the graph built by the partitioning procedure.

3.3.2 Central Limit Theorem

Generally, with the partition procedure presented in the previous subsection, O can be automatically divided into several sets, named O^{s_1}, O^{s_2}, ..., O^{s_g}, where g is the overall number of the groups. Thus, Equation 10 can be modified as follows:

$$\frac{p(\theta_1|O)}{p(\theta_2|O)} = \frac{p(O^{s_1}|\theta_1)}{p(O^{s_1}|\theta_2)}...\frac{p(O^{s_g}|\theta_1)}{p(O^{s_g}|\theta_2)}\frac{p(\theta_1)}{p(\theta_2)} \tag{11}$$

$$\Rightarrow ln\frac{p(\theta_1|O)}{p(\theta_2|O)} = \sum_{i=1}^{g} ln\frac{p(O^{s_i}|\theta_1)}{p(O^{s_i}|\theta_2)} + ln\frac{p(\theta_1)}{p(\theta_2)}$$

$$\Rightarrow ln\phi = \sum_{i=1}^{g} w_i + c, \text{ where } \phi = \frac{p(\theta_1|O)}{p(\theta_2|O)}, \ w_i = ln\frac{p(O^{s_i}|\theta_1)}{p(O^{s_i}|\theta_2)}, \ c = ln\frac{p(\theta_1)}{p(\theta_2)}$$

In the above equation, c can be regarded as a constant reflecting the state of Θ before any new observation is obtained and any new decision is taken. Here, we assume $p(\theta_2|O)$, $p(O^{s_i}|\theta_2)$, and $p(\theta_2)$ are not equal to 0.

Let $W = \sum_{i=1}^{g} w_i$ be the sum of w_i. Following (Heckerman et al., 1993), we use the cental-limit theorem to approximate W. The central-limit theorem (Feller, 1971) states that the sum of independent variables approaches a Gaussian distribution when the number of variables becomes large. Also, the expectation and variance of the sum is the sum of the expectation and variance of each individual random variable. Thus, regarding each w_i as an independent variable, W then follows a Gaussian distribution. Then, based on Equation 11, ϕ will be a log-normal distribution. For a random variable X, if $\ln(X)$ has a Gaussian distribution, we say X has a log-normal distribution. The probability density

function is: $p(x) = \frac{1}{S\sqrt{2\pi}x}e^{-(\ln x - M)^2/(2S^2)}$, denoted as $X \sim LogN(M, S^2)$ (Crow and Shimizu, 1988), where M and S are the mean and standard deviation of the variable's logarithm (Balakrishnan and Chen, 1999). In order to assess the parameters (mean and variance) of the log-normal distribution, we need to compute the mean and the variance of each w_i. The computational process is shown as follows.

Assume O^{s_i} has r_i instantiations, $\{o_1^{s_i}, ..., o_{r_i}^{s_i}\}$, where r_i is the product of the number of the states for each node in the group O^{s_i}, e.g., if $O^{s_i} = \{O_1, O_2\}$, and both O_1 and O_2 have three states, then $r_i = 3 * 3 = 9$. Table 1 gives the value and the probability distribution for each w_i:

Table 1
The probability distribution of w_i.

w_i	$p(w_i|\theta_1)$	$p(w_i|\theta_2)$
$ln\frac{p(o_1^{s_i}|\theta_1)}{p(o_1^{s_i}|\theta_2)}$	$p(o_1^{s_i}|\theta_1)$	$p(o_1^{s_i}|\theta_2)$
...	...	...
$ln\frac{p(o_{r_i}^{s_i}|\theta_1)}{p(o_{r_i}^{s_i}|\theta_2)}$	$p(o_{r_i}^{s_i}|\theta_1)$	$p(o_{r_i}^{s_i}|\theta_2)$

Based on the table, the expected value μ, and the variance σ^2 for each w_i can be computed as follows:

$$\mu(w_i|\theta_1) = \sum_{j=1}^{r_i} p(o_j^{s_i}|\theta_1)ln\frac{p(o_j^{s_i}|\theta_1)}{p(o_j^{s_i}|\theta_2)} \tag{12}$$

$$\sigma^2(w_i|\theta_1) = \sum_{j=1}^{r_i} p(o_j^{s_i}|\theta_1)ln^2\frac{p(o_j^{s_i}|\theta_1)}{p(o_j^{s_i}|\theta_2)} - \mu^2(w_i|\theta_1) \tag{13}$$

By the central-limit theorem, the expected value and the variance of W can be obtained by the following equations:

$$\mu(W|\theta_1) = \sum_{i=1}^{g} \mu(w_i|\theta_1) \tag{14}$$

$$\sigma^2(W|\theta_1) = \sum_{i=1}^{g} \sigma^2(w_i|\theta_1) \tag{15}$$

Therefore, based on Equation 11, for $W \sim N(\mu(W|\theta_1), \sigma^2(W|\theta_1))$, we have $\phi \sim LogN(\mu(W|\theta_1) + c, \sigma^2(W|\theta_1))$, where LogN denotes the log-normal distribution. After getting the probability distribution function and the function parameters for ϕ in Equation 11, we are ready to assess the non-myopic VOI.

Before we go to the next section, we first analyze the computational steps involved in computing the parameters for the log-normal distribution, which is the most time-consuming part in the algorithm. Based on Equations 12 and 14, the overall number of the computational steps is $4\sum_{i=1}^{g} r_i + 2g$. We will show that this number is much smaller than the overall number of the computational steps in the exact computational method during the algorithm analysis in Section 3.5.

3.3.3 Approximate Non-myopic Value-of-Information

Based on Proposition 1 in Section 3.2, we know that d_k is the optimal action with the probability $p(p_{kl}^* \leq p(\theta_1|o) \leq p_{ku}^*)$, which is equivalent to $p(\frac{p_{kl}^*}{1-p_{kl}^*} \leq \phi \leq \frac{p_{ku}^*}{1-p_{ku}^*})$ as shown in Section 3.3.1. Let $\phi_{kl}^* = \frac{p_{kl}^*}{1-p_{kl}^*}$, and $\phi_{ku}^* = \frac{p_{ku}^*}{1-p_{ku}^*}$, thus, d_k is the optimal decision if and only if $\phi_{kl}^* \leq \phi \leq \phi_{ku}^*$. Then, based on Corollary 1 in Section 3.2, the following equation stands:

$$\sum_{o \in o_{d_k}} p(o|\theta_1) = p(\phi_{kl}^* \leq \phi \leq \phi_{ku}^*|\theta_1) \tag{16}$$

Furthermore, from Section 3.3.2, we know that $\phi \sim LogN(\mu(W|\theta_1)+c, \sigma^2(W|\theta_1))$, thus,

$$p(\phi_{kl}^* \leq \phi \leq \phi_{ku}^*|\theta_1) = \frac{1}{\sigma(W|\theta_1)\sqrt{2\pi}x} \int_{\phi_{kl}^*}^{\phi_{ku}^*} e^{\frac{-(\ln x - \mu(W|\theta_1)-c)^2}{2\sigma^2(W|\theta_1)}} dx \tag{17}$$

$p(\phi_{kl}^* \leq \phi \leq \phi_{ku}^*|\theta_2)$ can be computed in the same way by replacing θ_1 with θ_2 in the previous equations.

Therefore, VOI can be approximated by combining Equations 2, 6, 16, and 17. Figure 6 shows the key equations of the algorithm when Θ has only two states. In summary, to approximate $VOI(O)$ efficiently, the key is to compute $EU(O)$, which leads to an approximation of $\sum_{o \in o_{d_k}} p(o|\theta_1)$ with the log-normal distribution by exploiting the central-limit theorem and the decision boundaries.
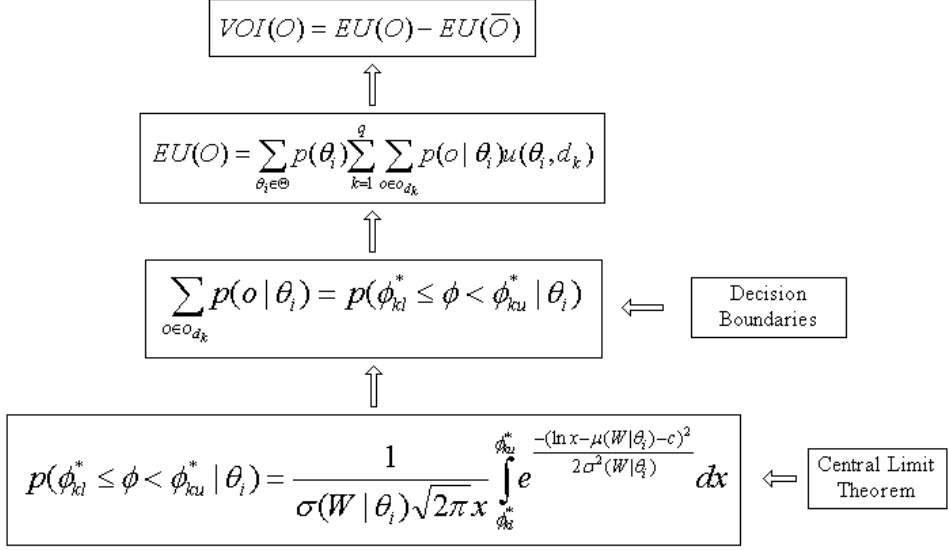
13

$$VOI(O) = EU(O) - EU(\overline{O})$$

$$EU(O) = \sum_{\theta_i \in \Theta} p(\theta_i) \sum_{k=1}^{q} \sum_{o \in o_{d_k}} p(o \mid \theta_i) \mu(\theta_i, d_k)$$

$$\sum_{o \in o_{d_k}} p(o \mid \theta_i) = p(\phi_{kl}^* \leq \phi < \phi_{ku}^* \mid \theta_i)$$

Decision Boundaries

$$p(\phi_{kl}^* \leq \phi < \phi_{ku}^* \mid \theta_i) = \frac{1}{\sigma(W \mid \theta_i)\sqrt{2\pi}x} \int_{\phi_{kl}^*}^{\phi_{ku}^*} e^{\frac{-(\ln x - \mu(W \mid \theta_i) - c)^2}{2\sigma^2(W \mid \theta_i)}} dx$$

Central Limit Theorem

Fig. 6. The key equations to approximate VOI when Θ has only two states, D has multiple rules, and the other nodes have multiple states.

3.4 Generalization

In the previous algorithm, the node Θ only allows two states, although the other random nodes and the decision node can be multiple states. However, in real-world applications, Θ may have more than two states. In this section, we extend the algorithm to the case that Θ can have several states too. Assume Θ has h states, $\theta_1, ..., \theta_h$, and still, d has q rules, $d_1, ..., d_q$, similarly to Equation 11, we have the following equations:

$$\frac{p(\theta_i|O)}{p(\theta_h|O)} = \frac{p(O^{s_1}|\theta_i)}{p(O^{s_1}|\theta_h)} ... \frac{p(O^{s_g}|\theta_i)}{p(O^{s_g}|\theta_h)} \frac{p(\theta_i)}{p(\theta_h)}, i \neq h \tag{18}$$

$$\Rightarrow ln\frac{p(\theta_i|O)}{p(\theta_h|O)} = \sum_{k=1}^{g} ln\frac{p(O^{s_k}|\theta_i)}{p(O^{s_k}|\theta_h)} + ln\frac{p(\theta_i)}{p(\theta_h)}$$

$$\Rightarrow ln\phi_i = \sum_{k=1}^{g} w_k^i + c_i,$$

$$\text{where } \phi_i = \frac{p(\theta_i|O)}{p(\theta_h|O)}, \ w_k^i = ln\frac{p(O^{s_k}|\theta_i)}{p(O^{s_k}|\theta_h)}, \ c_i = \ln\frac{p(\theta_i)}{p(\theta_h)}$$

Let $W_i = \sum_{k=1}^{g} w_k^i$, $i \neq h$, W_i still has a Gaussian distribution. Here, we assume $p(\theta_h|O)$, $p(O^{s_k}|\theta_h)$, and $p(\theta_h)$ are not equal to 0. The similar method in Section 3.3 can be used to compute the variance and the mean. Specifically, for the new defined w_k^i in the above equation, Table 1 can be modified as follows:

Thus, we get the following equations:

14

Table 2

The probability distribution of w_k^i.

w_k^i	$p(w_k^i|\theta_1)$	...	$p(w_k^i|\theta_h)$
$ln\frac{p(o_1^{s_k}|\theta_i)}{p(o_1^{s_k}|\theta_h)}$	$p(o_1^{s_k}|\theta_1)$	...	$p(o_1^{s_k}|\theta_h)$
...	...	...	...
$ln\frac{p(o_{r_k}^{s_k}|\theta_i)}{p(o_{r_k}^{s_k}|\theta_h)}$	$p(o_{r_k}^{s_k}|\theta_1)$	...	$p(o_{r_k}^{s_k}|\theta_h)$

$$\mu(w_k^i|\theta_j) = \sum_{l=1}^{r_k} p(o_l^{s_k}|\theta_j)ln\frac{p(o_l^{s_k}|\theta_i)}{p(o_l^{s_k}|\theta_h)}, 1 \le i < h, 1 \le j \le h, 1 \le k \le g \quad (19)$$

$$\sigma^2(w_k^i|\theta_j) = \sum_{l=1}^{r_k} p(o_l^{s_k}|\theta_j)ln^2\frac{p(o_l^{s_k}|\theta_i)}{p(o_l^{s_k}|\theta_h)} - \mu^2(w_k^i|\theta_j) \quad (20)$$

Similar to Equation 14, the expected value and the variance of W_i can be obtained as we see here:

$$\mu(W_i|\theta_j) = \sum_{k=1}^{g} \mu(w_k^i|\theta_j), 1 \le i < h, 1 \le j \le h \quad (21)$$

$$\sigma^2(W_i|\theta_j) = \sum_{k=1}^{g} \sigma^2(w_k^i|\theta_j) \quad (22)$$

Accordingly, ϕ_i follows the log-normal distribution with $S_{ij} = \sigma(W_i|\theta_j)$ and $M_{ij} = \mu(W_i|\theta_j) + c_i$. We denote the probability density function of ϕ_i given θ_j as $f_{\theta_j}(\phi_i)$. Equations 19 and 21 show that the overall number of the computational steps to assess the parameters for the log-normal distributions is $4h\sum_{k=1}^{g} r_k + 2h(h-1)g$ when $h > 2$.

Even though $f_{\theta_j}(\phi_i)$ can be easily obtained, it is still necessary to get the decision boundaries for each optimal decision in order to efficiently compute $\sum_{o \in o_{d_k}} p(o|\theta_j)$. Therefore, a set of linear inequality functions need to be solved when Θ has more than two states. For example, if d_k is the optimal action, $EU(d_k)$ must be larger than the expected utility of taking any other action. Based on this, a set of linear inequality functions can be obtained:

$$p(\theta_1)u_{1k} + p(\theta_2)u_{2k} + ... + p(\theta_h)u_{hk} \ge p(\theta_1)u_{1j} + ... + p(\theta_h)u_{hj}$$

$$\Rightarrow \frac{u_{1k} - u_{1j} + u_{hj} - u_{hk}}{u_{hj} - u_{hk}} \cdot p(\theta_1) + ... + \frac{u_{(h-1)k} - u_{(h-1)j} + u_{hj} - u_{hk}}{u_{hj} - u_{hk}} \cdot p(\theta_{h-1}) \ge 1$$

$$\Rightarrow \frac{u_{1k} - u_{1j}}{u_{hj} - u_{hk}} \cdot \frac{p(\theta_1)}{p(\theta_h)} + ... + \frac{u_{(h-1)k} - u_{(h-1)j}}{u_{hj} - u_{hk}} \cdot \frac{p(\theta_{h-1})}{p(\theta_h)} \ge 1 \quad (23)$$

15

We assume $u_{hj} - u_{hk} > 0$; otherwise, "$\geq$" is changed to "$\leq$" in the last inequality.

Let A_k be the solution region of the above linear inequalities, then

$$\sum_{o \in o_{d_k}} p(o|\theta_j) = \int_{A_k} f_{\theta_j}(\phi_1)...f_{\theta_j}(\phi_{h-1})dA_k, 1 \leq j \leq h, 1 \leq k \leq q \qquad (24)$$

The right side of Equation 24 is an integral over the solution region A_k decided by the linear inequalities. We first demonstrate how to solve the integral when Θ has three states, and then introduce the method for the case that Θ has more than three states.

When Θ has three states, Equation 23 can be simplified as follows:

$$p(\theta_1)u_{1k} + p(\theta_2)u_{2k} + p(\theta_3)u_{3k} \geq p(\theta_1)u_{1j} + p(\theta_2)u_{2j} + p(\theta_3)u_{3j} \qquad (25)$$

$$\Rightarrow \qquad \alpha_{1kj}.\frac{p(\theta_1)}{p(\theta_3)} + \alpha_{2kj}.\frac{p(\theta_2)}{p(\theta_3)} \geq 1$$

$$\text{where } \alpha_{1kj} = \frac{u_{1k} - u_{1j}}{u_{3j} - u_{3k}}, \text{ and } \alpha_{2kj} = \frac{u_{2k} - u_{2j}}{u_{3j} - u_{3k}}$$

In the above, it is assumed that $u_{3j} > u_{3k}$; if $u_{3j} < u_{3k}$, then "$\geq$" is changed to "$\leq$" in the last inequality.

And Equation 24 can be simplified as follows:

$$\sum_{o \in o_{d_k}} p(o|\theta_j) = \int_{A_k} f_{\theta_j}(\phi_1)f_{\theta_j}(\phi_2)dA_k, 1 \leq k \leq q, 1 \leq j \leq 3 \qquad (26)$$

A_k is decided by $(q-1)$ linear inequalities and each inequality has two variables ϕ_1 and ϕ_2 as defined in Equation 25. We use the following steps to solve this integral when A_k is a finite region.

1. Identify all the lines that define the inequalities and find all the intersection points between any two lines as well as the intersection points between any line and the x (or y) axis.

2. Choose the intersection points that satisfy all the linear inequalities, and use them as vertices to form a polygon.

3. Divide the polygon into several simple regions:

16

Specifically, for each vertex, we generate a line crossing this vertex and parallel to the y-axis. The lines then divide the polygon into several simple regions.

4. Evaluate the integral in each simple region and sum the values together.

An example of the solution region is shown in Figure 7. In this example, if $\alpha_{1kj} > \alpha_{1kj}(i \neq j)$, then $\alpha_{2kj} > \alpha_{2kj}$ too. Therefore, the solution region can be decided by the intersection points of the lines that are defined by the linear inequalities and the axes. For example, in Figure 7, A_k is decided by a, b, c, and d, which are selected from the intersection points $\{(1/\alpha_{1kj}, 0), (0, 1/\alpha_{2kj}), j = 1, ..., q, j \neq k\}$. Based on (Cohen and Megiddo, 1994), the time complexity of solving m linear inequalities with n variables (each inequality only has 2 variables) is $O(mnlogm + mn^2log^2n)$. In this case, n is 2 and m is $q - 1$.
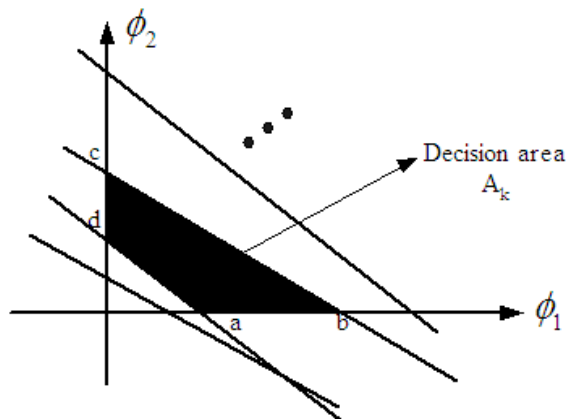

Fig. 7. A solution region of a group of linear inequalities.

When Θ has more than three states, the integral needs to be performed in a high-dimension space (dimension is larger than 2). Therefore, we solve it with Quasi-Monte Carlo integration (Hammersley, 1960; Kalos and Whitlock, 1986), which is a popular method to handle multiple integral. Quasi-Monte Carlo integration picks points based on sequences of quasirandom numbers over some simple domain A'_k which is a superset of A_k, checks whether each point is within A_k, and estimates the area (n-dimensional content) of A_k as the area of A'_k multiplied by the fraction of points falling within A_k. Such a method is implemented by Mathematica (Mathematica, 2006), which can automatically handle a multiple integral with a region implicitly defined by multiple inequality functions.

Figure 8 shows the key equations of the algorithm when Θ has multiple states. The main equations are similar to those in Figure 6. However, since Θ has multiple states, it becomes more complex to obtain the parameters of the log-normal distribution and perform the integration.

17

$$VOI(O) = EU(O) - EU(\overline{O})$$

$$EU(O) = \sum_{\theta_i \in \Theta} p(\theta_i) \sum_{k=1}^{q} \sum_{o \in o_{d_k}} p(o \mid \theta_i) u(\theta_i, d_k)$$

$$\sum_{o \in O_{d_k}} p(o \mid \theta_j) = \int_{A_k} f_{\theta_j}(\phi_1) \dots f_{\theta_j}(\phi_{h-1}) dA_k$$

Decision Region

$$\phi_i \sim \text{Log normal distribution}$$
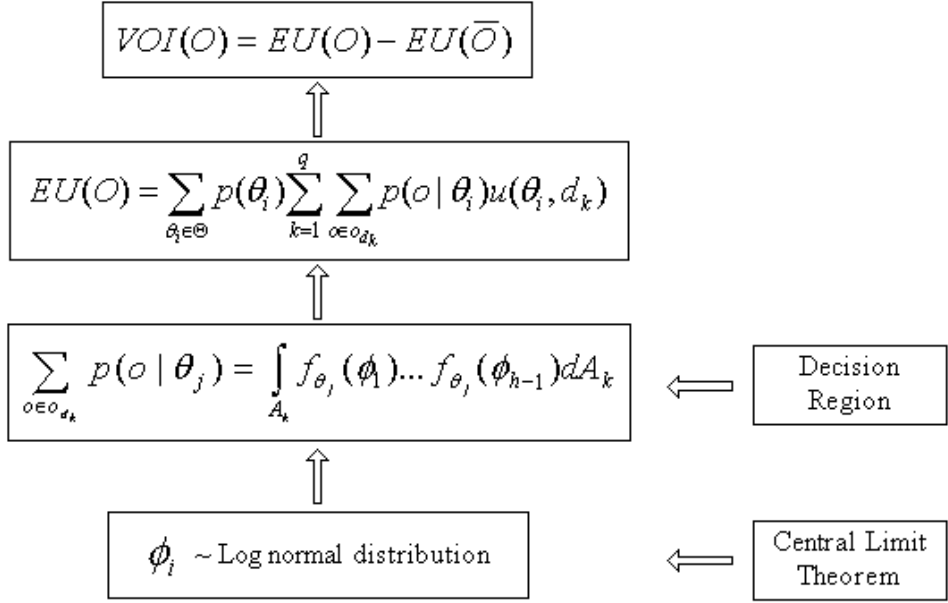
Central Limit Theorem

Fig. 8. The key equations to compute VOI when Θ has multiple states.

3.5 Algorithm Analysis

Now, we analyze the computational complexity of the proposed approximation algorithm compared to the exact computational method. For simplicity, assume that the number of the state of each O_i node is m, and there are n nodes in the set O. Assume we only count the time used for computing expected utilities. Then the computational complexity of the exact VOI computational method is approximately hm^n, where h is the number of the state of the Θ node. With the approximation algorithm, the computational complexity is reduced to hm^k, where h is the number of the state of the Θ node, and k is the number of O_i nodes in the maximum group among $\{O^{s_1}, ..., O^{s_g}\}$. In the best case, if all the O_i nodes are conditionally independent given Θ, the time complexity is about linear with respect to m. In the worst case, if all the O_i nodes are dependent, the time complexity is approximately m^n. However, usually, in most real-world applications, k is less than n, thus, the approximate algorithm is expected to be more efficient than the exact computational method, as will be shown in the experiments. For example, for the ID in Figure 5, $n = 9$, $m = 4$, $h = 3$, and $q = 3$. Then, for the exact computation, the number of computations is around $3 * 4^9 = 786432$, while using the approximate algorithm, the number of computations is only around $3 * 4^3 = 192$.

However, in addition to the cost of computing expected utilities, the approximation algorithm also includes some extra costs: sorting the utility functions (Section 3.2), partitioning the O set (Section 3.3.1), and deciding the decision boundaries (Section 3.2) when Θ has two states, or performing the integral when Θ has more than two states (Section 3.4). These costs are not included

18

in the above analysis. In general, the extra time in these steps is much less than the time used for computing expected utilities. For example, the time complexity of sorting is $O(qlog(q))$, the time complexity of the partition procedure is $O(|V| + |E|)$ (V is the set of vertex, and E is the set of edges in an ID), and the time complexity in deciding the decision boundaries when θ has two states is $O(q^2)$. When θ has more than two states, deciding the decision boundaries needs additional time. Empirically, it doesn't affect the overall speed, as will be shown in the experiments. In addition, most steps in computing expected utilities involve performing inferences in an ID, which is usually NP-hard and thus consumes much more time than a step in the procedures of sorting, partitioning, and integrating.

4 Experiments

The experiments are designed to demonstrate the performance of the proposed algorithm compared to the exact VOI computation. We limit the ID test model with at most 5 layers [2] and up to 11 information sources due to the exponential computational time behind the exact computation. Ten different ID models are constructed, where in one of the IDs the O nodes are conditionally independent given the Θ node. Table 3 describes the structures of these IDs. The IDs are parameterized with 150 sets of different conditional probability tables and utility functions, a process which yields 1500 test cases. In each the one-third of them, Θ node has 2, 3, and 4 states respectively. Without loss of generality, all the other random nodes and the decision node have four states.

Table 3
ID structures. k is the size of the biggest group after partitioning.

k	5	4	3	2	1
Number of IDs	2	3	3	1	1

For each test case, the VOIs for different O subsets with the size from 3 to 11 are computed. The results from the approximation algorithm are compared to the exact computation implemented with the brute-forth method. Let VOIt be the ground-truth, and VOI be the value computed with the proposed algorithm. Assuming $VOIt \neq 0$, the error rate is defined as follows:

$$Err = \frac{|VOIt - VOI|}{VOIt}$$

The 1500 test cases described previously are divided into six groups, named as

[2] The length of the longest path starting from (or ending at) the hypothesis node is 5.

19

ID_indep: 2-state, ID_indep:3-state, ID_indep:4-state, ID_dep:2-state, ID_dep: 3-state, and *ID_dep:4-state*. Table 4 describes the six groups.

Table 4
Testing cases.

ID_indep: 2-state	50 test cases, where O_i nodes are conditionally independent given Θ whose state is binary.
ID_indep: 3-state	50 test cases, where O_i nodes are conditionally independent given Θ who has 3 states
ID_indep: 4-state	50 test cases, where O_i nodes are conditionally independent given Θ who has 4 states
ID_dep: 2-state	450 test cases, where O_i nodes are conditionally dependent given Θ whose state is binary.
ID_dep: 3-state	450 test cases, where O_i nodes are conditionally dependent given Θ who has 3 states
ID_dep: 4-state	450 test cases, where O_i nodes are conditionally dependent given Θ who has 4 states

Figure 9 illustrates the results from the six groups of 1500 test cases. Chart (a) shows the average errors for each group, while Chart (b) shows the VOIs for one specific case, which is randomly chosen from the test cases from *ID_dep: 3-state*. As the set size of the O_i nodes increases, the error rate decreases. When the state number of Θ is the same, the error rates of the dependent cases are larger than the error rates of the conditional independent cases. This can be explained by the reason that the IDs in the dependent cases have fewer independent O subsets than the ID in the independent groups. Since the central-limit theorem is the basis of our algorithm, it works better when the number of w_i increases, which corresponds to the number of independent O subsets. Even when the size of O set is as small as 6, the average error is less than or around 0.1 for all the cases. We could run several larger IDs with much more O_i nodes, and the error curve would be progressively decreasing. Here, we intend to show the trend and the capability of this algorithm.

Charts (c) and (d) show the average computational time with the exact computation and the approximation computation. When the set size of the O_i nodes is small, the computational time is similar. However, as the size becomes larger, the computational time of the exact computation increases exponentially, while the computational time of the approximation algorithm increases much slower. Thus, the larger the O set size is, the more time the approximation algorithm can save. Likewise, as the number of the state of each O_i node
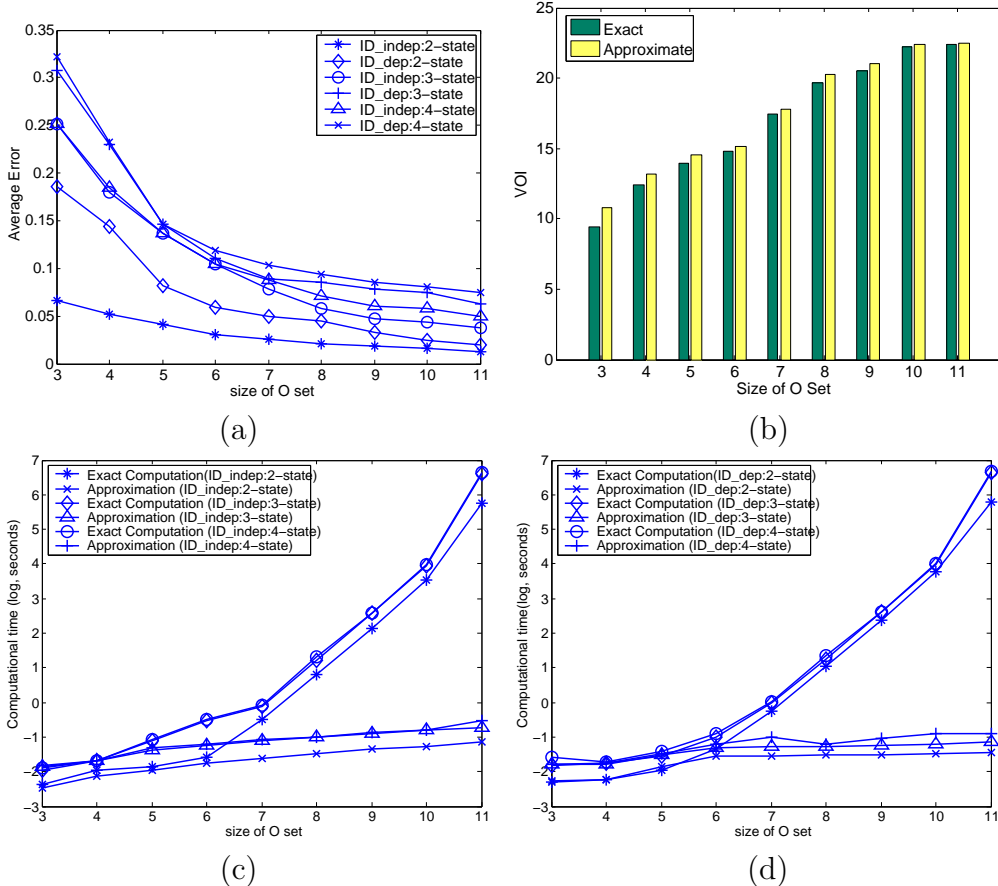
20

Fig. 9. Results from the four groups of 1500 test cases: (a) Average error rates with the approximation algorithm; (b) VOIt vs. VOI for one test case from *ID_dep: 3-state*; (c) Computational time (log(t), unit is second) for the groups of *ID_indep:n-state*, n = 2, 3, 4; and (d) Computational time (log(t), unit is second) for the groups of *ID_dep:n-state*, n=2,3,4

further increases, the computational saving would be more significant. As the number of states of Θ increase, the computational time also slightly increases.

5   An illustrative Application

We use a real-world application in human computer interaction to demonstrate the advantages of the proposed algorithm. Figure 10 shows an ID for user stress recognition and user assistance. The diagram consists of two portions. The upper portion, from the top to the "stress" node, depicts the elements that can alter human stress. These elements include the workload, the environmental context, specific character of the user such as his/her trait, and importance of the goal that he/she is pursuing. This portion is called predictive portion. On the other hand, the lower portion of the diagram, from the "stress" node to the leaf nodes, depicts the observable features that reveal stress. These

features include the quantifiable measures on the user physical appearance, physiology, behaviors, and performance. This portion is called diagnostic portion. The hybrid structure enables the ID to combine the predictive factors and observable evidence in user stress inference. For more detail please refer to (Liao et al., 2005).
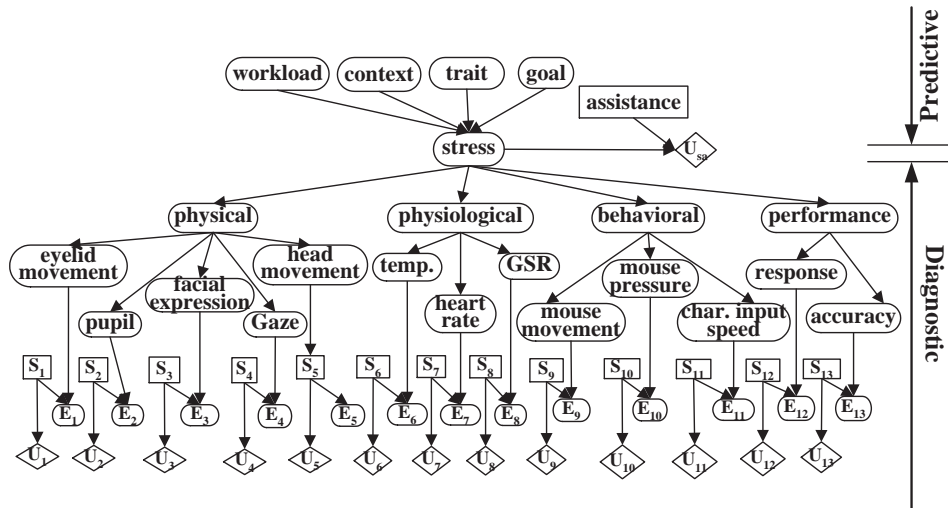

Fig. 10. An influence diagram for recognizing human stress and providing user assistance. Ellipses denote chance nodes, rectangles denote decision nodes, and diamonds denote utility nodes. All the chance nodes have three states.

To provide timely and appropriate assistance to relieve stress, two types of decision nodes are embedded in the model to achieve this goal. The first type is the assistance node associated with the stress node, which includes three types of assistance that have different degrees of impact and intrusiveness to a user. Another type of decision nodes is the sensing action node (S_i node in Figure 10). It decides whether to activate a sensor for collecting evidence or not. Through the ID, we decide the sensing actions and the assistance action sequentially. In order to first determine the sensing actions (which sensors should be turned on), VOI is computed for a set S consisting of S_i. Using the notations defined before, we have $VOI(S) = VOI(E) - \sum_{S_i \in S} u_i(S_i)$, where E is the set of observations corresponding to S and $VOI(E) = EU(E) - EU(\bar{E})$.

Figure 11 shows the experimental results for the stress model. We enumerate all the possible combinations of sensors and then compute the value-of-information for each combination. Chart (a) illustrates the average VOI errors for different sensor sets with the same size. And Chart (b) displays the Euclidean distance between the true and estimated probabilities $\sum_{o \in o_{d_k}} p(o|\theta_i)$

(Equation 26). Similarly to the simulation experiments, the error decreases as the size of O set increases, and the computational time increases almost linearly in the approximation algorithm.
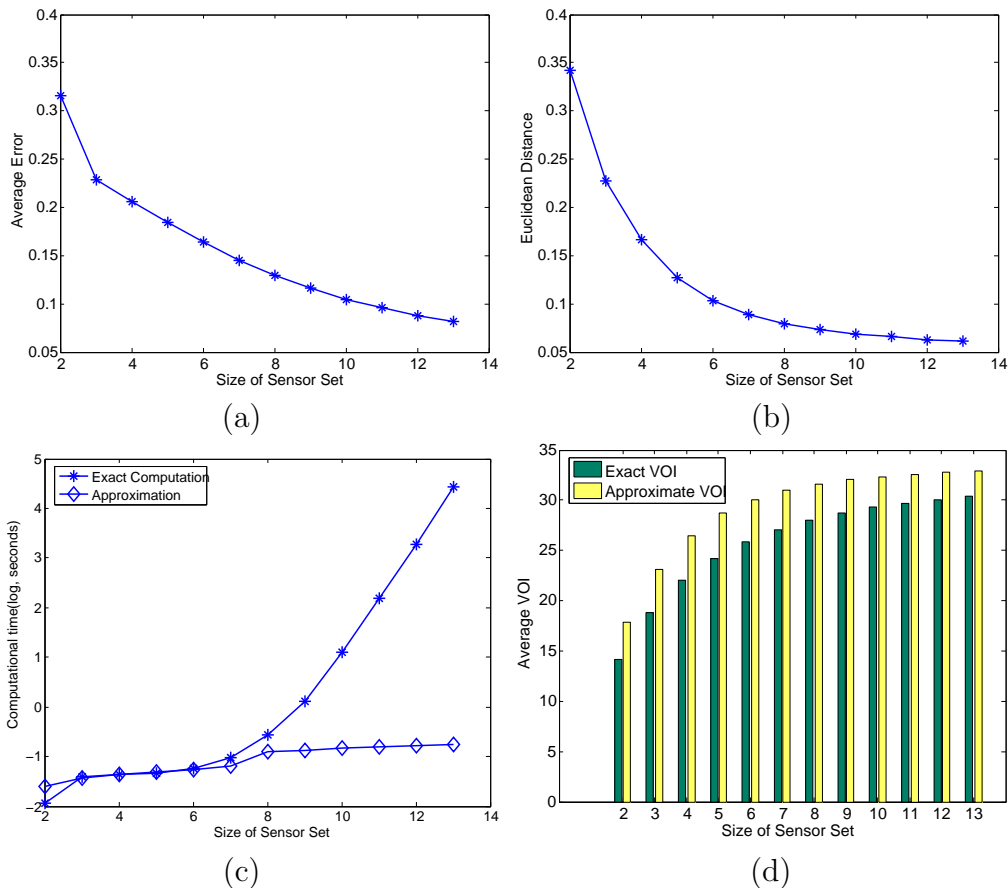
22

Fig. 11. Results for the stress modeling: (a) average errors with the approximation algorithm; (b) Euclidean distance between the true and approximated $\sum_{o \in o_{d_k}} p(o|\theta_i)$; (c) computational time (log(t), unit is second); (d) true VOI vs approximated VOI.

6 Conclusions and Future Work

As a concept commonly used in influence diagrams, VOI is widely used as a criterion to rate the usefulness of various information sources, and to decide whether pieces of evidence are worth acquiring before actually using the information sources. Due to the exponential time complexity of computing non-myopic VOI for multiple information sources, most researchers focus on the myopic VOI, which requires the assumptions ("No competition" and "One-step horizon") that may not meet the requirements of real-world applications.

We thus proposed an algorithm to approximately compute non-myopic VOI efficiently by utilizing the central-limit theorem. Although it is motivated by the method of (Heckerman et al., 1993), it overcomes the limitations of their method, and works for more general cases, specifically, no binary-state assumption for all the nodes and no conditional-independence assumption for the information sources. Table 5 compares our method with the method in

(Heckerman et al., 1993). Due to the benefits of our method, it can be applied to a much broader field. The experiments demonstrate that the proposed algorithm can approximate the true non-myopic VOI well, even with a small number of observations. The efficiency of the algorithm makes it a feasible solution in various applications when efficiently evaluating a lot of information sources is necessary.

Table 5
The proposed algorithm vs. the algorithm in (Heckerman et al., 1993).

Our algorithm	Heckerman's algorithm
Hypothesis node (Θ) can be multiple states	Θ has to be binary
Decision node (D) can have multiple rules	D has to be binary
Information sources nodes (Os) can be dependent from each other	Os have to be conditionally independent from each other

Nevertheless, the proposed algorithm focuses on the influence diagrams with one decision node under certain assumptions. For example, currently, we assume the the hypothesis node Θ and the decision node d are independent. If D and Θ are dependent, but conditionally independent given the observation set O, Equations (5) and (6) will not be affected, so our algorithm can still apply. However, if D and Θ are dependent given O, it may be difficult to directly apply our algorithm. Another scenario is that when there are more than one hypothesis node and/or utility nodes. One possible solution is to group all these hypotheses nodes into one. We would like to study these issues in the future.

Appendix

Proposition 1: *Let $r_{jk} = \frac{u_{2j}-u_{2k}}{u_{1k}-u_{1j}+u_{2j}-u_{2k}}$, $p_{kl}^* = \max\limits_{k<j\leq q} r_{jk}$, and $p_{ku}^* = \min\limits_{1\leq j<k} r_{jk}$, then d_k is the optimal action if and only if $p_{kl}^* \leq p(\theta_1) \leq p_{ku}^*$.*

Proof of Proposition 1:

$\Rightarrow$ In this direction, we prove that if d_k is the optimal action, $p(\theta_1) \geq \max\limits_{k<j\leq q} r_{jk}$ and $p(\theta_1) \leq \min\limits_{1\leq j<k} r_{jk}$.

If d_k is the optimal action, $EU(d_k)$ must be larger than or equal to the expected utility of any other action. Based on the definition, the expected utility of taking the action d_k is $EU(d_k) = p(\theta_1)*u_{1k}+p(\theta_2)*u_{2k}$, where $u_{1k} = u(\theta_1, d_k)$, and $u_{2k} = u(\theta_2, d_k)$. Therefore, we get the equations as follows:

$$EU(d_k) \geq EU(d_j), \text{ for } \forall j, j \neq k \tag{27}$$

$$\Rightarrow p(\theta_1) * u_{1k} + p(\theta_2) * u_{2k} \geq p(\theta_1) * u_{1j} + p(\theta_2) * u_{2j} \tag{28}$$

$$\Rightarrow \quad p(\theta_1) \geq \frac{u_{2j} - u_{2k}}{u_{1k} - u_{1j} + u_{2j} - u_{2k}} = r_{jk} \text{ if } j > k \tag{29}$$

$$p(\theta_1) \leq \frac{u_{2j} - u_{2k}}{u_{1k} - u_{1j} + u_{2j} - u_{2k}} = r_{jk} \text{ if } j < k \tag{30}$$

Thus, based on the above equations, $p(\theta_1) \geq \max_{k < j \leq q} r_{jk}$ and $p(\theta_1) \leq \min_{1 \leq j < k} r_{jk}$.

$\Leftarrow$ In this direction, we prove that if $p(\theta_1) \geq \max_{k < j \leq q} r_{jk}$ and $p(\theta_1) \leq \min_{1 \leq j < k} r_{jk}$, then d_k is the optimal action.

If $p(\theta_1) \geq \max_{k < j \leq q} r_{jk}$, $\forall j, k < j \leq q$, we get

$$p(\theta_1) \geq r_{jk} = \frac{u_{2j} - u_{2k}}{u_{1k} - u_{1j} + u_{2j} - u_{2k}} \tag{31}$$

$$\Rightarrow \quad p(\theta_1)(u_{1k} - u_{1j} + u_{2j} - u_{2k}) \geq u_{2j} - u_{2k} \tag{32}$$

$$\Rightarrow p(\theta_1) * u_{1k} + (1 - p(\theta_1)) * u_{2k} \geq p(\theta_1) * u_{1j} + (1 - p(\theta_1)) * u_{2j} \tag{33}$$

$$\Rightarrow \quad EU(d_k) \geq EU(d_j) \tag{34}$$

Similarly, for $\forall j, 1 \leq j < k$, we can get $EU(d_k) \geq EU(d_j)$. Therefore, d_k has the maximal expected utility and thus is the optimal decision.

Proposition 2: $\sum_{o \in o_{d_k}} p(o) = p(p_{kl}^* \leq p(\theta_1|o) \leq p_{ku}^*)$

Proof of Proposition 2:

Based on Proposition 1, d_k is the optimal decision if and only if the value of $p(\theta_1)$ is between p_{kl}^* and p_{ku}^*. Therefore, given an instantiation o, the probability that d_k is the optimal decision is equal to the probability that $p(\theta_1|o)$ is between p_{kl}^* and p_{ku}^*, i.e., $p(p_{kl}^* \leq p(\theta_1|o) \leq p_{ku}^*)$.

On the other hand, we know that o_{d_k} is a subset of instantiations, each of which corresponds to the optimal action d_k. Therefore, as long as o belongs to the set of o_{d_k}, d_k will be the optimal decision. In other words, the probability of d_k being the optimal decision is the sum of the probability of each $o \in o_{d_k}$, which is $\sum_{o \in o_{d_k}} p(o)$. Therefore, $\sum_{o \in o_{d_k}} p(o) = p(p_{kl}^* \leq p(\theta_1|o) \leq p_{ku}^*)$.

25

References

Balakrishnan, N., Chen, W. W. S., 1999. Handbook of Tables for Order Statistics from Lognormal Distributions with Applications. Amsterdam, Netherlands: Kluwer.

Casella, G., Berger, R., 1990. Statistical Inference. Brooks/Cole, Ch. 2, pp. 45–46.

Cohen, E., Megiddo, N., 1994. Improved algorithms for linear inequalities with two variables per inequality. SIAM Journal on Computing 23 (6), 1313–1347.

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C., 2002. Introduction to Algorithms. MIT Press and McGraw-Hill.

Crow, E., Shimizu, K., 1988. Lognormal Distributions: Theory and Applications. New York: Dekker.

Diehl, M., Haimes, Y., 2004. Influence diagrams with multiple objectives and tradeoff analysis. IEEE Transactions on Systems, Man and Cybernetics, Part A 34 (3), 293–304.

Dittmer, S., Jensen, F., 1997. Myopic value of information in influence diagrams. Proceedings of the Thirteenth Conference on Uncertainty in Artificial Intelligence, 142–149.

Ezawa, K. J., 1998. Evidence propagation and value of evidence on influence diagrams. Operations Research 46 (1), 73–83.

Feller, W., 1971. An Introduction to Probability Theory and Its Applications, 3rd Edition. Vol. 2. New York: Wiley.

Hammersley, J., 1960. Monte carlo methods for solving multivariable problems. Annals of the New York Academy Sciences 86, 844–874.

Heckerman, D., Horvitz, E., Middleton, B., 1993. An approximate nonmyopic computation for value of information. IEEE Transactions on Pattern Analysis and Machine Intelligence 15 (3), 292–298.

Howard, R., 1967. Value of information lotteries. IEEE Transactions of Systems Science and Cybernetics 3 (1), 54–60.

Howard, R., Matheson, J., 1981. Influence diagrams. Readings on the Principles and Applications of Decision Analysis 2, 721–762.

Jensen, F., 2001. Bayesian Networks and Decision Graphs. Springer Verlag, New York.

Jensen, F., Jensen, F. V., Dittmer, S., 1994. From influence diagrams to junction trees, 367–374.

Kalos, M., Whitlock, P., 1986. Monte Carlo Methods. New York: Wiley.

Korb, K. B., Nicholson, A. E., 2003. Bayesian Artificial Intelligence. Chapman and Hall/CRC.

Liao, W., Ji, Q., 2006. Efficient active fusion for decision-making via voi approximation. Twenty-first National Conference on Artifical Intelligence (AAAI).

Liao, W., Zhang, W., Zhu, Z., Ji, Q., 2005. A decision theoretic model for stress recognition and user assitance. Twentieth National Conference on Artifical Intelligence (AAAI), 529–534.

Mathematica, 2006. http://www.wolfram.com/products/mathematica/index.html.

Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems. Morgan Kaufmann Publishers.

Poh, K. L., Horvitz, E., 1996. A graph-theoretic analysis of information value. Proceedings of the 12th Annual Conference on Uncertainty in Artificial Intelligence (UAI-96), 427–435.

Raiffa, H., 1968. Decision Analysis. Addison-Wesley.

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

Shachter, R., 1999. Efficient value of information computation. Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-99), 594–601.

Shachter, R., Ndilikilikesha, P. M., 1993. Using potential influence diagrams for probabilistic inference and decision making. Proceedings of the Ninth Annual Conference on Uncertainty in Artificial Intelligence (UAI-93), 383–390.

Yokota, F., Thompson, K. M., 2004. Value of information analysis in environmental health risk management decisions: Past, present, and future. Risk Analysis 24, 635–650.

Zhang, N. L., Qi, R., Poole, D., 1993. Incremental computation of the value of perfect information in stepwise-decomposable influence diagrams. Proceedings of the Ninth Conference on Uncertainty in Artificial Intelligence, 400–410.

Dear Reviewers and Editor,

   We are very glad to know that our paper will become acceptable after a minor revision. Thanks for the comments about our paper (IJA-D-07-00004R1). We have addressed every issue raised by the reviewer. The details can be found in the following itemized answers to the reviewer' questions as well as in the revised manuscript, where the new changes are marked in bold.

1) *If section 2 should be a general introduction to influence diagram, it might be appropriate to also discuss no-forgetting and the requirement of a linear ordering of the decisions. If you only want to consider the type of IDs used for VOI analysis you could drop the representation of e.g. time precedence and functional dependence, since these type of relations doesn't appear in the diagrams you consider.*

Following the reviewer's suggestion, we added more information about different types of IDs. We also clearly indicate the ID type considered in this paper. Please refer to paragraph 2, page 5.

2)*On page 22: Doesn't the computational time increase exponentially in the size of the O set (the y-scale is logarithmic).*

For the exact computation, the computational time increases exponentially in the size of the O set, as shown in Figure 11(c). But in the approximate algorithm, it only depends on the size of the largest group among the conditionally independent O subsets. As shown in Figure 11(c), the approximate algorithm saves time significantly.

3) *Comment (on our response to question 1 in the previous response letter): I don't quite agree with this. If you have, say, two decisions D1 and D2 (with D1<D2) and you want to calculate the VOI for a set of observations O, then your algorithm would compare the EU of observing O before D1 with the EU of observing O (possibly never) after D2. You don't consider the possibility of observing O before D2 but after D1.*

Moreover, I would prefer that this was clearly stated already in the introduction. As it is now, this is first described on page 6, and here it appears only implicitly through the description of the type of influence diagram that you consider.

We agree with the reviewer that the proposed algorithm doesn't take decision sequence into account. What we tried to say in the previous response is that, our algorithm can still apply to some IDs with a sequence of actions. The application given in Section 5 is such a case.

Following the reviewer's suggestion, we added the descriptions of IDs that our algorithm fits in to the Introduction section. Please refer to the last paragraph in page 3.

5) *Comment: The limitations of the algorithm shouldn't be tugged away in the conclusions section, but should be stated explicitly when it becomes relevant. For*

1

example, it is only at the conclusion where you explicitly state that the D decision variable and the hypothesis variable should be independent. Similar, you assume that there is only one hypothesis node, but this is only stated in the conclusion.

Following the reviewer's suggestion, the limitations are presented not only in the Introduction and Conclusion, but also Section 3 when the algorithm is presented. Please refer to paragraph 2 of page 6.

6) Finally, the manuscript could still benefit from a thorough check of spelling and grammar. Below are a few examples…

We fixed the examples pointed by the reviewer and also proofread the paper.