An Improved Method for Solving Hybrid Influence Diagrams

1   Barbaros Yet*[1], Martin Neil[2], Norman Fenton[2], Anthony Constantinou[2],

2   Eugene Dementiev[2]

3   [1] Department of Industrial Engineering, Hacettepe University, Ankara, Turkey, 06800

4   [2] School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK, E1 4NS

5   *Corresponding Author. Tel: +90 312 780 5577 E-Mail: barbaros.yet@hacettepe.edu.tr

6

10

11   **Abstract**

12   While decision trees are a popular formal and quantitative method for determining an optimal

13   decision from a finite set of choices, for all but very simple problems they are computationally

14   intractable. For this reason, Influence Diagrams (IDs) have been used as a more compact and

15   efficient alternative. However, most algorithmic solutions assume that all chance variables are

16   discrete, whereas in practice many are continuous. For such 'Hybrid' IDs (HIDs) the current-state-

17   of-the-art algorithms suffer from various limitations on the kinds of inference that can be

18   performed. This paper presents a novel method that overcomes a number of these limitations. The

method solves a HID by transforming it to a Hybrid Bayesian Network (HBN) and carrying out inference on this HBN using Dynamic Discretization (DD). It generates a simplified decision tree from the propagated HBN to compute and present the optimal decisions under different decision scenarios. To provide satisfactory performance the method uses 'inconsistent evidence' to model functional and structural asymmetry. By using the entire marginal probability distribution of the continuous utility and chance nodes, rather than expected values alone, our method also enhances decision analysis by offering the possibility to consider additional statistics other than expected utility, such as measures of risk. We illustrate our method by using the oil wildcatter example and its variations with continuous nodes. We also use a financial score to combine risk and return measures, for illustration.

Keywords: Hybrid Influence Diagrams; Bayesian Networks; Dynamic Discretization

1   Introduction

An Influence Diagram (ID) is a graphical probabilistic model that offers a general and compact representation of decision making problems under uncertainty (Howard and Matheson, 2005, Jensen and Nielsen, 2009). Figure 1 shows an ID of the oil wildcatter problem (Raiffa, 1968). In this problem the wildcatter is searching for oil, and has to decide whether to *drill* (D) a particular site. He is uncertain about the *quantity of oil available* (O). The wildcatter can make a *seismic test* (T), which can reveal more information about presence of oil, but the r*esult of this test* (R) is not totally accurate. In this ID rectangles represent decision nodes, ellipses represent chance nodes, and diamonds represent utility nodes. Each decision node represents a decision making stage, each chance node represents a random variable, and each utility node has an associated table or a continuous probability distribution that defines the utility values based on the states of its parents.

1   Chance nodes can either be observed or not - for example, the chance node O generally cannot be

2   observed, whereas a chance node, such as the test result R, may be observed if the decision maker

3   decides to go ahead with a decision (in this case to undertake the seismic test T). Incoming arcs to

4   chance or utility nodes represent causal, deterministic or associational relations between the node

5   and its parents. Incoming arcs to decision nodes (shown by a dashed line) are 'informational' arcs,

6   representing the assumption that the state of any parent node must be known before the decision is

7   made. Informational arcs also specify a partial sequential order of decisions and observations.
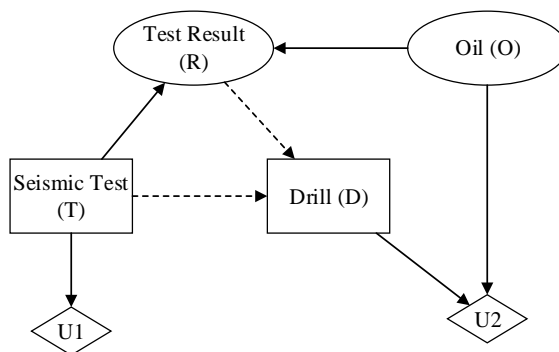
8

9   **Figure 1 Influence Diagram of Oil Wildcatter Example**

10   Generally, the chance, decision and utility nodes of an ID can be discrete or continuous variables.

11   Many real-world problems can be represented by using a mixture of both discrete and continuous

12   variables. An ID used to represent such a problem is called a Hybrid ID (HID), and, as we explain

13   in Section 2, the current state-of-the-art algorithms suffer severe limitations when attempting to

14   solve HIDs. This paper describes a novel method and algorithm, to solve HIDs, designed to

15   overcome these limitations. Our method is based on using the Dynamic Discretization (DD)

16   algorithm (Neil et al., 2007), which was developed to solve Bayesian Networks (BNs) containing

17   continuous and discrete variables, to solve HIDs and to provide optimal strategies in a simplified

18   Decision Tree (DT) that contains only decision and observable chance nodes. Unlike previous

19   algorithms, our method provides a fully automated solution for HIDs that contain continuous

chance nodes with virtually any probability distribution, including non-Gaussian types, or any conditionally deterministic function of these distributions.

The paper is structured as follows: In Section 2 we describe the state-of-the-art of DTs, IDs and their algorithms, highlighting their limitations with respect to HIDs. Sections 3 and 4 describe our novel method that adapts an existing BN DD algorithm to solve HIDs, and Section 5 presents our conclusions.

2   Decision Trees (DTs), Influence Diagrams (IDs) and Hybrid Influence Diagrams (HIDs)

In this section, we discuss the advantages and limitations of previous DT, ID and HID methods for solving decision making problems under uncertainty.

2.1   Decision Trees (DTs)

DTs have traditionally been used to choose an optimal decision from a finite set of choices, which are sometimes called policies. Typically, the value being optimized is some utility function expressed for each possible outcome of the decision. A DT represents the structure of a decision problem by modeling all possible combinations of decisions and observations, usually in the particular sequence which one would expect observations and decisions to be made. DTs are composed of three types of nodes: chance nodes, decision nodes and utility nodes. Each outgoing arc from a chance node represents an outcome and is labeled with the name and the probability of this outcome. Each outgoing arc from a decision node is labeled with a decision alternative. The DT in Figure 2 is a representation of the wildcatter ID problem shown in Figure 1.

Figure 2 Decision Tree of Oil Wildcatter Example

While DTs are a conceptually simple and popular method for decision analysis in practice there

are a number of known limitations, the main ones being:

- A DT specifies all possible sequences of observations and decisions as paths from the root node to the leaf nodes. This causes the number of state combinations to grow in size exponentially as the number of decisions and outcomes increase. This means that even simple decision problems can have infeasibly large DTs especially when there are multiple unobservable chance nodes. Domain experts may not be able to build or interpret such complex DTs effectively.

- Each path from the root to a leaf of the model represents a sequence of decisions and observations, called a decision scenario. A DT assumes 'no forgetting', i.e. at any point in the DT the decision maker knows the states of all previous nodes from the root node. The sequential order between the decision nodes and chance nodes is defined according to information available at each decision making stage. However, the order between consecutive chance nodes in a tree is usually arbitrary regardless of the conditioning and informational relationships that exist in the real world, such as those that represent causality (Howard and Matheson, 2005). This further increases the difficulty of understanding of complex DTs as experts often describe and interpret domain knowledge by using causal statements (Fenton and Neil, 2012).

- The optimization is generally based on the expected values of the utility function. This is restrictive, especially when the utility nodes are continuous. We might instead be interested in the complete probability distribution of the utility function (which might be non-linear), any moments of that function, such as the variance, or some arbitrary combined function such as the "risk". Knowing the complete distribution of outcomes enables us to analyze and compare the optimal decisions for each of those functions. This can be more useful than the probability distribution of the expected utility (i.e. the risk profile) alone.

The first two restrictions above are well known in the computer science and operations research literature and, in response, IDs were developed as an alternative.

2.2   Influence Diagrams (IDs)

An ID is an acyclic directed graph composed of chance, decision and utility nodes. A chance node represents a random variable in the same way as in a BN; indeed a BN can be considered as a special case of ID with only chance nodes. As in a BN a chance node has an associated Conditional Probability Distribution (CPD) defined over each state configuration of its parent nodes. CPDs of discrete chance nodes are defined in Conditional Probability Tables (CPTs). A decision node does not have a CPD since its incoming arcs represent the sequential order of decisions and observations. The states of a decision node represent the decision alternatives available at that stage in the overall decision problem. As for the states of chance nodes, a decision node's states must be mutually exclusive and exhaustive. Decisions are totally ordered in standard IDs, and hence a decision node and its parents must be ancestors of all subsequent decision nodes. However, this can be relaxed in some variations of IDs (Jensen and Vomlelová, 2002, Lauritzen and Nilsson, 2001). A utility node has an associated table that contains the utility values or distributions for all state combinations of its parents. We assume that a utility node cannot have non-utility nodes as children and must be specified as continuous point values conditioned on each relevant decision and chance node state combination.

Early methods for 'solving' IDs involved converting them to DTs and performing computations on the associated DTs. Olmsted (1983) and Shachter (1986) devised efficient approaches that involved eliminating nodes from an ID using value preserving transformations. Pearl (1988), and others (Cooper, 1988, Shachter and Peot, 1992) showed that IDs could be converted into BNs and

the computation could be carried out using any of the popular BN propagation algorithms. When representing IDs as BNs the general rules for IDs apply with the additional requirement that decision nodes are transformed into BN chance nodes with uniform probability distributions over all parent chance and decision nodes.

Afterwards, BN inference algorithms were adapted and optimized for IDs with the goal to speed up inference (Zhang, 1998). One of the most popular exact BN propagation algorithms is the junction tree (JT) algorithm. JT transforms the BN into an equivalent tree structure of clusters by merging nodes, and computes belief propagation on that structure (Lauritzen and Spiegelhalter, 1988, Jensen et al., 1990, Shenoy and Shafer, 1990). JT exploits the conditional independence assertions in the BN structure to ensure that global consistency can be obtained from local computations. Jensen and Dittmer (1994) developed an efficient algorithm to compute IDs, using BNs, which uses a strong JT with a special propagation scheme to calculate expected utilities. Cabanas et al. (2016) exploited context-specific independencies for inference by encoding the parameters of IDs in tree structures. Lazy propagation (Cabanas et al., 2013, Madsen and Jensen, 1999, Madsen and Nilsson, 2001) and sampling based approximate algorithms (Cano et al., 2006) have also been used to compute large IDs efficiently.

Since most popular BN algorithms, including JT, have been designed to solve discrete models, most ID algorithms require discrete IDs, i.e. those for which all nodes are discrete. Many real-world problems, however, are most naturally and accurately represented with both discrete and continuous variables, and thus their modeling requires HIDs that incorporate both discrete and continuous chance and utility nodes. Of course, it is possible to statically discretize the continuous variables - an assumption which is required of those commercial tools that implement IDs (Norsys, 2016, Hugin, 2016, BayesFusion, 2017) - but this presupposes that users can do so before observing

1 the posterior distributions and associated maxima involved. However, as this is the goal of

2 inference in the first place, this defeats the purpose of static discretization.

3 The earliest research on solving HIDs used simple manual discretization approaches for

4 approximating continuous nodes (Smith, 1993, Keefer and Bodily, 1983). Unlike the dynamic

5 discretization algorithm (which is discussed in Section 4.2), these simple approaches did not

6 optimize discretization according to the posterior density and thus they did not provide accurate

7 approximations of the posterior marginal probability distributions when conditioned on new

8 evidence. Early researchers also focused on Gaussian distributions due to their convenient

9 computational properties. The 'Gaussian' IDs developed by Shachter and Kenley (1989) contain

10 only Gaussian chance nodes. In their framework, chance nodes without parents are defined by

11 Gaussian distributions, and chance nodes with parents are defined by a linear function of its parents

12 and a constant variance. Discrete chance nodes are not allowed in this Gaussian framework but

13 Poland III (1994) extended Gaussian IDs by incorporating both Gaussian and discrete chance

14 nodes. In their model the probability distribution of conditioned nodes can be defined by a mixture

15 of Gaussian distributions but discrete nodes cannot have continuous parents. Madsen and Jensen

16 (2005) proposed a method that computes an exact solution for Gaussian models with an additively

17 decomposing quadratic utility function.

18 Cobb and Shenoy (2006), (2008) proposed a method that can adopt a wider variety of statistical

19 distributions, including beta, gamma and lognormal distributions, in IDs. Their method

20 approximates continuous chance nodes and utility functions to Mixtures of Truncated Exponential

21 functions (MTEs) whose exponent is a linear combination of the state of the child variable and its

22 parents. After this approximation, they marginalize continuous variables by integration, and then

23 solve the remaining discrete ID by using a discrete ID-solving technique (Shenoy, 1992). Cobb

(2007) extended the use of MTEs to solve IDs with continuous decision nodes. Their technique computes a piecewise linear decision rule for continuous decision nodes. The parameters of MTE are fitted to different distributions by using optimization methods, but this can be challenging for conditional continuous variables with continuous parents (Bielza et al., 2011). MTEs have closed form under multiplication and integration but they do not allow additive factorization of the joint utility function. Moreover, MTEs cannot deal with multidimensional linear deterministic conditional functions.

Li and Shenoy (2012) proposed using Mixtures Of Polynomials (MOPs) for approximating continuous nodes. These MOP approximations are computed more easily than MTEs, and they are closed for multidimensional linear deterministic conditional functions. Both MOP and MTE functions are also not closed for non-linear deterministic functions; they need to be approximated to piecewise linear functions (Cobb and Shenoy, 2005). Moreover, both MOP and MTE approximations of conditional distributions cannot currently be fully automated.

Markov Chain Monte Carlo (MCMC) methods have also been used to compute approximate solution to HIDs. Bielza et al. (1999) used MCMC to solve HIDs with continuous chance nodes and a single continuous decision node. Charnes and Shenoy (2004) used MCMC to solve HIDs with continuous chance and discrete decision nodes.

The outstanding challenges to solving HIDs, which we overcome with the new method described in the following section, are to:

- Offer a fully automated way for computing marginal distributions of continuous nodes in a HID.

- Allow statistical distributions (rather than just point values) for chance and utility nodes so that we are able not just to maximize expected utility but also reduce risk by minimizing variation (or by using any other moment of the utility distribution or measure derived from it)

- Present computed decision strategies in a simplified DT that only contains the decision and observable chance nodes. The tree structure offers a useful way of presenting strategies under different conditions, and the DT is kept concise by excluding the information about unobservable nodes.

- Avoid evaluating functional and structural asymmetries by using inconsistencies computed during BN propagation to detect asymmetry and thus avoid exploring impossible branches in the DT.

- Allow any factorization of the joint utility function, so it is possible to model linear and non-linear deterministic functions as well as uncertainty of all parameters of statistical distributions.

The method we describe in the following section first transforms a HID to a BN with discrete and continuous random variables, called a Hybrid BN (HBN), and then uses off-the-shelf BN algorithms (Jensen et al., 1990, Lauritzen and Spiegelhalter, 1988, Shenoy and Shafer, 1990) combined with the DD algorithm to propagate the HBN and generate a simplified DT containing only decision and observable chance nodes. It then solves the decision problem by quasi-exponential exploration of the DT. Whilst not as efficient as using strong JT propagation (Jensen and Dittmer, 1994), it has the advantage of being easier to implement and is more general. Recently, Luque et al. (2017) used a similar tree structure, called strategy trees, to present the optimal strategy

1 computed from a discrete ID model. The main difference between the structure of a strategy tree

2 and a simplified DT is that a strategy tree only contains the decision alternatives that are associated

3 with the optimal strategy, whereas a simplified DT contains all decision alternatives.

4 Because it uses DD our method offers a convenient way of modeling and computing HIDs where

5 there are continuous chance and utility nodes with a wide variety of statistical distributions and

6 conditionally deterministic functions. In the following section, we describe how a simplified DT is

7 built and solved by using a discrete ID example. In Section 4, we describe how this approach is

8 applied to HIDs by using the DD algorithm.

9 **3 Evaluation of IDs**

10 In what follows we assume:

11 • An ID contains $\bar{D} = D_1, \dots, D_k$ decision nodes, each with finite mutually exclusive states.

12   At each decision node we can express a decision alternative as a single state, d_{ij}, for

13   decision node D_i.

14 • There are two classes of chance nodes: those that precede a decision and are observable at

15   the time of, or before, the decision is made. These are called ***observable chance nodes***,

16   $\bar{O} = O_1, \dots, O_p$. The other chance nodes, $\bar{N} = N_1 \dots N_q$, are ***unobservable*** for all decisions

17   in the ID.

18 • There are utility nodes $\bar{U} = U_1, \dots, U_l$ that are combined by an additive, a multiplicative or

19   some other form of a conditionally deterministic utility function.

20 In Section 3.1, we describe how the method transforms an ID into a BN, and discuss the advantages

21 of BNs for dealing with asymmetric decision problems. In Section 3.2, we describe an algorithm

1   for generating a simplified DT from this BN to evaluate the decision problem. We use the wildcatter

2   ID example to illustrate these approaches.

3   **3.1   Converting Influence Diagrams to Bayesian Networks and its implications to**

4   **Asymmetric Influence Diagrams**

5   To convert the ID to a BN we first use the following algorithm:

6   1. Record the sequential order of the decisions $\overline{D} = D_1, \dots, D_k$ and observable chance nodes

7           $\overline{O} = O_1, \dots, O_p$ according to the informational arcs in the HID.

8   2. Convert the decision nodes $\overline{D} = D_1, \dots, D_k$ to corresponding BN nodes $\overline{\Delta} = \Delta_1, \dots, \Delta_k$

9   representing decisions. In this conversion, a decision state d_{ij} of a decision node D$_i$, is

10   converted to a state δ_{ij} of the corresponding BN node Δ_i. If there is no asymmetry, all states

11          $\delta_{i1}, \dots, \delta_{im}$ of Δ_i have equal probabilities in its CPT.

12   3. Convert the utility nodes $\overline{U} = U_1, \dots, U_l$ to corresponding BN nodes $\overline{Y} = Y_1, \dots, Y_l$. In our

13   framework $Y_1, \dots, Y_l$ can have child nodes that aggregate the utilities by a deterministic

14   equation (Tatman and Shachter, 1990). Other types of child nodes, however, are forbidden

15   for utility nodes.

16   A decision problem is asymmetric if the number of possible decision scenarios is less than the

17   cardinality of the Cartesian product of the state spaces of all decision and chance variables (Bielza

18   and Shenoy, 1999). Often, such asymmetry arises because the states in one decision or observable

19   chance node are constrained by the states in others i.e. they are not mutually exclusive. We can

20   classify asymmetry into three categories (Jensen and Nielsen, 2009, Bielza et al., 2011):

1. **Functional asymmetry** is present if the availability of some decision alternatives d_{ij} depend on preceding observations or decisions.

2. **Structural asymmetry** is present if the possibility of an observation O_i or a decision D_i depends on the preceding observations or decisions. In other words, a chance or decision node is defined based on the preceding nodes in this case. Therefore, an entire decision or chance node becomes impossible when structural asymmetry is present, whereas only a decision state becomes unavailable when functional asymmetry is present. For example, there is a structural asymmetry in the oil wildcatter example (Figure 1) because, if the seismic test (T) is not done, it is impossible to observe test results (R).

3. **Order asymmetry** is present if the sequential order of the decision and observable chance nodes is not defined.

Our framework encodes asymmetry constraints in the CPTs of corresponding BN nodes, and uses inconsistencies discovered during BN propagation to avoid evaluating those state combinations that are functionally or structurally asymmetric. The order asymmetry is considered beyond the scope of this paper.

In the presence of functional asymmetry, we assign zero probability values to the state combinations of decisions that are impossible due to the instantiated states of previous chance or decision nodes. When there is structural asymmetry we have to add a synthetic '*NA*' state to the variable which shows that the decision is not possible due to a previous decision or observation (Bielza et al., 2011, Fenton and Neil, 2012). The probability of the '*NA*' state will be one if the instantiations of its parents makes the variable impossible and zero otherwise. Assigning zero probabilities to asymmetric state combinations will cause BN propagation algorithms to throw an

1   inconsistency when that state is instantiated, thus pruning any decision tree branches emanating

2   from that point in the scenario.

3   In an ID informational arcs do not represent conditional dependence. Converting decision nodes

4   with informational arcs to BN nodes with parents adds d-connections that were not present in the

5   original ID. However, this does not change the conditional independence assertions as we always

6   instantiate the parents of a decision node when we propagate the BN when evaluating the ID. This

7   makes the decision node conditionally independent of its non-descendants (Pearl, 1988) and

8   therefore any backward propagation from the decision nodes is not possible, in the same way as in

9   IDs.

10   In the remainder of this section, we illustrate the above approach of transforming IDs to BNs, and

11   show how an optimal policy is determined by using the oil wildcatter example described in Section

12   1. This example is a discrete ID, and it is selected due to its simplicity, but transformations of HIDs

13   are also done in the same way. Extensions of this example with continuous chance and utility nodes

14   are discussed and solved in Section 4. Figure 1 shows the ID model of the oil wildcatter example.

15   This ID is transformed to a BN as follows:

16   1. Record the informational arcs to define the sequential order of the decisions and

17   observations. The wildcatter model has two decisions and one observable chance node and

18   the sequential order is as follows:

19 $T \prec R \prec D$

20   2. Transform the decision nodes into BN nodes.

1   3. Assign uniform distributions to the CPT of the '*Drill*' node as there is no functional

2   asymmetry.

3   4. Add an '*NA*' state to the '*Test Results*' node to represent the structural asymmetry ('*Test

4   Results*' cannot be observed when the '*Seismic Test*' is not done, so we model this

5   asymmetry by making the probability of the '*NA*' state one for the corresponding state

6   combinations) .

7   5. Transform the utility nodes into BN nodes.

8   6. Assign point integer values to the utility nodes *Y1* and *Y2*. and add an 'aggregate' utility

9   node *Y3* as a child of *Y1* and *Y2* to sum the utilities from these nodes (Tatman and Shachter,

10   1990).

T	Yes	Yes	Yes	No	No	No
O	Dry	Wet	Soaking	Dry	Wet	Soaking
No	0.6	0.3	0.1	0	0	0
Open	0.3	0.4	0.4	0	0	0
Closed	0.1	0.3	0.5	0	0	0
NA	0	0	0	1	1	1

O		
	Dry	0.5
	Wet	0.3
	Soaking	0.2

T		
	Yes	0.5
	No	0.5

T	Yes	Yes	Yes	Yes	No	No	No	No
R	No	Open	Closed	NA	No	Open	Closed	NA
Yes	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
No	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

T	Yes	No
Y1	-10	0

D	Yes	Yes	Yes	No	No	No
O	Dry	Wet	Soaking	Dry	Wet	Soaking
Y2	-70	50	200	0	0	0

Y3	Y1 + Y2

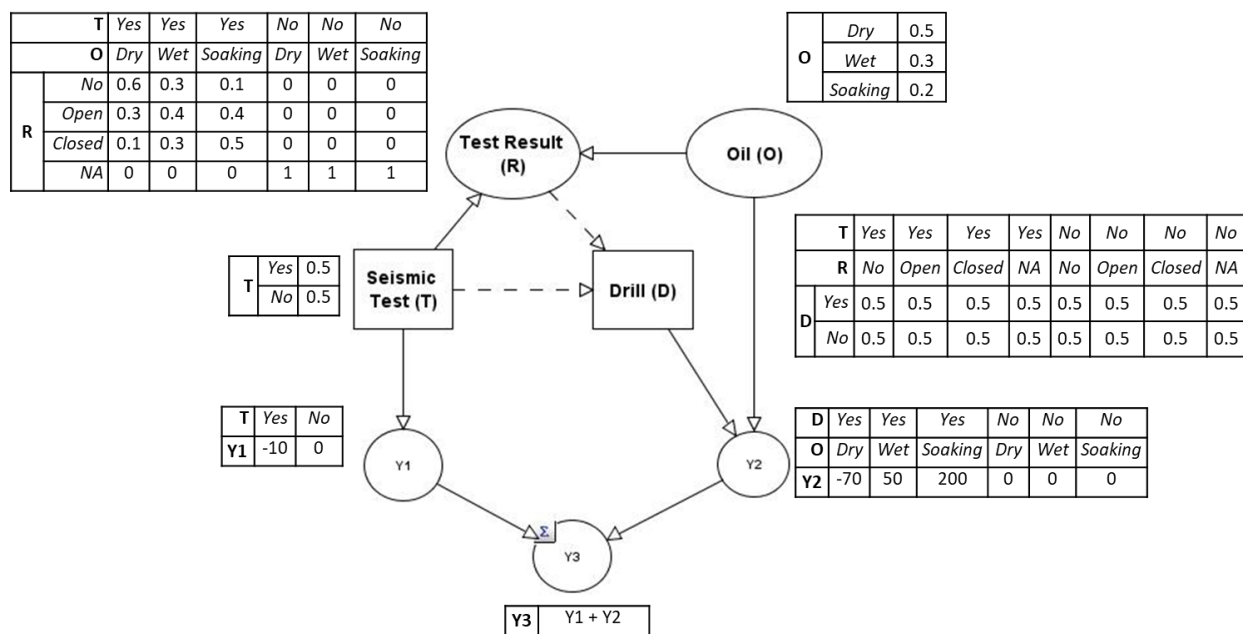11

12   **Figure 3 BN representation of Oil Wildcatter ID**

13   The resulting BN representation, and its CPDs, are shown in Figure 3. All of the variables in Figure

14   3 are BN nodes, which are equivalent to chance nodes in IDs. However, 'Seismic Test' and 'Drill'

15   are still shown by rectangles, and the incoming arcs to 'Drill' are still shown by dashed lines to

16

1 highlight that these nodes and arcs were respectively decision nodes and informational arcs in the

2 corresponding ID.

3 **3.2 Evaluating the transformed BN using a DT**

4 After converting an ID to a BN, and propagating using JT, the algorithm described in this section

5 generates a simplified DT from the propagated BN. The simplified DT only contains decision,

6 utility and observable chance nodes, and its size is not affected by the number of unobservable

7 chance nodes in a decision problem. Our algorithm uses the following recursive function to build

8 the simplified DT.

9   *1. Record the sequential order of decision and observable chance nodes in the ID.*

10   *2. Build a decision tree by calling Build DT(1st node in the sequential order)*

11   *3. Evaluate the decision tree using the standard 'averaging out and folding back'*

12   *algorithm, a.k.a. the rollback algorithm (e.g. see Chapter 2 of Raiffa (1968))*

13 where the function *Build DT* is:

14 *Build DT (i^{th} node in the sequential order)*

15   **for** *each state of the node:*

16   *Remove all evidence from the node and subsequent nodes in the sequential order.*

17   **if** *a corresponding node does not exist in the DT*

18   **if** $i = 1$

19   *Add a decision or chance node to the DT corresponding to the type*

20   *of the node in the ID.*

21   **else**

22   *Add a decision or chance node next to the last arc added in the DT*

23   *corresponding to the type of the node in the ID.*

24   **end if**

1	**end if**
2	*Add an arc next to the corresponding node in the DT.*
3	*Label the name of the state on that arc.*
4	**if** *the current state is from an observable chance node*
5	*Label its posterior probability from the BN on the arc added in the DT.*
6	**end if**
7	*Instantiate the state and propagate the BN.*
8	**if** *the instantiated state results in 'inconsistent evidence' after propagation*
9	*Remove the last arc added to the DT*
10	**else**
11	**if** *the state entered is from the last node in the sequential order*
12	*Add a utility node next to the last arc added in the DT, and label*
13	*the value of this node with the posterior value of the aggregate*
14	*utility node from the BN.*
15	**else**
16	*Recursively call Build DT by using the i+1$^{\text{th}}$ node in the sequential*
17	*order.*
18	**end if**
19	**end if**
20	**end for**

21 We illustrate this algorithm on the BN conversion of the wildcatter ID shown in Figure 3 as follows:

22 1. The sequential order of the decision and observable chance nodes of this model is $T \prec R \prec$

23 D (see Section 3.1).

24 2. As the DT is initially empty add a decision node labeled 'T' and arc next to this node labeled

25 '*Yes*'.

26 3. Start generating the DT with the first node in the sequential order 'T' and its state is '$T=Yes$'.

27 4. Instantiate '$T=Yes$' in the BN and propagate it to compute the posterior distributions of

28 other chance and utility nodes.

5. The second node in the sequence is an observable chance node '*R*'. The posterior probability distribution of *P(R/T=Yes)* is shown in Table 1. Note that, we do not need to evaluate state combinations including (*T=Yes, R=NA*) as the posterior probability in the BN reveals a structural asymmetry. We only evaluate the states that have non-zero posterior probabilities, and we first instantiate the state 'R=No'.

Table 1 Posterior Probability of P(R|T=Yes)

P(R| T = Yes)	
No	0.410
Open	0.350
Closed	0.240
NA	0

6. In the DT, add a chance node labeled *R*, and an arc labeled with both the name of this state and the posterior probability extracted from the BN. Again instantiate this state, and propagate the BN.

7. The last node in the sequential order is *'D'*, and both of its states *'D=Yes'* and *'D=No'* have non-zero posterior probabilities. Add a decision node and arc to the DT, labeled *'D'* and *'Yes'* respectively.

8. Instantiate *'D=Yes'* in the BN and propagate.

9. Since *'D=Yes'* is the last node in the sequential order of this state combination, add a utility node next the corresponding arc in the DT, and label the value of this node with the expected value of the posterior distribution of the aggregate utility node from the BN.

10. Clear the evidence entered on '*D*' and continue with its second state '*D=No*'. We need to add an arc labelled 'No' next to the node '*D*'.

1   *11.* Instantiate '*D=No*' and, propagate the BN. Add another utility node in the DT. Note that

2   the evidence of (T=Yes, R=No) is already instantiated in the BN, as we only changed the

3   evidence on 'D' at this step.

4   *12.* Since we have evaluated all states of '*D*', the algorithm proceeds to evaluating the second

5   state of *'R'.*
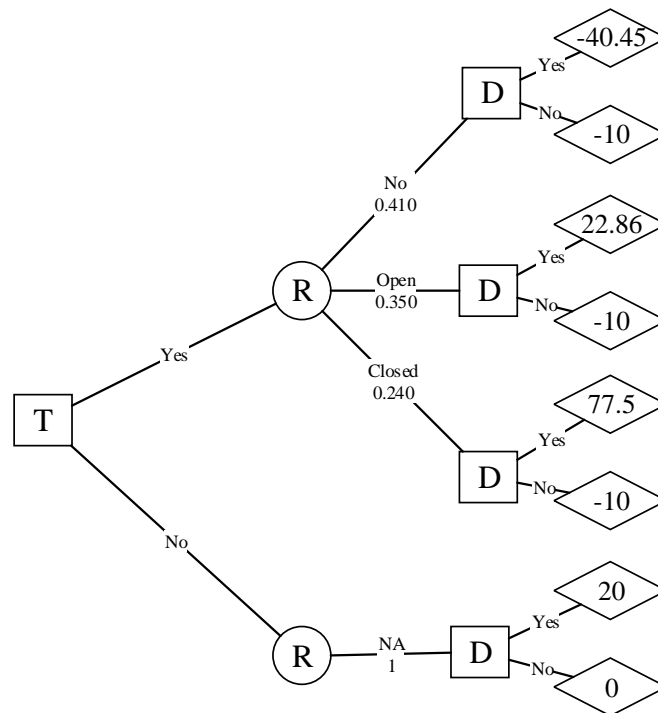
6

7   **Figure 4 DT generated from the BN**

8   The remainder of the possible state combinations in the sequential order are evaluated in the same

9   way by following the algorithm described. The resulting DT is shown in Figure 4, after evaluating

10   all possible state combinations. The algorithm propagated the BN for each arc in the resulting DT.

11   Note that the nodes 'T', 'R' and 'D' have sixteen state combinations but only eight of them have

12   been evaluated. The algorithm avoided evaluating the remaining eight combinations by discovering

13   zero posterior probabilities due to asymmetry. Moreover, computations of expected utility from

20

1   the posteriors of utility nodes and marginalization of unobservable chance nodes is handled by the

2   algorithm.

3   In Figure 4, the arc between *R* and *D*, in the state combinations with *(T=No, R=NA)*, is redundant

4   as its probability is one and it represents the structural asymmetry associated with *R*. Therefore,

5   this arc and *R* can be removed from the DT, and the arc representing *T=No* can be directly

6   connected to *D* (see Figure 5). The simplified DT grows exponentially as the number of decision

7   and observable chance variables increases but the size of the simplified DT is much smaller than a

8   normal DT as it excludes the state combinations associated with unobservable chance nodes.

9   Moreover, we use the simplified DT only to compute and show the optimal decision strategy. The

10   decision maker uses the underlying ID to build and interpret the decision problem. Therefore, our

11   method is an improvement considering the limitations discussed in Section 2.1 as it presents both

12   the decision problem and decision strategies in a clearer and more concise way than DTs.

13   After the simplified DT is prepared, the optimal decision strategy is computed from the DT by

14   using the standard 'averaging-out and falling-back' algorithm (e.g. see Chapter 2 of Raiffa (1968)).

15   This algorithm starts from the utility nodes, and rolls back towards the root node by computing the

16   weighted average of chance nodes and maximum of decision nodes. Figure 5 shows the optimal

17   policy on the simplified DT.
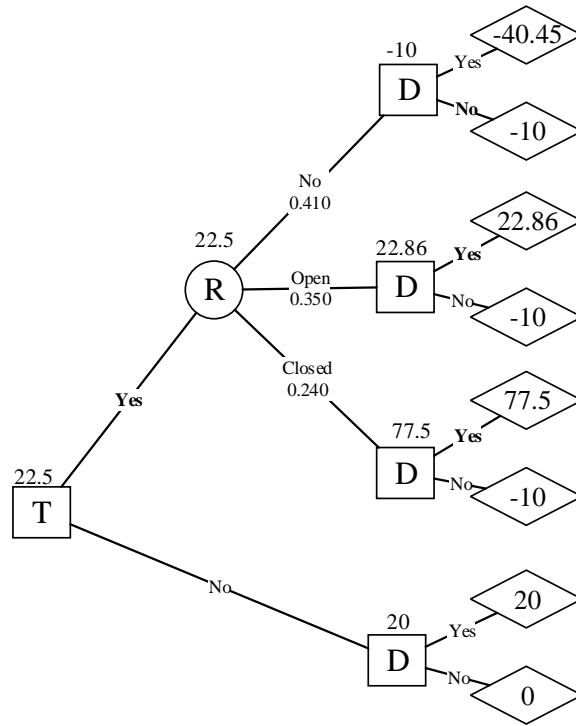
Figure 5 Simplified DT generated from BN

When an ID is converted to a BN, the informational arcs are converted to conditional BN arcs. These arcs only encode information about asymmetry; they do not represent CPDs as decision nodes do not have a probability distribution in an ID. Presence of these arcs in the BN is useful: it causes impossible state combinations to have zero posterior probabilities, and thus enables us to avoid their evaluation. These arcs, however, can also slow down computation as they can lead to large CPTs and thus large cliques in the JT. We can speed up the algorithm by removing the incoming arcs to decision nodes from the BN model. In order to do this, we still need to convert an ID to a BN as described in Section 3.1, examine the CPTs of the decision nodes, and note the impossible state combinations with zero probabilities. Afterwards, we remove the incoming arcs to decision nodes, and we avoid evaluating these impossible state combinations based on our list rather than zero posterior probabilities in the BN model. Removing these arcs does not change the results as informational arcs do not represent d-connections in IDs. Likewise, removing incoming

22

1 arcs does not change the structure of the algorithm as it still needs to use BN propagation to

2 discover and avoid evaluating impossible states due to asymmetry about observable chance nodes.

3 The structural asymmetry associated with the wildcatter example is encoded in the CPT of *R* by

4 using zero probabilities (see Figure 3). This can be detected if we examine the CPTs of all decision

5 and observable chance nodes for zero-probabilities. The impossible state combinations, due to this

6 asymmetry, are *(T=Yes, R=NA)*, *(T=No, R=No)*, *(T=Nos, R=Open)*, and *(T=Yes, R=Closed)*. We

7 can save these state combinations by examining the CPTs and skip evaluating them when we run

8 our algorithm. This improves the computational speed of the algorithm and preserves the results.

9 ## 4   Evaluation of HIDs

10 This section describes how our framework and algorithm is applied to HIDs by using the DD

11 algorithm. We first present a HID version of the wildcatter example with continuous unobservable

12 chance nodes (Section 4.1), and describe the DD algorithm by using the fragments of this HID

13 (Section 4.2). We illustrate the use of our framework and algorithm by also using this HID (Section

14 4.3), and another variant of it with continuous observable chance nodes (Section 4.5). An approach

15 to combine utility and risk measures when solving IDs (Section 4.4) is also described in this section.

16 AgenaRisk (AgenaRisk, 2017) was used to compute the underlying HBNs when solving these IDs.

17 ### 4.1   Converting HIDs to HBNs

18 A HID is an extension of an ID in which utility nodes $\bar{U}$, and observable and unobservable chance

19 nodes, $\bar{O}$ and $\bar{N}$, and decision nodes $\bar{D}$ can either be discrete or continuous. Figure 6 shows a revised

20 version of the oil wildcatter example that models the volume and price of oil, and the cost of drilling

21 with continuous variables (Poland III, 1994, Madsen and Jensen, 2005, Cobb and Shenoy, 2008).

1 In this model, the '*Oil Volume*' (V) has a mixture distribution conditioned on the type of the site.

2 A mixture distribution is a weighted combination of multiple probability distributions. In a BN, a

3 mixture distribution can be modelled by conditioning the probability distribution of a continuous

4 variable on a discrete random variable. If the site is '*dry*' the volume has a point value zero, but if

5 the site is '*wet*' or '*soaking*' the volume follows a normal distribution with mean 6 and 13.5, and

6 variance of 1 and 4 respectively. The price of drilling *U3* follows a normal distribution with mean

7 70 and variance 100. The oil price follows a lognormal distribution where the mean and variance

8 of the underlying normal distribution is 2.75 and 0.5. The probability distributions of '*Seismic*

9 *Test*', '*Oil*', '*Drill*', '*U1*', and '*Test Results*' are exactly the same as the discrete model shown in

10 Figure 3.
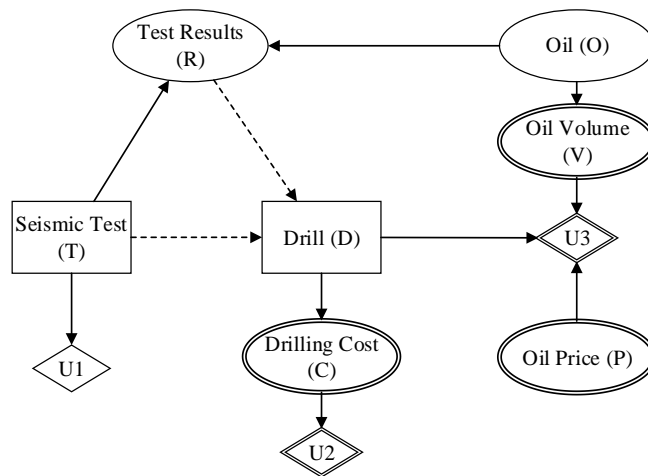
11

12 **Figure 6 Wildcatter HID with Unobservable Continuous Chance Nodes**

13 A HID is converted to a HBN in the same way as described in Section 3.1. An aggregate utility

14 node *Y4* is added to sum the values from other utility nodes. Figure 7 shows the graphical structure

15 and parameters of the BN representation of this revised oil wildcatter model.

Figure with tables:

T	Yes	Yes	Yes	No	No	No
O	Dry	Wet	Soaking	Dry	Wet	Soaking
R No	0.6	0.3	0.1	0	0	0
Open	0.3	0.4	0.4	0	0	0
Closed	0.1	0.3	0.5	0	0	0
NA	0	0	0	1	1	1

T	Yes	Yes	Yes	Yes	No	No	No	No
R	No	Open	Closed	NA	No	Open	Closed	NA
D Yes	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
No	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5

T	
Yes	0.5
No	0.5

O	
Dry	0.5
Wet	0.3
Soaking	0.2

O	Dry	Wet	Soaking
V	0	Normal (6, 1)	Normal (13.5, 4)

D	Yes	No
Y2	Oil Price * Oil Volume	0

P	Lognormal(2.75, 0.5)

T	Yes	No
Y1	-10	0

D	Yes	No
Y3	Normal(-70, 100)	0

Y4	Y1 + Y2 + Y3

Figure 7 Oil Wildcatter BN with continuous unobservable chance nodes

4.2 Dynamic Discretization (DD) Algorithm

Until recently, the complexity of solving HBNs was a major limitation of using BNs for realistic decision problems. As explained in Section 2, the studies that use BNs to solve IDs also limited their techniques to discrete models for the same reason. However, the DD algorithm (Neil et al., 2007) offers a powerful and flexible solution for the computation of HBNs.

The DD algorithm iteratively discretizes continuous variables by minimizing the relative entropy error between the true and the discretized marginal probability densities. It does this by adding more states to high-density areas and merging states in the zero density areas. At each iteration each continuous variable is discretized, in the area of highest density, and then a standard discrete propagation algorithm, such as JT, is used to calculate the resulting posterior marginals given this

1   discretization. The discretization of each continuous node is revised every time new evidence is

2   entered. The convergence threshold of the DD algorithm sets an upper bound relative entropy that

3   stops the algorithm, and therefore it enables us to set the trade-off between accuracy of the

4   discretization and the speed of computation. The DD algorithm is formally summarized as follows:

5   *Initialize discretization for each continuous variable by partitioning the state space by*

6   *orders of magnitude from 10^{-38} to 10^{38}.*

7   *Set values for* **convergence threshold** *and* **maximum number of iterations**

8   **for** *each iteration until* **maximum number of iterations**

9   *Compute the CPT of each node for the current discretization.*

10   *Enter evidence, and compute propagation, using a standard JT algorithm, and get*

11   *the posterior marginal for each node.*

12   **for** *each continuous node*

13   *Compute the approximate relative entropy error between the true*

14   *probability density function, f, and its discretization by*

15
$$E_j = \left[\frac{f_{max} - \bar{f}}{f_{max} - f_{min}} f_{min} \log \frac{f_{min}}{\bar{f}} + \frac{\bar{f} - f_{min}}{f_{max} - f_{min}} f_{max} \log \frac{f_{max}}{\bar{f}} \right] |\omega_j|$$

16   *where E_j is the approximate relative entropy error, and f_{max}, f_{min}, $\bar{f}$ are the*

17   *maximum, minimum and mean values of the function in a given*

18   *discretisation interval ω_j respectively.*

19   **if** *approximate relative error is smaller than* **convergence threshold**

20   *Stop discretization for this node*

21   **else**

22   *Split the interval with the highest entropy error*

23   *Merge consecutive intervals that have zero entropy error*

24   **end if**

25   **end for**

26   **end for**

27   We illustrate the use of the DD algorithm by using a BN fragment from the wildcatter example that

28   only contains *O* and *V*. This fragment has one continuous node *V* that is a mixture of normal

1   distributions conditioned on a discrete node O. Convergence threshold and maximum number of

2   iterations are used as the stopping rules for the algorithm and were set at 0.01 and 50 respectively.

3   Firstly, the algorithm chooses an initial discretization by dividing the state space to 77 states based

4   on orders of magnitude from 10^{-38} to 10^{38}. In other words, the continuous variable is replaced with

5   a discrete variable that has a single state for each interval in $\{[10^{-38}, 10^{-37}), [10^{-37}, 10^{-36}), \ldots, [10^{36},$

6   $10^{37}), [10^{37}, 10^{38}]\}$. The maximum and minimum values of the initial discretization, i.e. 10^{-38} and

7   10^{38}, are chosen to encompass almost all ranges a user might envisage, so that the user can avoid

8   pre-calculating the domain of the posterior. The CPT of the initial discretization is computed based

9   on the density function of the normal distribution by using standard statistical methods. Table 2

10 shows a part of the initial CPT of V. Note that, the CPD of all other intervals of V that are not shown

11 in Table 2 (i.e. intervals that are smaller than -10000, intervals that are greater than 10000) have

12 zero probabilities.

13 **Table 2 CPT of V at first iteration**

	O	Dry	Wet	Soaking
	...	0	0	0
	-1000 - -10000	0	0	0
	-1000 – -100	0	0	0
	-100 – -10	0	6.39E-58	3.53E-32
	-10 – -1	0	1.27E-12	2.08E-13
	-1 – 0	0	9.85E-10	7.18E-12
	0 – 0	1	0	0
V	0 – 1	0	2.85E-7	1.97E-10
	1 – 10	0	0.999968	0.040059
	10 – 100	0	3.16E-5	0.959941
	100 – 1000	0	0	0
	1000 – 10000	0	0	0
	...	0	0	0

14 Next, the marginal distributions of O and V are computed, given the current discretization, using

15 the JT algorithm (see Figure 8a). The DD algorithm requires a JT propagation at each iteration.

16 The approximate relative entropy error is computed by using the marginal densities computed by

1   the JT and the associated interval widths. Table 3 shows the marginal probabilities and approximate

2   entropy errors for the first iteration. The total approximate entropy error is 0.55, which is greater

3   than the convergence threshold. Therefore, the algorithm divides the intervals with the highest

4   entropy error and proceeds to the second iteration. The intervals $[10 - 100)$ have the highest error,

5   thus the algorithm splits this interval. The algorithm merges all consecutive intervals with zero

6   errors.

7

a) 1st Iteration   **b)** 3rd Iteration   **c)** 32nd Iteration

$\cdots$ $\cdots$

(Error: 0.546)   (Error: 0.109)   (Error: 0.01)

$\mu_{v1}=12.25 \ \sigma_{v1}=20.98$ $\mu_{v2}=4.97 \ \sigma_{v2}=6.22$ $\mu_{v2}=4.50 \ \sigma_{v2}=28.1$
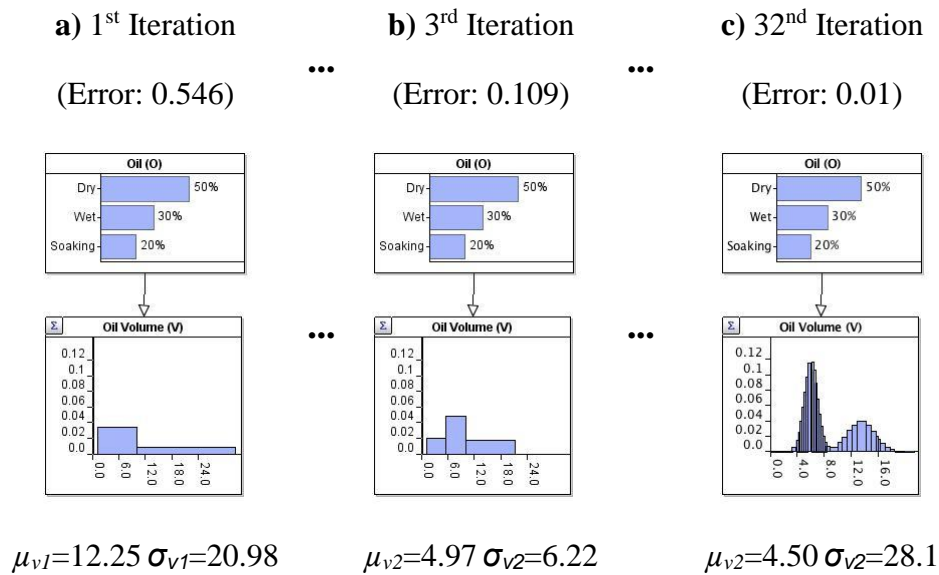
8   **Figure 8 Marginals of BN Fragment in DD iterations**

9   In the second iteration, the algorithm follows the same steps and computes the approximate error

10   as 0.3 (see Table 3). Since this is also greater than the convergence threshold, the algorithm splits

11   $[10 - 55)$. The algorithm merges $[55 - 100)$ and $[100 - 1000)$ intervals, as these consecutive

12   intervals have zero errors. In the third iteration, the algorithm computes the approximate error as

13   0.109 (see Figure 8b). The algorithm continues to revise the discretization until the approximate

14   relative entropy error is less than or equal to the convergence threshold (see Figure 8c). The

15   algorithm stops at the 32nd iteration, requiring 32 JT propagations in total. When the algorithm

stops, the mean and standard deviation of the dynamically discretized V is very close to the

analytical solution of this mixture distribution (i.e. $\mu=4.50$ and $\sigma=28.1$).

Table 3 Discretized Intervals, Marginal Probabilities and Entropy Errors of V

1st Iteration			2nd Iteration			...	32nd Iteration		
Interval	Marginal Prob.	Entropy Error	Interval	Marginal Prob.	Entropy Error		Interval	Marginal Prob.	Entropy Error
...	0	0	-100 – -10	0	0		-100 – -10	0	0
-1000 – -10000	0	0	-10 – -1	4.26E-13	4.25E-10		-10 – -1	4.26E-13	4.25E-10
-1000 – -100	0	0	-1 – 0	2.97E-10	4.72E-11		-1 – 0	2.97E-10	4.72E-11
-100 – -10	0	0	0 – 0	0.5	0		0 – 0	0.5	0
-10 – -1	4.26E-13	4.25E-10	0 – 1	8.57E-08	0.005		0 – 2.1	1.60E-05	2.65E-04
-1 – 0	2.97E-10	4.72E-11	1 – 10	0.308	0.049		2.1 – 3.3	8.78E-04	0.001
0 – 0	0.5	0	10 – 55	0.192	0.246	...	3.3 – 3.8	0.003	0.001
0 – 1	4.46E-13	0.005	55 – 100	0	0		3.8 – 4.1	0.004	3.91E-04
1 – 10	0.308	0.049	100 – 1000	0	0		4.1 – 4.4	0.007	3.65E-04
10 – 100	0.192	0.492					4.4 – 4.7	0.011	4.05E-04
100 – 1000	0	0					...	...	...
1000 – 10000	0						26.9 – 32.5	2.27E-12	1.70E-06
...	0						32.5 – 100	0	0
Total Entropy Error: 0.55			**Total Entropy Error:** 0.30				**Total Entropy Error:** 0.01		

4.2.1 Conditionally Deterministic Functions in DD

Computing approximate inference on conditionally deterministic functions of random variables is

a major limitation of previous techniques for solving HIDs (as we discussed in Section 2.2). The

DD algorithm tackles this problem by modelling them as a mixture of Uniform distributions. We

describe how the approach works by using a fragment of the wildcatter HID that contains P, V and

$Y2$ (see Figure 9):

1. When the wildcatter decides to drill, $Y2$ is defined as a deterministic function of g(P, V),
   i.e. $Y2 = P \times V$.

2. Suppose P and V have discretizations Ψ_P and Ψ_V. For each pair of intervals in the respective
   sets Ψ_P and Ψ_V, such as interval (p_1,p_2) in Ψ_P and (v_1,v_2) in Ψ_V, the approach computes the
   minimum l and maximum u of the set of values $g(p1,v1)$, $g(p1,v2)$, $g(p2,v1)$ and $g(p2,v2)$.

1   3. If *I* is the set of all such pairs of intervals then we get a set of intervals (l_i, u_i) for each $i \in I$,

2   and this generates a Uniform probability density mass, $U(l_i, u_i)$, over the range of *Y2*.

3   Assuming that Ψ_{Y4} is composed of the intervals $\omega_1, \ldots, \omega_n$, then the approximate CPT for

4   the node *Y4* is defined as a weighted Uniform distribution by

5
$$P_{l_i, u_,}(Y2 \in \omega_k) \times U(Y2; l_i, u_i)$$

6   where $P_{l_i, u_,}(Y2 \in \omega_k)$ represents the fraction of the uniform mass $U(Y2; l_i, u_i)$

7   corresponding to the interval ω_k.

8   For example, the initial discretization of *V*, *P* and *Y2* in the first iteration is shown in Table 4. In

9   order to build a CPT for *Y2*, the approach defines the CPD for all combinations of intervals of the

10   *V* and *P*. Suppose we take the intervals V=[-10,1) and P=[0.1,9). In order to define the CPD

11   conditioned on these intervals, we compute g(-10,0.1), g(-10,0.9), g(-1,0.1) and g(-1,0.9), and take

12   the minimum and maximum values of these combinations. Since g(P, V) represents the

13   multiplication operation P × V, the minimum and maximum values are g(-10,0.9) = -9 and g(-1,0.1)

14   = -0.1 respectively. Therefore, P(Y2| V=[-10,-1), P=[0.1,9)) ~ U(-9,-0.1). If a deterministic

15   function has more than two variables, binary factorization is used to prevent the combinatorial

16   explosion of the state combinations (Neil et al., 2012).

17   Next, the approach defines the conditional probabilities corresponding to the intervals of Ψ_{Y2} by

18   using this uniform distribution. Two intervals of Ψ_{Y2}, i.e. [-10,-1) and [-1,0), intersects with the

19   U(-9,0.1) distribution, and the fraction of the uniform mass corresponding to these intervals are P-

20   $_{9,0.1}$(Y2 $\in$ [-10,-1)) = 0.8989 and P$_{-9,0.1}$(Y2 $\in$ [-1,0)) = 0.0011. Table 5 shows a part of the CPT of

21   the *Y2* built from these discretizations. Once, the CPT is built, the DD algorithm is carried out in

1 the same way as described in the previous section. Figure 9 shows the posterior marginals of the

2 *Y2* and its parents computed under different convergence thresholds for DD. The analytic solution

3 of the mean and variance of Y2 is $\mu=90.4$ and $\sigma=154.9$. The DD computes accurate solutions starting
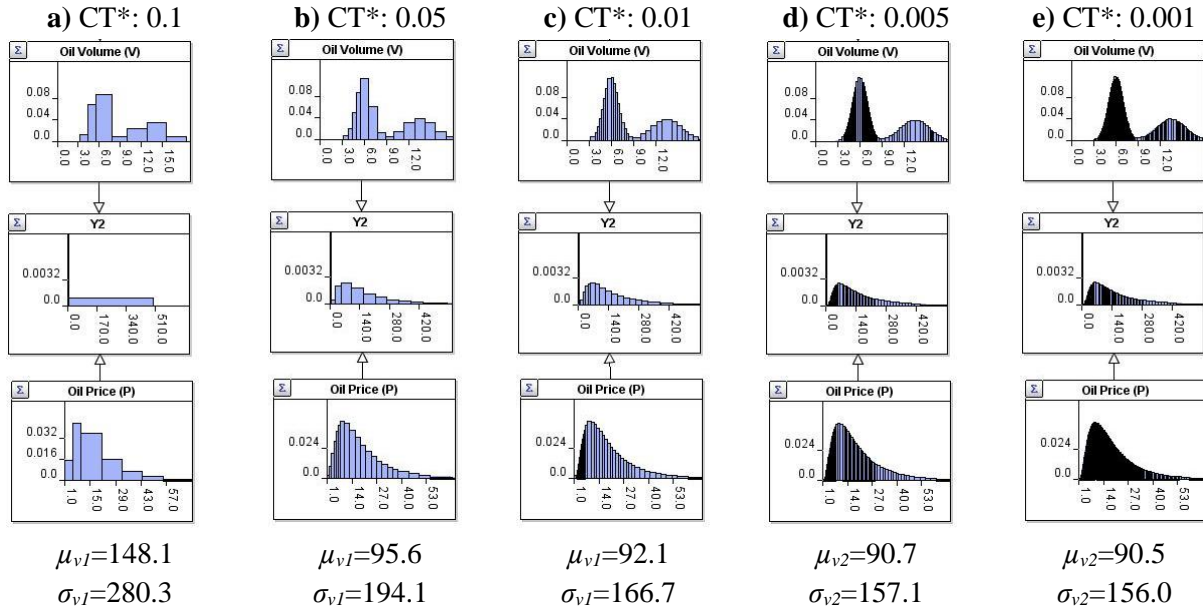
4 from the convergence threshold of 0.01.

5 **Table 4 Initial Discretization of *V*, *P* and *Y2***

V	P	Y2
-100 - -10	0.001 - 0.01	-1E+10 - -10000
-10 - -1	0.01 - 0.1	-10000 - -1000
-1 - 0	0.1 - 0.9	-1000 - -100
0 - 0	0.9 - 0.999999	-100 - -10
0.0 - 1.0	0.999999 - 1	-10 - -1
1.0 - 10.0	1 - 10	-1 – 0
10.0 - 32.5	10 - 55	0 – 0
32.5 – Inf.	55 - 100	0 - 10
	100 - 550	10 - 100
	550 - 1000	100 - 1000
	1000 - 10000	1000 - 10000
	10000 – Inf.	10000 – 1E+5
		1E+5 - 1E+10
		1E+10 - Inf.

6 **Table 5 Fragment of CPT of Y2 in the first iteration**

	V	...	-10 - -1	-10 - -1	...	0 – 1
	P		0.1 – 0.9	0.9 – 0.999		10 -55
Y2	-1E+10 - -10000		0	0		0
	-10000 - -1000		0	0		0
	-1000 - -100		0	0		0
	-100 - -10		0	0		0
	-10 - -1		0.8989	0.9890		0
	-1 – 0		0.0111	0.0110		0
	0 – 0		0	0		0
	0 - 10		0	0		0.1818
	10 - 100		0	0		0.8182
	100 - 1000		0	0		0
	1000 - 10000		0	0		0
	10000 – 1E+5		0	0		0
	1E+5 - 1E+10		0	0		0
	1E+10 - Inf.		0	0		0

1

a) CT*: 0.1   b) CT*: 0.05   c) CT*: 0.01   d) CT*: 0.005   e) CT*: 0.001

$\mu_{v1}=148.1$ $\mu_{v1}=95.6$ $\mu_{v1}=92.1$ $\mu_{v2}=90.7$ $\mu_{v2}=90.5$

$\sigma_{v1}=280.3$ $\sigma_{v1}=194.1$ $\sigma_{v1}=166.7$ $\sigma_{v2}=157.1$ $\sigma_{v2}=156.0$

CT: Convergence Threshold

2   **Figure 9 Posteriors of a Deterministic Function under Different Convergence Thresholds**

3   Readers are referred to (Neil et al., 2007, Lin et al., 2014) for a more detailed description of the

4   DD algorithm's technical details, performance assessments and comparisons with other hybrid

5   inference methods. In a review of HID algorithms, Li and Shenoy (2012) discussed the need for

6   adapting the DD algorithm for solving HIDs. The use of the DD algorithm provides several

7   advantages for solving IDs:

8   • It enables the use of practically any parameterized statistical distribution and conditionally

9   deterministic functions for chance nodes.

10   • The utility nodes can contain continuous distributions, and their linear and non-linear

11   deterministic functions. Since the computations of DD are made with the whole probability

12   distribution, the results are not limited to expected utilities; other useful utility metrics such

13   as variance, credible interval and probability of a positive utility are also calculated.

- It provides the optimal discretization for a given convergence threshold, and these intervals are used to instantiate the observable chance nodes that are parents of decision nodes when computing the optimal decisions.
- The algorithm is implemented in commercial software (AgenaRisk, 2017), and it can be computed in a fully automated way.

4.3 Solving the Oil Wildcatter ID with Continuous Nodes

The solution of the Oil Wildcatter HID by using the DD algorithm is exactly the same as described in Section 3.2 because computation of the continuous observable chance nodes and utility nodes are automatically handled by the DD algorithm when the BN is solved. We again start generating the DT with '*T=Yes*', '*R=No' and 'D=Yes*' scenario, and instantiate these nodes and compute the BN in this order to compute posterior of (*Y4| T=Yes, R=No, D=Yes*). Figure 10 shows the marginal posteriors of the converted BN for this scenario when the convergence threshold is selected as 0.01. The DD algorithm computed the optimal discretizations for the continuous nodes. The expected value of the aggregate utility node, i.e. (*Y4| T=Yes, R=No, D=Yes*), is -40.11 for this scenario, and this is written to the corresponding utility node in the DT. Figure 11 shows the DT generated from this HBN model and the optimal decision policy. Our method computed the optimal decision policy correctly by using the discretization provided by our algorithm.

1
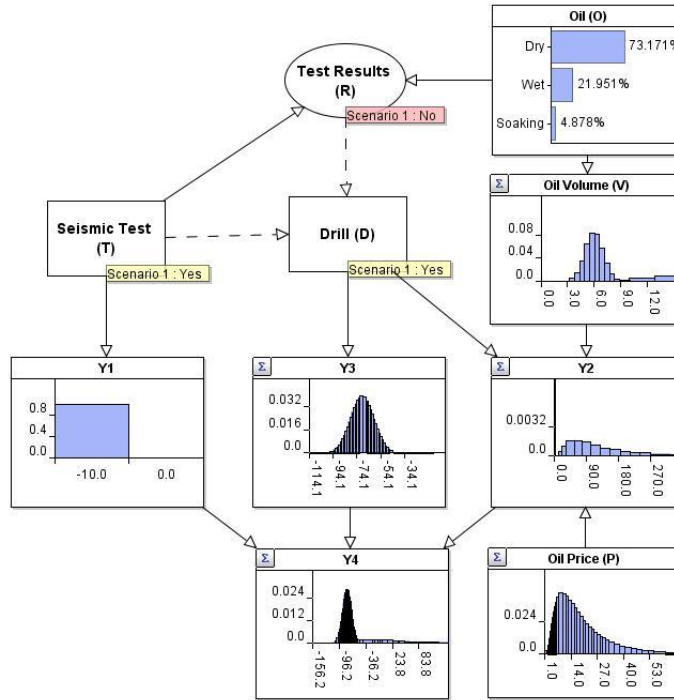
Figure 10 Posteriors of the hybrid Oil Wildcatter BN for *(T=Yes, R= No, D= Yes)* scenario

3
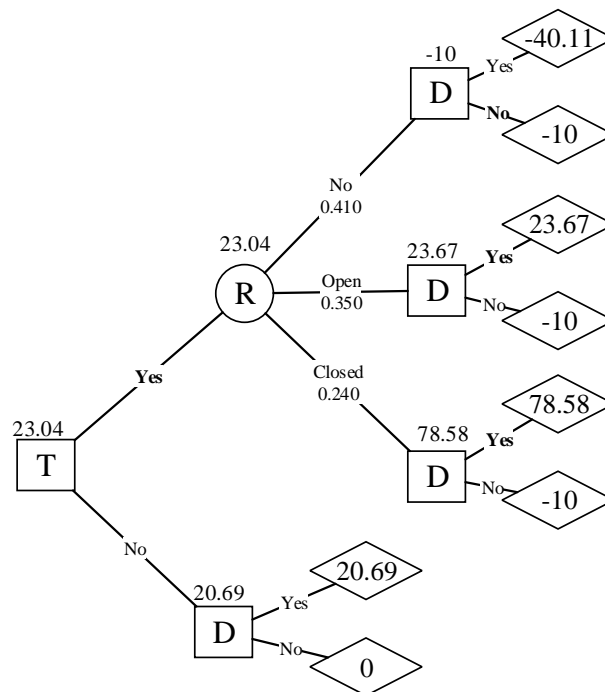
Figure 11 DT generated from Oil Wildcatter BN with continuous unobservable chance nodes

34

4.4 Analyzing both Risk and Expected Utility

A common way to reflect both return and risk into decision analysis is to subjectively map a utility function for different outcomes. Our algorithm offers a richer way to cope with risk as well as expected utility as it makes computations and infers results by using the marginal probability distribution of continuous nodes. Although the DT shown in Figure 11 shows the point values of expected utilities, the underlying BN calculated the marginal utility distribution. Therefore, rather than using only expected utilities to calculate the optimal decision, we can use different measures that combine different summary statistics such as variance and expectation. For example, Figure 12 shows the posterior marginal utility distribution for the decision scenario in which a seismic test has been done, the result of the test is '*Open*', and a hole is drilled. This enables us to generate detailed information about the 'risk' from this distribution, including statistics such as variance, credible intervals, Sharpe ratio (Sharpe, 1994) or *value at risk*, and to use this information together with expected value for decision-making.
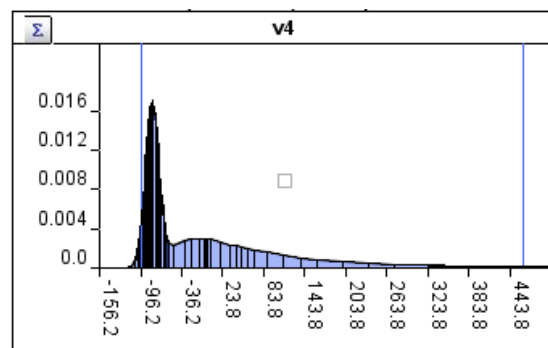
Figure 12 Posterior marginal Utility Distribution for *P(v4 | T=Yes, R=Open, D=Yes)*

For illustration, consider the utility score, used by financial analysts, to combine the expected value and risk (Bodie et al., 2009):

$$U = E(R) - 0.5\,\lambda\,\sigma_R^2$$

1 where $E(R)$ and $\sigma_R{}^2$ are the expected value and the variance of the rate of return, and λ is a measure

2 of the decision maker's risk aversion. The λ value is greater than zero for risk-averse decision

3 makers, is equal to zero for risk-neutral decision makers, and is less than zero for risk-seeking

4 decision makers.

5 In order to use this score, we have to change our utility distributions to rates rather than absolute

6 measures as the score uses expected value and variance of rates of financial return. In our example,

7 we assume that the oil wildcatter has an initial investment of 600 for oil searching and drilling, and

8 we add a node 'R' as a child of 'Y4' with the definition for rate of return, $R = Y4 / 600$. Afterwards

9 we use our algorithm to compute the expected values and variances for R under different decision

10 scenarios.
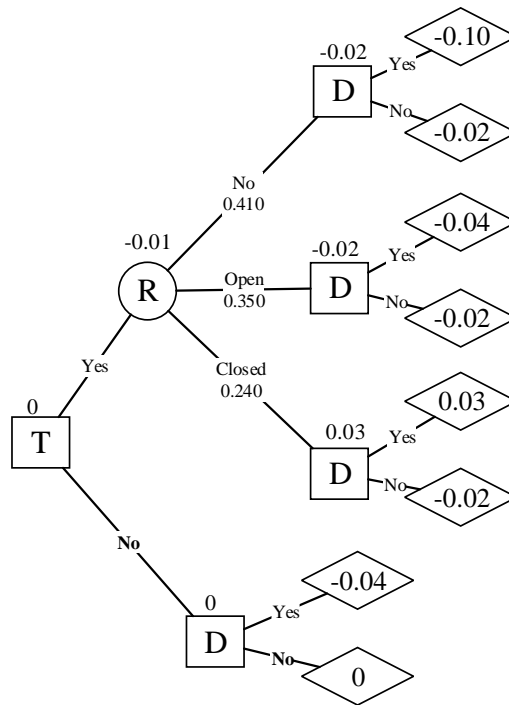
12 **Figure 13 DT for risk averse decision maker with $\lambda = 2$**

13 Figure 13 shows the resulting DT for a risk averse decision maker with $\lambda = 2$. In this figure, the

14 optimal decision is completely different from the case in which utility coincides with the objective

36

1   value. Under these conditions, a risk-averse decision maker avoids drilling and testing due to risks

2   associated with these decisions, even though the absolute expected values were positive.

3   Figure 14 shows the optimal decision strategies for different risk aversion levels, which are

4   completely opposite for highly risk averse and risk seeking decision makers. Each bar in this figure

5   shows the optimal decision strategy for the corresponding λ values under different test result

6   scenarios. While risk seeking decision makers, whose $\lambda \leq -0.39$, prefer to drill without even making

7   a test, highly risk averse decision makers, whose $\lambda \geq 1.52$, neither make a test nor decide to drill.

8   Risk neutral and moderately risk averse decision makers ($-0.39 < \lambda < 1.52$) make drilling decision

9   after observing the results of a test. These decision makers prefer to drill if the test result is 'Wet',

10   and not to drill if the test result is 'No'. If the test result is 'Open', more risk averse decision makers

11   in this category, ($1.45 < \lambda < 1.52$) prefers not to drill while the others ($-0.39 < \lambda \leq 1.45$) prefers to

12   drill.
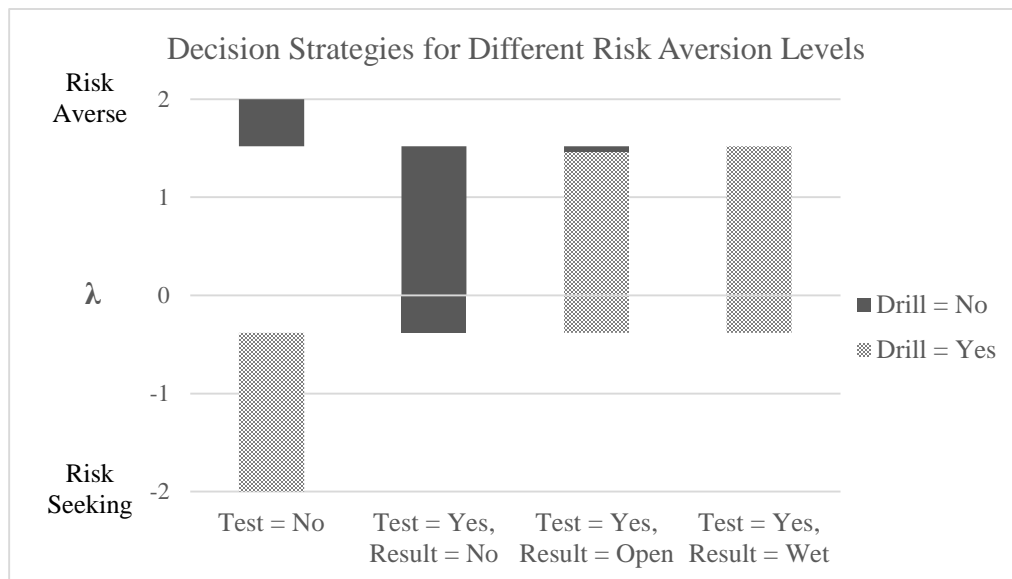
13

14   **Figure 14 Optimal Decision Strategies for Different Risk Aversion Levels**

1 **4.5 Oil Wildcatter Example with Continuous Observable Chance Nodes**

2 Our framework is able to compute models with continuous observable chance nodes and decision

3 nodes. To illustrate the solution of these models, we change the 'Test Results' node to a continuous

4 chance node with the mixture distribution shown in Table 6. This example is taken from Cobb and

5 Shenoy (2004; 2008) so that we can make direct comparisons with their algorithm, other examples

6 of continuous node functions can be easily implemented in AgenaRisk. In this model, the

7 probability of finding no oil increases as the test results are close to 0 and 1, and the probability of

8 finding a large amount of oil increases if they are close to 0.5.

9   **Table 6 Conditional Probability Distribution of Continuous Test Results (R)**

T	Yes	Yes	Yes	No	No	No
O	Dry	Wet	Soaking	Dry	Wet	Soaking
R	Beta(1,1)	Beta(3,3)	Beta(5,5)	0	0	0

10

11 This HID can be solved in our algorithm by discretizing the '*Test Results*' variable and compute it

12 by using the technique described in Section 2. Figure 15 shows a discretization of the '*Test Results*'

13 by using DD with a convergence threshold of 0.01.
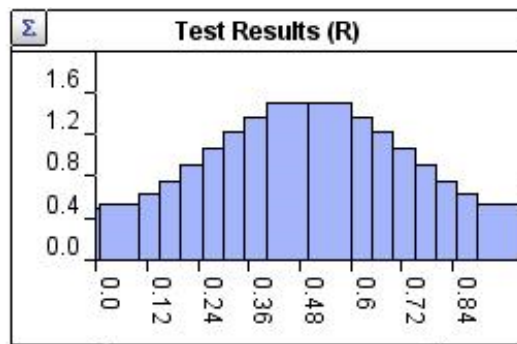
14

15   **Figure 15 Discretization of Test Results by DD**

1 The discretization enables us to solve this ID as shown in Section 3. Table 7 shows a subset of the

2 probability values and expected utilities associated with each decision option and each discretized

3 state of '*Test Results*' computed. For example, when the wildcatter performs the test, the test result

4 is 0.005, and he decides to drill, the expected utility of this combination is -79.965 and P(R = 0.005

5 | T = Yes) = 0.005 as shown in the first row of Table 7. We use these values to compute the optimal

6 decision policy, but if we build a DT for all those state combinations, our DT would have many

7 branches with the same policy. This would make the DT unnecessarily complex. In order to avoid

8 this, we show intervals of the continuous test node where the optimal decision policy is the same

9 rather than showing each discretized state in our DT. Figure 16 shows the expected utilities of

10 'Drill = Yes' and 'Drill = No' when the test result is observable, for all values of the test result.

11 The optimal decision for drilling is 'No' for all test results between 0 and 0.225; therefore, we only

12 draw 1 branch in our DT for this interval. Similarly, the optimal decision for drilling is 'Yes' for

13 all values between 0.225 and 0.775 as shown in Figure 16. The DT and optimal policy is shown

14 in Figure 17.

15 HIDs with continuous decision nodes are also solved in the same way as shown above as DD also

16 provides the optimal discretization of them given a convergence threshold. However, modelling

17 and solving asymmetric continuous decisions by DD include algorithmic research topics that are

18 beyond the scope of this paper.

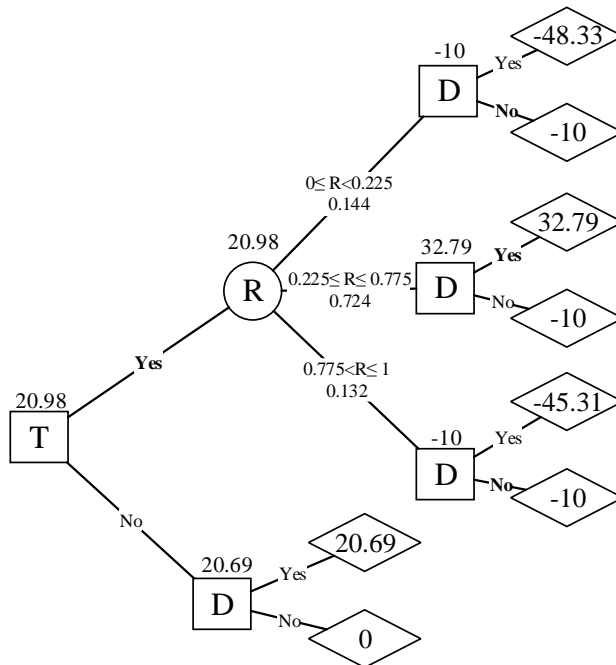19 **Table 7 Values Calculated based on Discretization of the DD algorithm**

T	R	D	P(R|T)	E(v4)
Yes	0.005	Yes	0.005	-79.965
Yes	0.055	Yes	0.048	-72.174
		…		
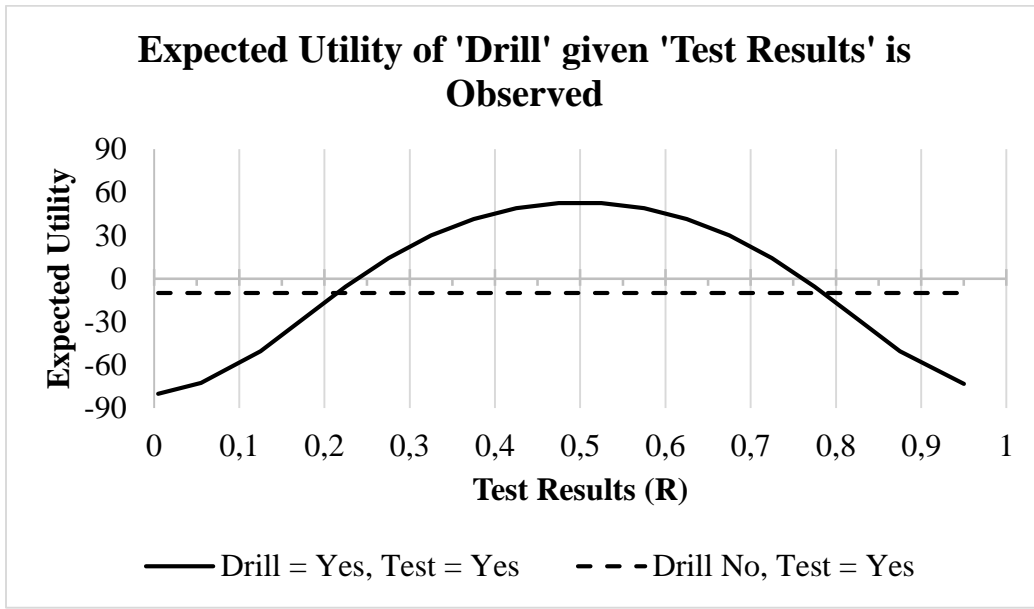
Figure 16 Expected Utility of Drilling Decision given Test Result is Observed

Figure 17 DT generated from the BN with the continuous 'Test Results' node

5 Conclusion

This paper presented a novel method that overcomes a number of key limitations for solving Hybrid

Influence Diagrams (HIDs), which are an extension of IDs containing both and discrete chance

nodes. The state-of-the-art algorithms have severe limitations in computing HIDs especially when dealing with asymmetry, computing parameter uncertainty and deterministic functions of continuous variables. Our method solves a HID by transforming it to a Hybrid Bayesian network (HBN) and propagating this HBN by using Dynamic Discretization (DD). Next, it generates a simplified DT from the propagated BN to compute and present the optimal decisions under different decision scenarios. Our framework offers a number of features to address the limitations of previous HID algorithms:

- To improve efficiency, it uses 'inconsistent evidence' to model functional and structural asymmetry and to avoid further evaluation of inconsistent scenarios.
- It uses the entire probability distributions of the continuous utility and chance nodes, and this enhances decision analysis by offering the possibility to consider statistics other than expected utility alone, such as risk.
- It allows any factorization of the joint utility function, so any liner or non-linear deterministic function and uncertainty of all parameters can be modeled.
- It computes and presents the optimal decision strategy in a simplified DT concisely and effectively.
- The computation of the posteriors of the continuous variables and their deterministic functions can be automatically handled by existing software.

The proposed method propagates the BN for each arc in the simplified DT. Therefore, its complexity grows exponentially with the number of observable chance and decision nodes. Moreover, both the complexity and the accuracy of the DD algorithm increase while the convergence threshold decreases. This enables us to speed up the algorithm with the cost of decreased accuracy.

We illustrated our framework by using the oil wildcatter example and its variations with continuous nodes. In this paper, we used a financial score to combine risk and return measures. However, other measures of risk such as the Sharpe ratio (Sharpe, 1994) or *value at risk* could be easily implemented in our framework. As further research, we plan to examine efficient ways of computing value of information in our framework by using the DD algorithm, and to incorporate decisions with continuous quantities. We also plan to examine the use of multi-criteria decision making methods (MCDM) in HIDs to make a more comprehensive analysis of decision criteria that may conflict such as risk and return.

Acknowledgement

References

AGENARISK. 2017. *AgenaRisk: Bayesian network and simulation software for risk analysis and decision support* [Online]. Available: http://www.agenarisk.com [Accessed 19.07.2017].

BAYESFUSION. 2017. *Genie* [Online]. Available: http://www.bayesfusion.com/ [Accessed 17/06/2017].

BIELZA, C., GÓMEZ, M. & SHENOY, P. P. 2011. A review of representation issues and modeling challenges with influence diagrams. *Omega,* 39**,** 227-241.

BIELZA, C., MÜLLER, P. & INSUA, D. R. 1999. Decision analysis by augmented probability simulation. *Management Science,* 45**,** 995-1007.

BIELZA, C. & SHENOY, P. P. 1999. A comparison of graphical techniques for asymmetric decision problems. *Management Science,* 45**,** 1552-1569.

BODIE, Z., KANE, A. & MARCUS, A. 2009. *Investments,* New York, McGraw-Hill Higher Education.

CABANAS, R., CANO, A., GÓMEZ-OLMEDO, M. & MADSEN, A. L. Approximate Lazy Evaluation of Influence Diagrams. Conference of the Spanish Association for Artificial Intelligence, 2013. Springer, 321-331.

CABANAS, R., GÓMEZ-OLMEDO, M. & CANO, A. 2016. Using binary trees for the evaluation of influence diagrams. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems,* 24**,** 59-89.

CANO, A., GÓMEZ, M. & MORAL, S. 2006. A forward–backward Monte Carlo method for solving influence diagrams. *International Journal of Approximate Reasoning,* 42**,** 119-135.

CHARNES, J. M. & SHENOY, P. P. 2004. Multistage Monte Carlo method for solving influence diagrams using local computation. *Management Science,* 50**,** 405-418.

COBB, B. R. 2007. Influence diagrams with continuous decision variables and non-Gaussian uncertainties. *Decision Analysis,* 4**,** 136-155.

COBB, B. R. & SHENOY, P. P. Hybrid influence diagrams using mixtures of truncated exponentials. Proceedings of the 20th conference on Uncertainty in artificial intelligence, 2004. AUAI Press, 85-93.

COBB, B. R. & SHENOY, P. P. 2005. Nonlinear deterministic relationships in Bayesian networks. *Symbolic and Quantitative Approaches to Reasoning with Uncertainty.* Springer.

COBB, B. R. & SHENOY, P. P. 2006. Inference in hybrid Bayesian networks with mixtures of truncated exponentials. *International Journal of Approximate Reasoning,* 41**,** 257-286.

COBB, B. R. & SHENOY, P. P. 2008. Decision making with hybrid influence diagrams using mixtures of truncated exponentials. *European Journal of Operational Research,* 186**,** 261-275.

COOPER, G. F. A Method for Using Belief Networks as Influence Diagrams. Fourth International Conference on Uncertainty in Artificial Intelligence, 1988 Minneapolis, Minnesota, USA. North-Holland, 55-63.

FENTON, N. & NEIL, M. 2012. *Risk assessment and decision analysis with Bayesian networks*, CRC Press.

HOWARD, R. A. & MATHESON, J. E. 2005. Influence diagrams. *Decision Analysis,* 2**,** 127-143.

HUGIN. 2016. *Hugin Expert* [Online]. Available: http://www.hugin.com [Accessed 31.05.2016].

JENSEN, F., LAURITZEN, S. & OLESEN, K. 1990. Bayesian updating in recursive graphical models by local compuations. *Computational Statistic Quaterly,* 4**,** 269-282.

JENSEN, F. V. & DITTMER, S. L. From influence diagrams to junction trees. Proceedings of the Tenth international conference on Uncertainty in artificial intelligence, 1994. Morgan Kaufmann Publishers Inc., 367-373.

JENSEN, F. V. & NIELSEN, T. D. 2009. *Bayesian networks and decision graphs*, Springer.

JENSEN, F. V. & VOMLELOVÁ, M. Unconstrained influence diagrams. Proceedings of the Eighteenth conference on Uncertainty in artificial intelligence, 2002. Morgan Kaufmann Publishers Inc., 234-241.

KEEFER, D. L. & BODILY, S. E. 1983. Three-point approximations for continuous random variables. *Management Science,* 29**,** 595-609.

LAURITZEN, S. L. & NILSSON, D. 2001. Representing and solving decision problems with limited information. *Management Science,* 47**,** 1235-1251.

LAURITZEN, S. L. & SPIEGELHALTER, D. J. 1988. Local computations with probabilities on graphical structures and their application to expert systems. *Journal of the Royal Statistical Society. Series B (Methodological)***,** 157-224.

LI, Y. & SHENOY, P. P. 2012. A framework for solving hybrid influence diagrams containing deterministic conditional distributions. *Decision Analysis,* 9**,** 55-75.

LIN, P., NEIL, M. & FENTON, N. 2014. Risk aggregation in the presence of discrete causally connected random variables. *Annals of Actuarial Science,* 8**,** 298-319.

LUQUE, M., ARIAS, M. & DIEZ, F. J. Synthesis of strategies in influence diagrams. 33rd Conference on Uncertainty in Artificial Intelligence (UAI 2017), 12 - 14 August 2017 2017 Sydney, Australia.

MADSEN, A. L. & JENSEN, F. V. Lazy evaluation of symmetric Bayesian decision problems. Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence, 1999. Morgan Kaufmann Publishers Inc., 382-390.

MADSEN, A. L. & JENSEN, F. V. 2005. Solving linear-quadratic conditional Gaussian influence diagrams. *International Journal of Approximate Reasoning,* 38**,** 263-282.

MADSEN, A. L. & NILSSON, D. Solving influence diagrams using HUGIN, Shafer-Shenoy and Lazy propagation. Proceedings of the Seventeenth conference on Uncertainty in artificial intelligence, 2001. Morgan Kaufmann Publishers Inc., 337-345.

NEIL, M., CHEN, X. & FENTON, N. 2012. Optimizing the calculation of conditional probability tables in hybrid Bayesian networks using binary factorization. *IEEE Transactions on Knowledge and Data Engineering,* 24**,** 1306-1312.

NEIL, M., TAILOR, M. & MARQUEZ, D. 2007. Inference in hybrid Bayesian networks using dynamic discretization. *Statistics and Computing,* 17**,** 219-233.

NORSYS. 2016. *Netica* [Online]. Available: http://www.norsys.com [Accessed 31.05.2016].

OLMSTED, S. M. 1983. *On Representing and Solving Decision Problems.* PhD, Stanford University.

PEARL, J. 1988. *Probabilistic reasoning in intelligent systems: networks of plausible inference,* San Francisco, CA, Morgan Kaufmann.

POLAND III, W. B. 1994. *Decision analysis with continuous and discrete variables: A mixture distribution approach.* PhD Thesis, Stanford University.

RAIFFA, H. 1968. *Decision analysis: introductory lectures on choices under uncertainty,* Reading, MA, Addison-Wesley.

SHACHTER, R. D. 1986. Evaluating influence diagrams. *Operations research,* 34**,** 871-882.

SHACHTER, R. D. & KENLEY, C. R. 1989. Gaussian influence diagrams. *Management science,* 35**,** 527-550.

SHACHTER, R. D. & PEOT, M. A. Decision making using probabilistic inference methods. Proceedings of the Eighth international conference on Uncertainty in artificial intelligence, 1992. Morgan Kaufmann Publishers Inc., 276-283.

SHARPE, W. F. 1994. The sharpe ratio. *The journal of portfolio management,* 21**,** 49-58.

SHENOY, P. 1992. A new method for representing and solving Bayesian decision problems. *Artificial Intelligence Frontiers in Statistics: AI and Statistics,* 3**,** 119-138.

SHENOY, P. P. & SHAFER, G. Axioms for probability and belief-function propagation. Uncertainty in Artificial Intelligence, 1990. Citeseer.

SMITH, J. E. 1993. Moment methods for decision analysis. *Management science,* 39**,** 340-358.

TATMAN, J. A. & SHACHTER, R. D. 1990. Dynamic programming and influence diagrams. *IEEE transactions on systems, man, and cybernetics,* 20**,** 365-379.

ZHANG, N. L. 1998. Probabilistic inference in influence diagrams. *Computational Intelligence,* 14**,** 475-497.