Bachelor thesis
Computer Science

Radboud University

Towards More Flexible Automatic Decision Making

Author:
Jan Andre Schmidt
4049764

First supervisor/assessor:
Prof.dr. Peter Lucas
peterl@cs.ru.nl

Second assessor:
Dr. Patrick van Bommel
P.vanBommel@cs.ru.nl

July 14, 2016

Abstract

Decision networks and decision trees are typical frameworks that underly most of the systems that are able to make automatic decisions under uncertainty. Decision networks are also known as influence diagrams and apply probabilistic inference among chance variables in a similar way as Bayesian networks. In addition, decision networks contain decision nodes and value nodes with associated utilities. They support automatic decision making by determining those decisions that optimize the expected utility. Current decision networks have some limitations, in the sense that they are not capable of analyzing real-world decision problems that have asymmetric components. In this thesis, some modifications and extensions are proposed for decision networks to facilitate dealing with functional, structural and order asymmetric decision problems.

Contents

Chapter 1

Introduction

In reasoning about situations, one often has to take into account the uncertainty that is associated with the observations on which conclusions are based. It is about acting rationally in a world that we do not know completely or in which we know that various states can hold. In such situation, several aspects influence our argumentation. Artificial intelligence (AI) deals among other things with the analysis of these aspects and their translation to logical expressions that can be interpreted by a machine. Several approaches have been introduced to enable computers to handle reasoning under uncertainty.

This paper deals with decision making and reasoning under uncertainty using a combination of probability, decision and utility theory, and its implementation in decision networks. A decision network is also known as an influence diagram (ID) and was first designed as a tool to represent knowledge [9]. It was developed to bridge the gap between the analysis and formulation and was meant to help non-experts to get used to decision systems. Originally decision system were essentially decision tree implementations used to illustrate and solve decision problems. Nowadays, decision networks are interpreted using algorithms that support solving decision networks directly.

One preparation to analyze a problem with a decision system is to build a model of the real world. In this modelling stage, the decision maker is asked to initialize the required utilities and probabilities. To set the utilities he or she has to evaluate possible outcomes on a numerical scale, not only to express the order but also the relative distances between them in line with his or her preferences. The probabilities represent the uncertainties in the real world and they address the degree of belief in a particular state. Except for the probabilities based on frequencies, the mentioned specifications lie in the eye of the beholder. Hence, the modelling phase obeys a lot of subjectivity going out from the decision maker that may lead to different results for the same decision problem. In this thesis, however, we do not question the correctness of the product of the modelling stage.

Instead, our starting point is a completely initialized influence diagram, a framework that we will study and investigate for its ability to handle different types of decision problems. One property of the network is its straightforward way to demonstrate relations among components. However, this feature usually demands that the decision problem is symmetric. In the following paragraph, we will examine three scenarios that will introduce three types of asymmetry that may occur in decision problems. The example is devised and unlikely to match reality.

Suppose we are asked to implement an automatic decision system for a transplantation department of a hospital. To get a first insight we ask one of the surgeons to give us a walkthrough for the case that a brain dead organ donor patient enters the operating room. He informs us about numerous tests before a patient is decided to be brain death. With the aid of their results, the attendant doctor declares the brain death and detects its cause. The next step is to encounter if the organs are qualified to be donated. Once again various tests are required. Depending on the cause of death some of the tests become unnecessary. This is an example of *functional asymmetry.* It indicates that new information can inhibit later decisions.

Furthermore, the doctor explains that it also may be the case that the sequence of the tests changes to receive accurate results. This is called *order asymmetry.* Its intention is that the decision maker can define the order of some decisions during the decision process.

Of course, it also may happen that some test results lead to the consequence that particular organs do not come into question for a donation. In this case, no additional tests are needed. This type of asymmetry is named *structural asymmetry.* It indicates that observed information effects the termination of the remaining decision problem.

For each of the three introduced types of asymmetry, there is no straightforward way to design a proper model and use decision networks for their analyzation. Hence, we will investigate possible extensions of the influence diagram to facilitate the support for this kind of asymmetric decision problems.

The thesis is subdivided into five remaining chapters. The succeeding chapter is about the preliminaries, it gives a review of probability, decision and utility theory and introduces the predecessors of decision networks, in particular, Bayesian networks (BN) and decision trees. The third chapter gives a definition of IDs and illustrates three different algorithms to solve them. Additionally, we examine the drawbacks of influence diagrams and make a comparison with decision trees. In chapter 4, potential alternations and extension are introduced for decision networks to facilitate their transformation to a Bayesian network and the representation of asymmetries. The last chapter 5 provides an conclusion.

Chapter 2

Preliminaries

In this chapter, we review the foundations of decision networks consisting of a graph, probability, utility and decision theory part and establish some notations used in this paper.

2.1 Graph Theory

Graph theory defines one of the basic concepts to create probabilistic graphical models (PGMs) and decision network representations. Graph representations have been shown to be particularly useful for explaining probabilistic reasoning and decision-making to non-experts[10].

Nowadays, the models themselves have become a research subject. This chapter provides general graph characteristics serving as a foundation for upcoming detailed definitions in the subsequent chapters.

A *graph* G is defined as a pair, consisting of a set of nodes $N(G)$ and a set of edges $E(G) \subseteq N(G) \times N(G)$, i.e.

$$G = (N(G), E(G))$$

An *edge* is a pair of two nodes and marks a connection between them. An edge $e_1 = (n_1, n_2)$ that is *directed*, i.e. $(n_1, n_2) \in E(G)$ and $(n_2, n_1) \notin E(G)$, represents an arrow or arc from n_1 to n_2. It establishes the relation that node n_1 is a *parent* node of n_2. The set of parents of a node n_i is notated as $par(n_i)$ and the set of *children* of a node n_i is notated as $chi(n_i)$.

Edges can be directed or undirected. So called Markov networks only contain undirected edges and represent a join probability distribution with independence information. However, in this thesis we are only concerned with PGMs that have directed edges, i.e. Bayesian networks and decision networks.

Another characteristic of a PGM is that it has to be *acyclic*. Hence, the directed edges are not allowed to form a directed cycle with a $path(n_1, n_1)$

having a start node that is also an end node. Graphs fulfilling these premises are called *directed acyclic graphs* (DAGs).

2.2 Probability Theory

Reasoning under uncertainty uses probability theory to measure and calculate beliefs of whether a variable in the domain has a particular value or not. There are two different interpretations for these measurements. The first is to interpret probabilities in terms of the frequency at which an outcome occurs over the long term in relation to the frequency of all possible outcomes, referred as relative frequency. The second interpretation is to assign a degree of belief based on reasoned arguments to a variable for which certain knowledge is impossible, i.e. in court cases or medical diagnosis, to declare some degree of belief whether a defendant is guilty or a a disease is present. These probabilities are referred to as subjective probabilities [4] [5]. The probability value of a certain outcome is defined to be on the interval [0,1] and all probabilities of possible outcomes of a random variable sum up to 1. We will express random variables with capital letters, e.g., X and binary states with the corresponding small letter, e.g. x for $X = true$ and $\neg x$ for $X = false$.

Two rules linked to probability theory are crucial in understanding the principles of probabilistic reasoning algorithms in probabilistic and decision-theoretic networks. These rules are referred to as *marginalization* and *conditioning* and are used to derive any probability of interest out of a joint probability distribution in a Bayesian network. Joint probabilities define the probability of intersections of single outcomes of multiple chance variables. A joint probability distribution states all associated combinations in a table. Suppose we have the two chance variables X and Y, that can either be true or false, and the following probability distribution $P(X, Y)$.

X	Y	$P(X,Y)$
x	y	0.15
x	$\neg y$	0.45
$\neg x$	y	0.1
$\neg x$	$\neg y$	0.3

From this joint probability distribution we can retrieve the probability distributions X and Y. Therefore, the probabilities of the same variable value or state are summed. This can also be done to get joint probability distributions. It is called *marginalization* and variables that do not appear in the resulting probability distribution are called to be summed out.

$$P(X) = \sum_{y \in Y} P(X, y)$$

Applying marginalization for X gives us $P(x) = 0.6$ and $P(\neg x) = 0.4$.

Besides joint probabilities, conditional probabilities are used for probabilistic reasoning. A *conditional probability* is the probability of a chance variable as the value of another chance variable has already been observed and everything else with exception of this observation is irrelevant. Suppose we observed that $X = x$, and we want to know the recent probability of $Y = y$. Then the conditional probability of y given x is the joint probability of x and y divided by the probability of x, indicated as $P(y|x)$.

$$P(y|x) = \frac{P(x, y)}{P(x)} = \frac{0.15}{0.6} = 0.25$$

If some observation for X does not change our belief of Y, we say that Y is *independent* of X, written as $Y \perp\!\!\!\perp X$. In this case, the probability of Y is the same as the probability of Y given X.

$$P(Y) = P(Y|X)$$

This independence relation is symmetric. Hence, if Y is independent of X, X is also independent of Y.

$$X \perp\!\!\!\perp_P Y \leftrightarrow Y \perp\!\!\!\perp_P X$$

Furthermore, it may occur that two dependent chance variable X and Y become free from each others impact as an observation has been made on a third chance variable Z. This relation between chance variables is called *conditional independence*. It states that the conditional probability of X given Y and Z and X given Y are the same. As the concept of independence, conditional independence is also symmetric.

$$X \perp\!\!\!\perp_P Y|Z \leftrightarrow P(X|Y, Z) = P(X|Z)$$
$$X \perp\!\!\!\perp_P Y|Z \leftrightarrow Y \perp\!\!\!\perp_P X|Z$$

2.3 Bayesian Networks

A Bayesian network (BN) is a PGM defined as a pair $BN = (G, P)$, with G as a directed acyclic graph $G = (N(G), E(G))$ and a joint probability table P. It is a compact, expressive representation of uncertain relationships among parameters in a domain[3].

Decision networks apply it as a probabilistic framework. The starting point is a joint probability table P over the chance variables $X =$

$\{X_1, ..., X_n\}$. The size of the table is 2^n for binary state spaces and represents the full joint probability distribution given by:

$$P(X) = \prod_{i=1}^{n} P(X_i|par(X_i)).$$

Each variable of the joint probability is represented as a chance node. The data behind it depends on the number of parent nodes. For root nodes the data is just the probability distribution of the chance variable. Whereas for nodes having parent nodes, the data is represented as a conditional probability table. The linked conditional probability table states the conditional probabilities of the node given its parents. The size of such a table grows exponentially with the increasing number of parents nodes. As we assume that false values are deduced by $1 - P(x)$, the number of entries for binary state spaces that had to be specified for a chance variable with k parents is equal to 2^k.

A BN uses the independence and conditional independence relations among chance variables to deduce the data of the full joint probability distribution without loss of information. To address the independence of two nodes A and B, we will write $A \perp\!\!\!\perp_G B$. This independence relation is valid for the two nodes in a graph if they are d-separate. D-separation states that each path in between has been declared to be inactive. An inactive path is a queue of nodes that contains three sequenced nodes and linked edges that build up to a chain equivalent to one of the three types serial, convergent or divergent.

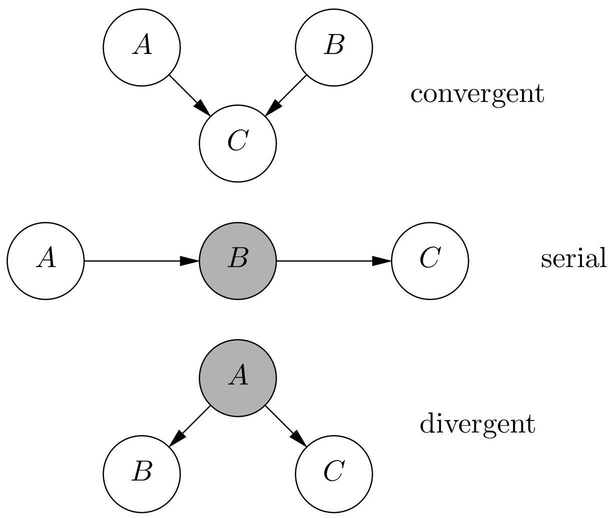
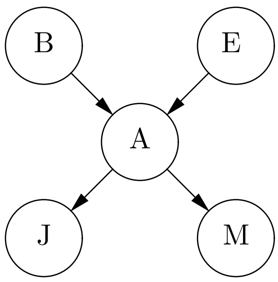


Figure 2.3.1: Three types of edge connections

The joint probability table required for a BN is specified by:

$$P(X) = \prod_{i=1}^{n} P(X_i|par(X_i)).$$

A famous example to illustrate the idea of a BN is a probability distribution linked to the reliability of a burglary alarm. The alarm also responds to earthquakes and it is reported by two neighbors, John and Marry, calling when they hear the alarm. Marry does not always hear the alarm, whereas John hears the alarm nearly every time, but sometimes calls when there is no alarm. The letters B, E, A, J and M stand for burglary, earthquake, alarm, John calls and Marry calls respectively. Figure 2.3.2 shows the structure of the corresponding BN on the left side and the probabilistic data of the chance nodes on the right side.



$P(B)$	
b	0.01

$P(E)$	
e	0.02

$P(A\|B,E)$				
	b		$\neg b$	
	e	$\neg e$	e	$\neg e$
a	0.95	0.94	0.29	0.01

$P(J\|A)$		
	a	$\neg a$
j	0.9	0.5

$P(M\|A)$		
	a	$\neg a$
m	0.7	0.1

Figure 2.3.2: Burglar alarm example

With a BN it is possible to compute any probability of interest out of a joint probability. If we want to know how likely it is that Marry and John call although there is nor burglary nor earthquake, we can calculate the probability $P(j, m|\neg b, \neg e)$ by using conditioning and marginalization. The scencario that there is no burglary and no earthquake is therefore inserted as an evidence $e = \{\neg b, \neg e\}$. An evidence is a finding on an chance variable that makes other states than the evidence itself impossible. Consequently, impossible states are summed out of the probability distribution.

The first step is to get the joint probability of $P(\neg b)$ and $P(\neg e)$.

$$P(\neg b, \neg e) = 0.99 \times 0.98 = 0.9702$$

The second step is to get the probability distribution of $P(A|\neg b, \neg e)$.

$$P(a|\neg b, \neg e) = 0.9702 \times 0.01 = 0.009702$$
$$P(\neg a|\neg b, \neg e) = 0.9702 \times 0.99 = 0.960498$$

The third step is to compute the probabilities $P(j|a, \neg b, \neg e)$,$P(j|\neg a, \neg b, \neg e)$ and $P(m|a, \neg b, \neg e)$, $P(m|\neg a, \neg b, \neg e)$.

$$P(j|a, \neg b, \neg e) = 0.009702 \times 0.9 \approx 0.009$$
$$P(j|\neg a, \neg b, \neg e) = 0.960498 \times 0.5 \approx 0.480$$
$$P(m|a, \neg b, \neg e) = 0.009702 \times 0.7 \approx 0.007$$
$$P(m|\neg a, \neg b, \neg e) = 0.960498 \times 0.1 \approx 0.096$$

The fourth step is to sum out A.

$$P(j|\neg b, \neg e) = 0.009 + 0.480 = 0.489$$
$$P(m|\neg b, \neg e) = 0.007 + 0.096 = 0.103$$

The last step is to get the joint probability of $P(j|\neg b, \neg e)$ and $P(m|\neg b, \neg e)$.

$$P(j, m|\neg b, \neg e) \approx 0.05$$

Hence, there is a 5% chance of being called by John and Marry although there is nothing to worry about.

2.4 Decision Trees

A decision tree is a directed acyclic graph G, that is read downward from its root node. Except for the root node every node has exactly one parent node. The graph represents a decision problem by illustrating every scenario in an individual sequence of chance and decision nodes ending up in an utility node. Therefore, it only represents temporal ordering and no conditional dependencies among nodes. For every state or decision alternative, there is an outgoing edge from a chance or decision node, respectively. Accordingly, a chance node has as many children as states, and a decision node has as many children as decision alternatives.

Consider the decision problem as to whether or not to go ahead with a fund raising garden party. If we go ahead with the party and it subsequently rains, then we will lose money (since very few people will show up); on the other hand, if we don't go ahead with the party and it does not rain we are free to go and do something else fun[1]. The chance variable rain is notated as R. The set of actions A consists of the action y stating that there is an party and action n stating that there is no party. The possible outcomes O

are the combinations of A and R. The utility function u maps the outcomes to an utility value. The formal description of the decision problem is as follows.

$$P(r) = 0.6, P(\neg r) = 0.4$$

$$A = \{y, n\}$$

$$O = \{\{y, r\}, \{y, \neg r\}, \{n, r\}, \{n, \neg r\}\}$$

$$\begin{gathered} u(\{y, r\}) = -100 \\ u(\{y, \neg r\}) = 500 \\ u(\{n, r\}) = 0 \\ u(\{n, \neg r\}) = 50 \end{gathered}$$

The corresponding decision tree is shown in figure 2.4.1.

To solve a decision tree an optimal policy is defined for every decision node. Therefore, every edge of a decision alternative is linked to the expected utility of its children nodes. The expected utilities are notated in brackets.

$$EU(d = y) = -100 \times 0.6 + 500 \times 0.4 = 140$$

$$EU(d = n) = 0 \times 0.6 + 50 \times 0.4 = 20$$

This is done repeatedly from the utility nodes to the root node. Consequently, the optimal policy of a decision node corresponds to the decision that maximizes the expected utility specified as MEU.

$$MEU(D) = 140(d = y)$$

The optimal strategy for multi decision problems consists of the optimal policies of the decision nodes from the root to leaf node. The optimal strategy is indicated by a boldfaced path in the tree.

2.5 Utility Theory

Which consequences do we prefer or avoid while making a decision? Only the decision maker can answer this question. Utility theory is about the form of the evaluation of outcomes to make them comparable and available for automatic reasoning. Therefore, a utility function $U(o, d_s)$ maps all outcomes $o \in O$ together with the instantiated sequence of decisions d_s onto a utility scale. The higher the assigned utility value on the scale, the more useful is the outcome and the linked decisions to the decision maker. As the occurrence of an outcome is linked to a set probability, the utility values to be compared have the form of lotteries. A lottery $l \in L$ is a set of outcomes

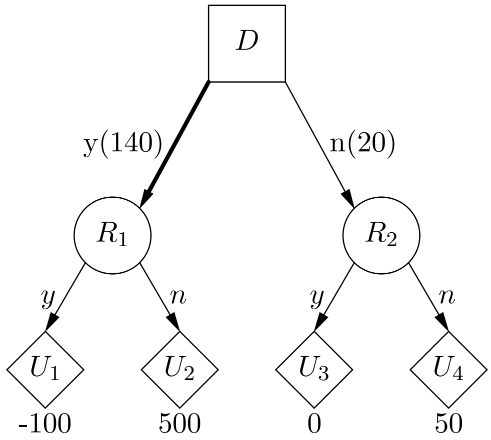


Figure 2.4.1: Decision Tree

o_i and its probability p_i. The probabilities of all outcomes sum up to 1. To compare two lotteries l_1 and l_2 we will use the following notations:

- $l_1 > l_2$, if l_1 is preferred over l_2.
- $l_1 \sim l_2$, if l_1 and l_2 are indifferent.
- $l_1 \geq l_2$, if l_1 is preferred over l_2 or l_1 and l_2 are indifferent.

To guarantee that an lottery l_1 is preferred to another lottery l_2 if and only if $l_1 > l_2$, the preferences have of the decision maker have to agree with the axioms of utility theory[4, 6, 8]:

- **Orderability:**
  $\forall l_1, l_2 \in L, (l_1 > l_2) \vee (l_1 > l_2) \vee (l_1 \sim l_2)$

  Orderability affirms that a statement about the preference is either one lottery is preferred, or both lotteries are preferred equally. It is not conceivable that there is no preference.

- **Transitivity:**
  $\forall l_1, l_2, l_3 \in L, (l_1 > l_2) \wedge (l_2 > l_3) \rightarrow (l_1 > l_3)$

  If lottery l_1 is preferred to lottery l_2 and l_2 is preferred to lottery l_3, then l_1 is also preferred to l_3.

- **Continuity**
  $\forall l_1, l_2, l_3 \in L, l_1 > l_2 > l_3 \Rightarrow \exists p[p, l_1; 1-p, l_3] \sim l_2$

  If l_1 is preferred to l_2 and l_2 is preferred to l_3, then there exists a probability p for l_1 and $1-p$ for l_3 so that both lotteries together are equally preferred as l_2.

- **Substitutability:**
  $\forall l_1, l_2, l_3 \in L, l_1 \sim l_2 \Rightarrow [p, l_1; 1-p, l_3] \sim [p, l_2; 1-p, l_3]$

  If l_1 and l_2 are equally preferred, then their combinations with a third lottery l_3 with same probabilities are also equally preferred.

- **Monotonicity**
  $\forall l_1, l_2 \in L, l_1 > l_2 \Rightarrow (p > q) \Leftrightarrow [p, l_1; 1-p, l_2] > [q, l_1; 1-q, l_2]$

  If l_1 is preferred to l_2 and probability p is higher than probability q, then the lottery that links p to l_1 is preferred to lotteries that link q to l_1.

- **Decomposability**
  $\forall l_1, l_2, l_3 \in L,$
  $[p, l_1; 1-p, [q, l_2; 1-q, l_3]] \sim [p, l_1; (1-p)q, l_2; (1-p)(1-q), l_3]$

  If l_1 is linked to probability p and l_2 is linked to $1-p$ and q and l_3 is linked to $1-p$ and $1-q$, then l_2 and l_3 can be separated form l_1 without changing the preference of the three lotteries together.

2.6 Decision Theory

Decision theory combines the preferences expressed by utilities with the probabilities of chance variables. The fundamental idea is that the decision maker acts rational if and only if he makes a decision that yields the highest expected utility, averaged over all possible outcomes of decisions.

$$\text{Decision theory} = \text{Probability theory} + \text{Utility theory}[8]$$

A decision D_i of a decision problem comprises an set of decisions $D_i = \{d_1, ..., d_n\}$. The expected utility (EU) of an decision d_i, is the sum of the utilities of outcomes weighted by their probabilities given that d_i is chosen.

$$EU(D_i = d_i) = \sum_{o \in O} U(o) \times P(o|d_i)$$

The decision to be chosen to act rational is the one with the highest expected utility defined by the maximum expected utility (MEU).

$$MEU(D_i) = \max_{d \in D_i} EU(D_i = d)$$

In a decision problem with only one decision made the MEU already gives us the optimal solution.

As multiple decisions are involved, an optimal solution has to be found for a sequence of decisions. To define the partial temporal orderings we will use the operator $\prec$. A sequence of n decisions follows the concept that a decision D_i is made before a decision D_j if $i < j$, i.e. decision D_1 is made before decision D_2.

$$D_0 \prec D_1 \prec ... \prec D_{n-1} \prec D_n$$

A solution to a multiple decision problem is called strategy. It is a set of policies with a policy δ_i for every decision D_i. A policy δ_i maps any combination of states rising because of previous decisions to an decision d_i. A strategy that maximizes the expected utility is called an optimal solution and every policy involved is called an optimal policy [4].

Chapter 3

Decision Networks

This chapter introduces decision networks also known as influence diagrams.

3.1 Background on Decision Networks

A *decision network*, DN for short, is defined as a triple DN $= (G, P, U)$, with $G = (N(G), E(G))$ a directed acyclic graph, P a joint probability distribution structured as an Bayesian network, and U a utility function. $N(G)$ is a set of nodes, where $C(G) \subseteq N(G)$ are *chance nodes*, introduced in the section about Bayesian networks, $D(G) \subseteq N(G)$ are *decision nodes*, linked to decision theory and $V(G) \subseteq N(G)$ is the set of *value nodes* holding the utility function U. The chance, decision and value nodes are mutually exclusive. Thus,

$$N(G) = C(G) \cup D(G) \cup V(G)$$

Furthermore, $E(G) \subseteq N(G) \times N(G)$ is a set of *arcs* with some restrictions that will become clear in the following.

All nodes have already been part of the formerly introduced decision trees. In decision networks their meaning is slightly different. The nodes are shown in figure 3.1.1.

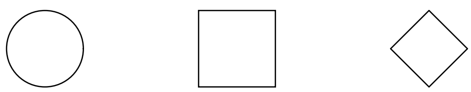


Figure 3.1.1: chance, decision and value node

The advantage of decision networks over decision trees is that they offer a much more compact representation. Figure 3.1.2 shows the influence diagram for the party decision problem introduced in the section about decision trees.

For multi decision problems the diagram requires a temporal ordering of nodes in order to determine the set of chance nodes observed before a

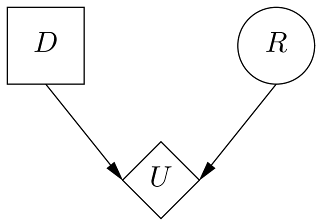


Figure 3.1.2: Influence diagram

decision D_i. The set of chance nodes observed before the first decision is notated as X_0 and the set of chance nodes observed after a decision D_i and before the decision D_{i+1} are notated as X_i. For n decision nodes the set of chance variables not observed before any decision is given by X_n.

$$X_0 \prec D_1 \prec X_1 \prec ... \prec D_n \prec X_n$$

Consider the following oil wildcatter problem [7]. The decision is whether or not to test the seismic structure (D_1) and whether or not to drill for oil (D_2). The utility (U) is in line with the profit that is made, the amount of oil (A), the costs of the test and the decision to whether or not to drill. This decision can be influenced by the test results (R), depending on the seismic structure (S). The seismic structure itself depends on the amount of oil. The influence diagram is shown in figure 3.1.3.

$$D_1 \prec \{R\} \prec D_2 \prec \{A, S\}$$

In an influence diagram the order of the decision nodes is defined by a path comprising all decisions, $D_1 \rightarrow R \rightarrow D_2$. Based on its target an edge can either be informational, probabilistic or functional. Informational edges are pointing towards decision nodes and state that the information of the child node is accessible when the decision is made. As an informational edge only states that an information of a decision is available for a succeeding decision, it is called a no forgetting edge. Probabilistic edges have the same meaning as the edges in Bayesian networks, they are pointing towards chance nodes and define probabilistic dependencies. Functional edges are outgoing edges from chance or decision nodes pointing towards utility nodes. The state space of the parent node increases the set of outcomes and affects the utility values that are given by the utility function.

One drawback of an ID is that all scenarios, hidden in the network, have to be examined to solve it. Several ways have been invented to deal with this problem, in the following subchapters, we will investigate three of them. The first way to investigate is the creation of the corresponding decision tree representation and to apply the rolling back algorithm. The second way is to transform the ID to a BN and solve it by instantiating all possible sequences of decisions. The third way is an algorithm that works with the

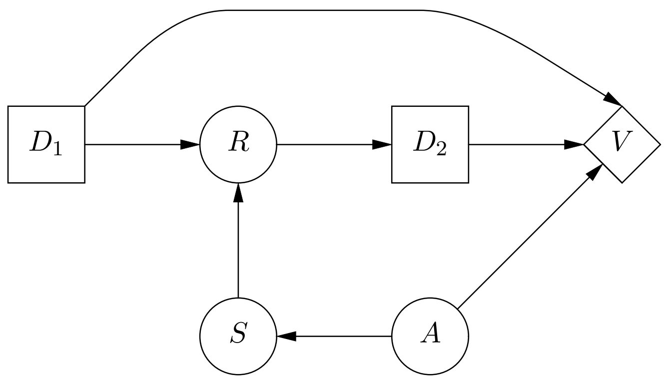


Figure 3.1.3: Influence diagram

ID itself. It"folds" the network up until one value node remains. We will investigate the different ways of solving an ID with the introduced wildcatter oil problem. Therefore, we will come up with numerical values for the decision problem, consisting of probability distributions, held by the chance nodes, and the utility values for all scenarios given by the utility function, encountered in the value node. Here are the probability distribution for the nodes amount of oil (A), seismic structure (S) and test result (R).

$P(A)$	
l	0.7
m	0.3

$P(S\|A)$		
	l	m
os	0.85	0.05
cs	0.15	0.95

$P(R\|S)$		
	os	cs
p	0.03	0.99
n	0.97	0.01

The small letters l and m are standing for little and much oil respectively. The seismic structure can be open (os) or closed (cs). A closed seismic structure indicates a higher amount of oil. The utility function has two inputs, an outcome and an instantiated sequence of decisions. There are three chance nodes with two states leading to 2^3 possible outcomes.

O	A	S	R
o_1	l	os	p
o_2	l	os	n
o_3	l	cs	p
o_4	l	cs	n
o_5	m	os	p
o_6	m	os	n
o_7	m	cs	p
o_8	m	cs	n

The table below lists all inputs and the resulting utility value. The utility value solely depends on the amount of oil and only if the decision is made to drill $D2 = y$.

O	$D1$	$D2$	$U(O, d_1, d_2)$
$o_1 - o_4$	y	y	$-35,000\$$
$o_1 - o_4$	n	y	$-30,000\$$
$o_5 - o_8$	y	y	$145,000\$$
$o_5 - o_8$	n	y	$150,000\$$
$o_1 - o_8$	y	n	$-5000\$$
$o_1 - o_8$	n	n	$0\$$

Hence, we can gather the outcomes with the same instantiations for A. If the decision is not to drill $D2 = n$, the instantiations of the chance nodes have no affect on the utility value.

3.2 From Influence Diagram to Decision Tree

One method to solve a decision network is to use the corresponding decision tree. The first step is to model the decision tree based on the knowledge about the decision problem. A direct transformation from a decision network is only reasonable if the decision problem is symmetric. In this case, we would receive a symmetric tree structure with the same order of nodes in each path from the root to the leaves.

To build up a decision tree the temporary order of the nodes is taken into consideration. The earlier a decision is made, or a chance node is observed; the higher is its position in the tree. The introduced wildcatter oil problem results in the decision tree shown in figure 3.2.1, figure 3.2.2 and figure 3.2.3. Three figures are needed, as the decision tree grows exponentially with the number of nodes involved.

The decision problem obeys two structural asymmetries. They occur after the two decisions D_1 and D_2. The chance node results R solely exist if we decide to make a test, and there will only be an impact of the chance nodes amount of oil A and seismic structure S if we decide to drill. The tree illustrates the corresponding asymmetries.

Once the decision tree is available, we can solve it with the aid of the average out and fold back algorithm. To get the expected utility of a subtree, with root node X, we can apply algorithm 1.

Algorithm 1 average out and fold back algorithm

if $X \in V$ **then**
  return

$$U(X)$$

else if $X \in C$ **then**
  return

$$EU(X) = \sum_{x \in X} P(X = x|par(X))EU(chi(X = x)))$$

else if $X \in D$ **then**
  return

$$EU(X) = \max_{x \in X} EU(chi(X = x))$$

  and label the corresponding edge x' with

$$\arg\max_{x \in X} EU(chi(X = x))$$

end if

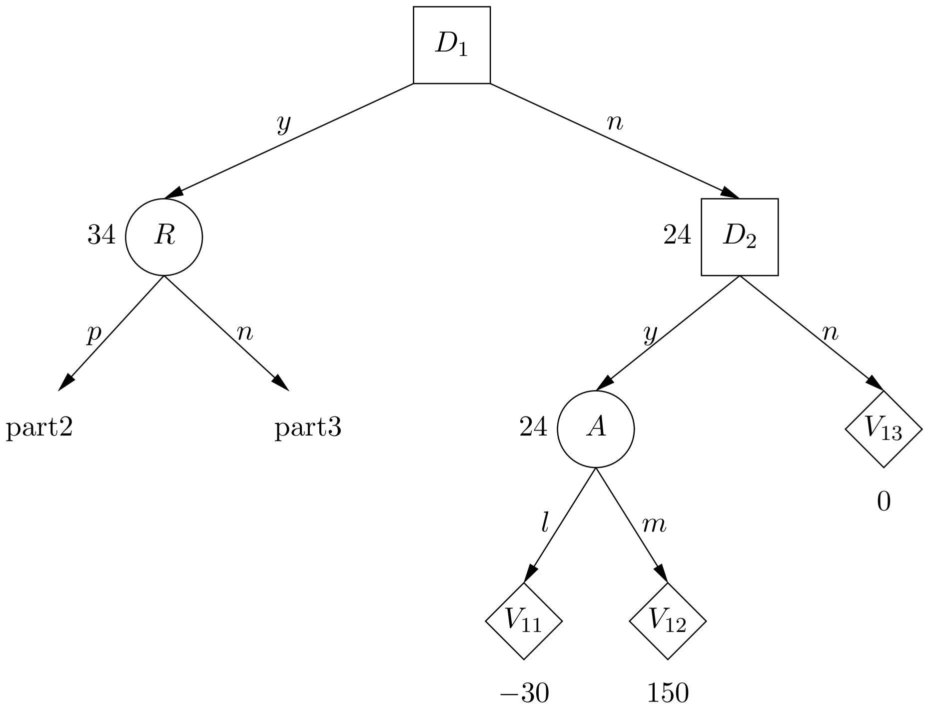


Figure 3.2.1: Decision tree part 1

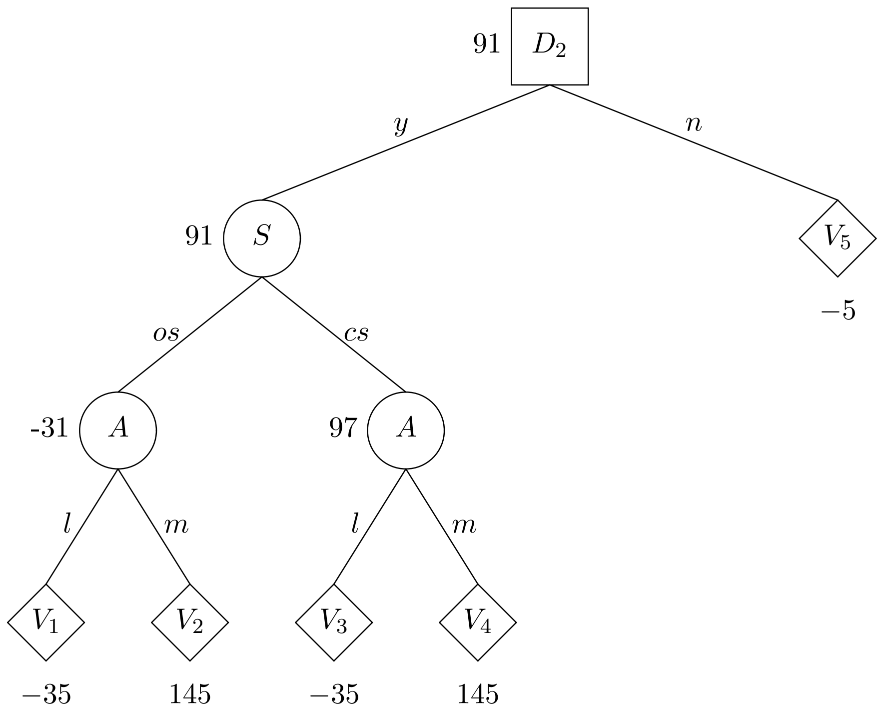


Figure 3.2.2: part 2 positive test result

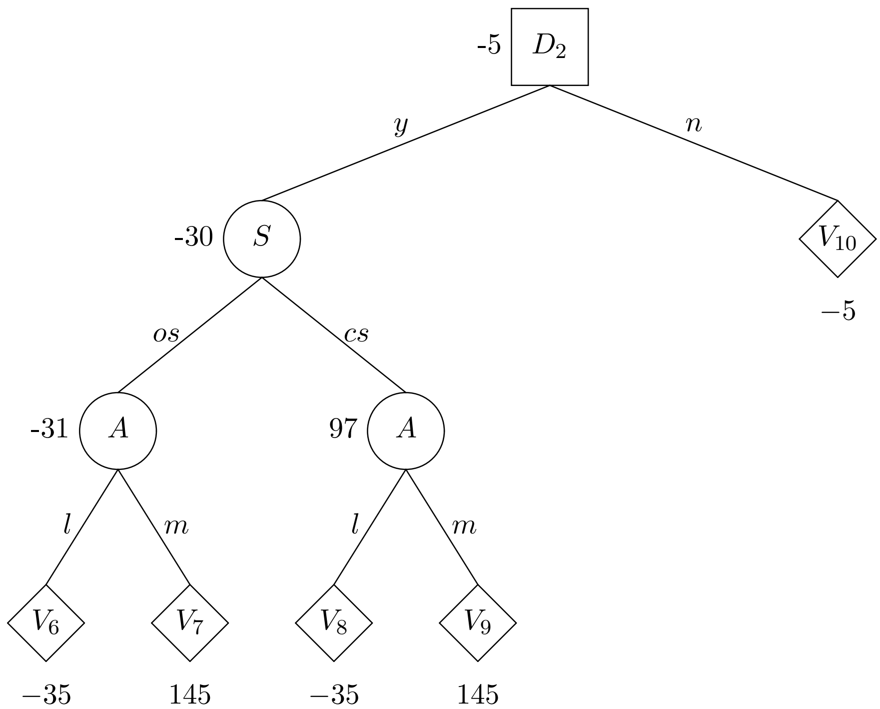


Figure 3.2.3: Decision tree part 3

3.3 From Influence Diagram to Bayesian Network

Another method to solve an influence diagram is its transformation to a Bayesian network. Gregory F. Cooper published the transformation steps in 2013 [2]. Three transformation steps are required. Decision nodes need to become chance nodes; value nodes need to become chance nodes, and the utility function needs to become a probability function.

Decision node to chance node
There are three steps to make a chance node out of a decision node. First, we remove the incoming edges, as these would yield to conditional probabilities. Second, we assign arbitrary probabilities between 0 and 1 to the states, former decision alternatives, in such a way that they sum up to 1 in total. The values can be arbitrary as the decision alternatives will be interpreted as evidence when the network is solved. Third, we adjust the shape of the node and the labelling, if necessary, to keep the model consistent.

Value node to chance
Similarly to the decision nodes we adapt the shape and the labelling of the nodes. The remaining modifications are part of the transformation from the linked utility function to a probability function.

Utility function to Probability function
The transformation of the utility function to a probability function is the mapping from each outcome and instantiated decision sequence d_s onto the interval [0, 1]. Hence, every outcome and decision sequcence receives a corresponding pseudo-probabilistic value $P(o, d_s)$.

$$P(o, d_S) = \frac{U(o, d_s) - \min_{o \in O} U(o, d_s)}{\max_{o \in O} U(o, d_s) - \min_{o \in O} U(o, d_s)}$$

The table below shows the porbabilistic values for the outcomes and corresponding decision sequences of the wildcatter oil problem and figure 3.3.1 represents the Bayesian network.

D_1	D_2	O	$U(o, d_s)$	$P(o, d_s)$
y	y	$o_1 - o_4$	$-35,000\$$	0
n	y	$o_1 - o_4$	$-30,000\$$	0,027
y	y	$o_5 - o_8$	$145,000\$$	0,973
n	y	$o_5 - o_8$	$150,000\$$	1
y	n	$o_1 - o_8$	$-5000\$$	0,162
n	n	$o_1 - o_8$	$0\$$	0,189

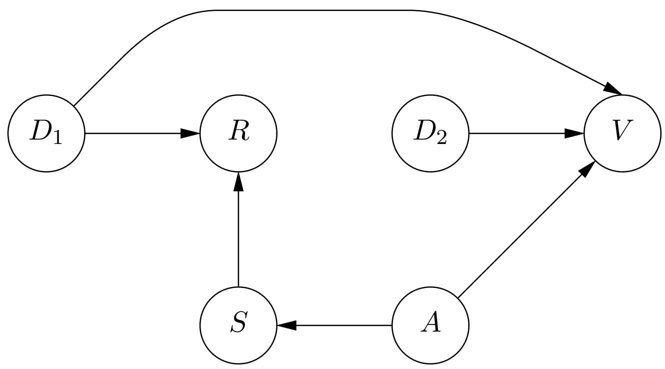


Figure 3.3.1: Bayesian network for the wildcatter oil problem

3.4 Solving an Influence Diagram

The algorithm to solve an influence diagram directly was invented by Ross D. Shachter in 1986 [9]. It is a stepwise instruction to "fold up" the network until the value node remains. Node by node is removed from the network following an appropriate method. The methods are edge reversals (ER), chance node removals (CNR), decision node removal (DNR) and barren node removal (BNR). In the following, v stands for the single value node in $V(G)$.

Edge reversal (ER)
An edge $(c_1, c_2) \in E(G)$ between two chance nodes with the probability distributions $P(A)$ and $P(B|A)$ respectively may be reversed, if there is no other path between the two nodes. The probability distribution of node c_1 changes from $P(A)$ to $P(A|B)$ and the probability distribution of node c_2 changes from $P(B|A)$ to $P(B)$.

$Precondition_{ER}(c_1, c_2)$
$= \exists c_1, c_2 \in C(G), \forall n \in N(G), n \in path(c_1, c_2) \rightarrow (n = c_1) \vee (n = c_2)$

$$P(A, B) = P(A)P(B|A)$$
$$P(B) = \sum_{a \in A} P(a, B)$$
$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

In addition, the set of parent nodes become the same. The set of parents of c_1, $par(c_1)$, is extended with the additional nodes of c_2 and vice versa. Additionally, the set of edges $E(G)$ is adjusted, yielding a new graph G'. The edge (c_1, c_2) is removed and the edge (c_2, c_1) is added.

$$E(G') \leftarrow (E(G) \setminus \{(c_1, c_2)\}) \cup \{(c_2, c_1)\}$$

Chance node removal (CNR)
A chance node X having the value node V as the only child node, may be removed by the following procedure. First, the nodes in $par(c)$ are added to $par(V)$. So, the value node inherits the parent nodes. Second, the old utility value is multiplied with the probability of $X = x$ given the parent nodes of X that are chance nodes. The set of parents that are chance nodes is notated as $par_c(X)$ and an instantiated set of $par_c(X)$ is written as $\widehat{par_c}(X)$. Furthermore, the set of outcomes O without X is given by O_{-X}.

$Precondition_{CNR}(x)$
$= \exists c \in C(G), \forall n \in N(G), c \in par(n) \rightarrow (c = V)$

$U_{new}(o_{-X}, \widehat{par_c}(X), d_s) = \sum_{X=x} u_{old}(o, d_s, X = x)P(X = x|\widehat{par_c}(X))$

Decision node removal (DNR)
This methods can be applied to a decision node D that is a parent node of v and a parent node of all other parent nodes of v. After the removal, the utilities are maximized over the decision alternatives. The sequence of decision nodes without D is notated as d_{s-1}.

$Precondition_{DNR}(d)$
$= \exists d \in D(G), \forall n \in N(G), n \in par(v) \rightarrow ((n = d) \vee (n \in chi(d))$

$U_{new}(o, d_{s-1}) = \max_{d \in D_i} U(o, d_{s-1}, D_i = d)$

Barren node removal (BNR)
During the application of the algorithm, it may happen that a chance or decision node becomes barren indicating that the node does not have a child node. These nodes can simply be removed from the network without consequences.

The algorithm
The algorithm can only be applied to influence diagrams with a single value node and requires that all informational edges were added beforehand. It stops when all parent nodes have been removed from the value node. Therefore, it first looks for a chance node fulfilling the preconditions to be removed. As a chance node is found the chance node removal method is called. As there is no chance node that fulfills the preconditions it searches for a decision node to apply the decision node removal method. Once a decision node is found the decision node removal method is called. Afterwards, the potential resulting barren nodes are removed from the network. If there is nor a chance nor a decision node that can be removed, the algorithm looks

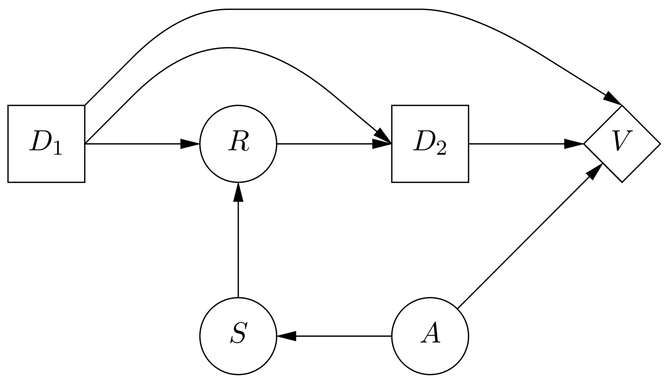
for a chance node that is a parent of the value node, but has other children nodes that are all chance nodes. Subsequently the outgoing edges are reversed until only the value node remains as a child node and the chance node can be removed in the next iteration.

Algorithm 2 Shachter's algorithm

while $par(v) \neq \emptyset$ **do**
  if $\exists c \in C(G), Precondition_{CNR}(c)$ **then**
    $CNR(c)$
  else if $\exists d \in D(G), Precondition_{DNR}(d)$ **then**
    $DNR(d)$
    BNR
  else
    while $\exists c_1, c_2 \in C(G), c_1 \in par(v) \wedge Precondition_{ER}(c_1, c_2)$ **do**
      $ER(c_1, c_2)$
    end while
  end if
end while

Subsequently, we will show the working of the algorithm with the aid of the wildcatter oil problem. Shachter also used this example in the cited paper to illustrate the shrinking of the diagram. However, the example offered in this paper is shortened and includes numerical values to reveal the development of the value node.

Starting point is the influence diagram shown in figure 3.1.3 with the added informational edge (D_1, D_2).



1. First, we must reverse the edge between chance node A and chance

node S.

$$ER(e(A,S))$$

$P(A)$	
l	0.7
m	0.3

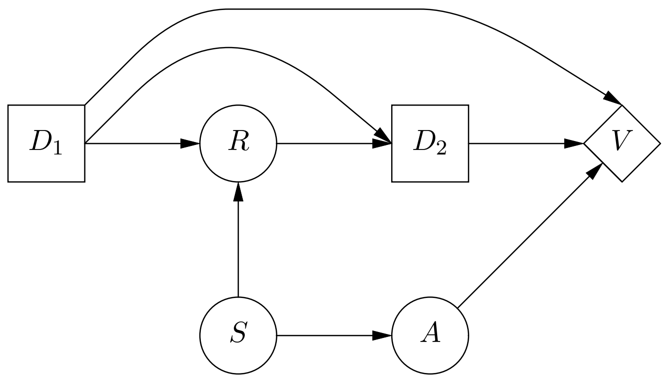
$\rightarrow$

$P(A\|S)$		
	os	cs
l	0.975	0.269
m	0.172	0.731

$P(S\|A)$		
	l	m
os	0.85	0.05
cs	0.15	0.95

$\rightarrow$

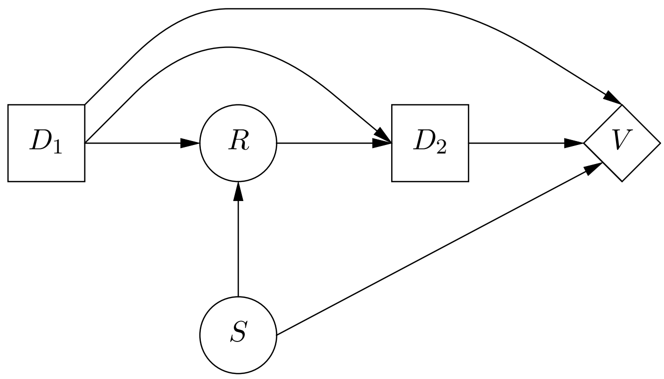
$P(S)$	
os	0.61
cs	0.39



2. Now, we can remove A from the diagram. As a consequence of the asymmetry, we have to treat three types of outcomes differently. Outcomes influenced by chance node S are multiplied with $P(A|S)$, outcomes solely depending on A are multiplied with $P(A)$ and outcomes that are not influenced maintain.

$$CNR(A)$$

O		d_s		
R	S	D_1	D_2	$U_{new}(O, d_s)$
p	os	y	y	-30.6
p	cs	y	y	96.5
p	–	y	n	-5
n	os	y	y	-30.6
n	cs	y	y	96.5
n	–	y	n	-5
–	–	n	y	24
–	–	n	n	0



3. Step four is the reversal of the edge from S to R.

$$ER((S, R))$$

$P(S)$	
os	0.61
cs	0.39

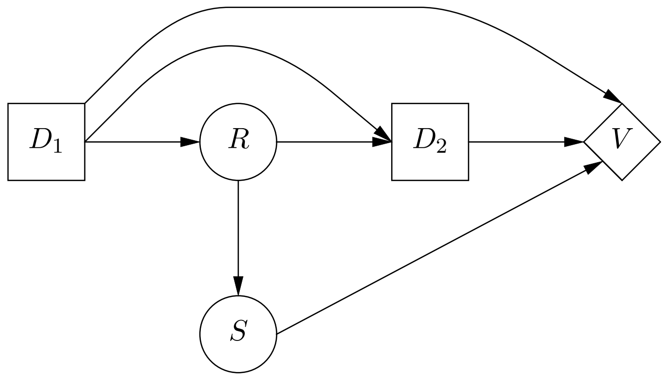
→

$P(S\|R)$		
	p	n
os	0,045	0,993
cs	0,955	0,007

$P(R\|S)$		
	os	cs
p	0.03	0.99
n	0.97	0.01

→

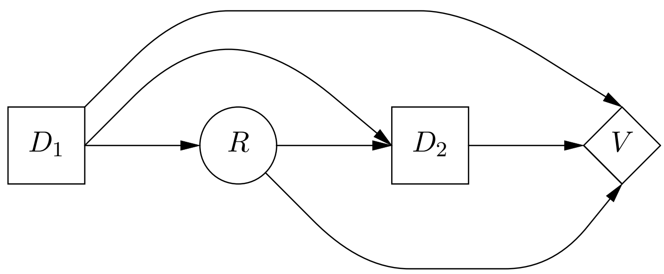
$P(R)$	
p	0,404
n	0,596



4. Step five is the removal of chance node S. Once again only outcomes that are influenced are adjusted.

$$CNR(S)$$

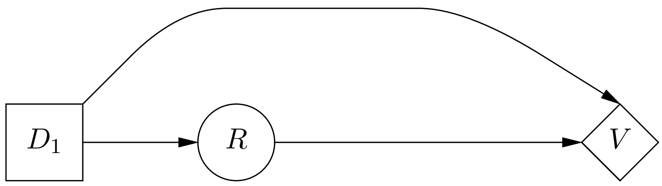
O	d_s		
R	D_1	D_2	$U_{new}(O, d_s)$
p	y	y	90.8
p	y	n	-5
n	y	y	-29.7
n	y	n	-5
$-$	n	y	24
$-$	n	n	0



5. Step six is the removal of the second decision node D_2. The expected utility is maximized for the decision alternatives of D_2 over the entries having the same instantiations for the remaining chance and decision nodes.

$$DNR(D_2)$$

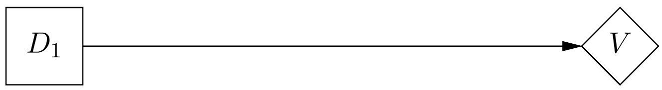
O	d_s	
R	D_1	$U_{new}(O, d_s)$
p	y	90.8
n	y	-5
$-$	n	24



6. Step seven is the removal of chance node R.

$$CNR(R)$$

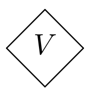
d_s	
D_1	$U_{new}(d_s)$
y	33.7
n	24



7. The last step is the removal of decision node D_1. The remaining entry in the table is the maximal expected utility, MEU.

$$DNR(D_1)$$

MEU
33.7



3.5 Decision Trees and Decision Networks

At first, the analyzation of decision problems was solely done with decision tree applications. The introduction of the influence diagram followed later. Influence diagrams were first meant to serve as a front-end tool to facilitate the design and understanding of decision problems. Nowadays, they are also used during the analytical process. In the following, we want to investigate the differences between the two decision system; decision trees and decision networks.

One difference is the representation of scenarios. A decision tree represents all scenarios as an individual path from the root node to the value node. In contrast to that, an influence diagram has to provide a single path that involves all nodes. These different representations have positive and negative impacts.

One impact refers to the size of the graphs. The size of a decision tree diagram grows at least exponentially with the number of nodes involved, whereas the growth of a decision network diagram remains linear. Hence, a decision tree representations become quite large for greater decision problems. The figures for the wildcatter oil problem illustrate this disadvantage, compare figure 3.1.3 with the figures 3.2.1, 3.2.2 and 3.2.3. An influence diagram presents the example with six nodes, and the decision tree representation demands 25 nodes.

However, the more detailed visualization in a decision tree diagram enables the facility to edit any scenario, which makes it possible to model and analyze asymmetric decision problems.

Decision networks, which use the more abstract representation are limited to symmetric decision problems. The one path structure defining the temporal order prohibits an alternation of the action sequence and the bundled representation of scenarios demands their equal treatment. Consequently, either order, functional or structural asymmetry can be expressed.

Another principal difference is that decision networks apply probabilistic inference. The network structure obeys information about the probabilistic dependencies of the chance nodes. If the system receives new knowledge about a probabilistic distribution, the affected nodes can automatically be updated. In a decision tree, such an observation would lead to a workover of the entire diagram. The decision tree structure does not embody probabilistic dependencies.

The last difference that we want to discuss is the solution process. Decision trees are solved with the average out and fold back algorithm. Decision networks on the other side originally did not have any algorithm to solve it. The different methods to solve an influence diagram were introduced later by Shachter and Cooper. Still, there is no method that has been added to the outline of decision networks.

Chapter 4

More Flexible Decision Networks

In this section, we want to extend the decision network to make it more flexible towards asymmetries. Therefore, we will investigate each type of asymmetry in particular and afterward we will introduce a way to enable the user of an advanced decision networks to analyze them. In the extended network, we will implement central elements of the algorithm by Cooper, using the transformation to a Bayesian network.

4.1 Structural Asymmetry

Structural asymmetry circumscribes the situation in which the existence of a node depends on its past. The most well-known case is the performance of a test. If the user decides to make the test, a result in the form of a chance node is available; otherwise, the chance node does not affect the utility value. A corresponding decision tree representation is shown in figure 4.1.1.

In the subtree starting with the chance node, the number of outcomes is multiplied by the number of states. In the path without the chance node, the number of outcomes is unchanged. In this instance, the structural asymmetry component consists of one node, but it may also consist of a sequence of chance and decision nodes. If this is the case, the number of outcomes is multiplied by each number of states or actions. Thus, the structural asymmetry has two effects. The number of outcomes of the resulting subtrees is different, and the probabilistic inference for the affected nodes is absent in the outstanding scenarios.

4.2 Functional Asymmetry

Functional asymmetry is given if an entered decision or observation restricts the states of succeeding chance or decision nodes. So, an action or a state

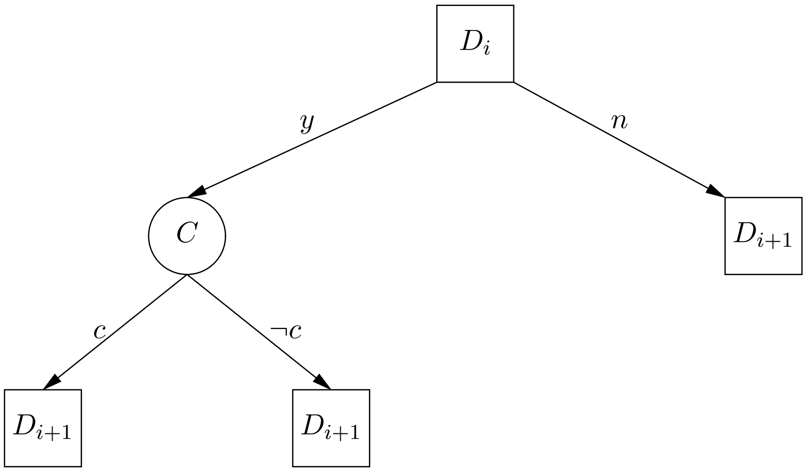


Figure 4.1.1: structural asymmetry

of a chance node may become impossible, because of the past. Figure 4.2.1 shows the case in which a state of a chance node inhibits the action of the subsequent decision node. The state $\neg c$ inhibts the second decision alternative of D_i. The outgoing edge and the linked subtree is removed in the decision tree representation.

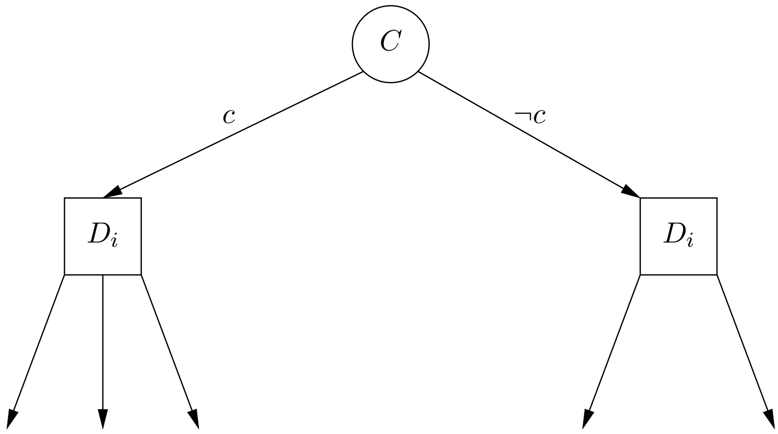


Figure 4.2.1: functional asymmetry

4.3 Order Asymmetry

Order asymmetry emerges in the situation, in which the decision maker wants to have the option to change the sequence of the decision while the system is running. In a decision tree, we can model this type of asymmetry with the aid of an additional decision node. Figure 4.3.1 illustrates an order asymmetry in a decision tree. The node D_i is the additional node and works like a railway switch. The decision maker indicates the order of the two decisions right before the first decision is made. The original order D_i, D_{i+1} and D_{i+2} is reversed in one of the resulting subtrees.

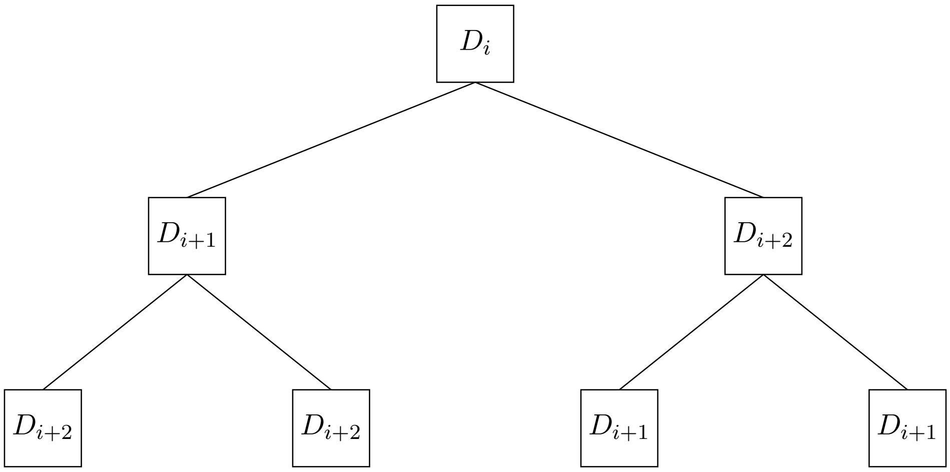


Figure 4.3.1: order asymmetry

4.4 The Advanced Decision Network

Besides the examined asymmetric components, the significant drawback of decision networks is the inconvenient way that has to be taken to solve it. As the Shachter algorithm already can be applied to any regular decision network, the advanced decision network goes towards a simplification of its transformation to a Bayesian network.

The design abbreviates the required steps, defined by Cooper [2], by making a clear distinction between edges performing probabilistic inference and informational edges solely denoting temporal precedence. Consequently, the diagram consists of a probabilistic network illustrated with solid lines and a temporal order defined by informational edges shown with dashed lines.

The decision nodes have a special role in the system. They are part of the probabilistic network without holding a probability table. Cooper solved this problem by linking them to arbitrary probabilities to bring them in line with the specifications of a Bayesian network. In the advanced decision network the demanded instantiations are illustrated with thick lines.

The residual layout guidelines are retained unchanged. Figure 4.4.1 shows the new network for the wildcatter oil problem.

So far, we established the visual distinction between the probabilistic and

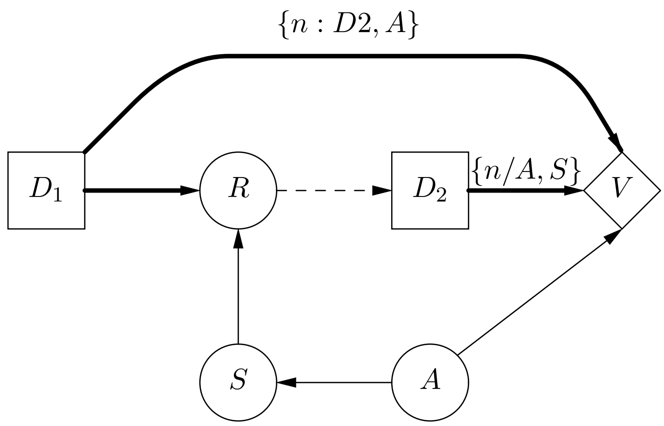


Figure 4.4.1: advanced decision network for the wildcatter oil problem

informational parts. Now, we want to introduce the features that are needed to work with structural and functional asymmetries.

The wildcatter oil problem already involves two asymmetric components. The first asymmetry is the popular test scenario with the decision to make the test D_1 and its results R.
The second asymmetry emerges at decision D_2. If the decision is made not to drill, the utility values become independent of the probability tables hold by the chance nodes A and S, see figure 3.2.2 and figure 3.2.3.

There are two unwanted effects because of asymmetries. First, the probabilistic network contains entries for redundant data, e.g. $P(R = p|D_1 = n)$. Second, it is not visible if a probabilistic edge is valid for a given scenario. This can only be deduced based on additional knowledge about the decision problem.

To overcome these drawbacks, two types of rules are introduced that are used to define multiple subnetworks to be able to deduce the valid networks for any structural or functional asymmetry involved.

4.4.1 Bounding and Excluding Rules

Asymmetries are defined by bounding and excluding rules in the advanced decision networks. These rules are attached to the incoming edges of the value node and have their starting point at the node where the asymmetry emerges. Both rules are notated in curly brackets and begin with a set of states X belonging to the outgoing node. The second set Y contains the set of nodes and states that are still valid for the states in X or that are not valid for the states in X. This depends on the type of the rule which either can be bounding or excluding.

Bounding rules express that the states in X may only be combined with the set of nodes in Y denoted as:

$$\{X : Y\}.$$

Excluding rules express that the states in X may not be combined with the set of nodes and sates in Y denoted as:

$$\{X/Y\}.$$

4.4.2 Informational Lines

With the aid of the new rules, structural and functional asymmetric components can be defined and illustrated as they emerge at direct predecessors of the value node. Direct predecessors are all decision nodes, because they are part of the input of the utility function, but a chance node is not necessarily a direct predecessor. To illustrate structural or functional asymmetry emerging at a chance node that is not a direct predecessor of the value node, we will use informational lines. These lines are dashed like informational edges but do not indicate temporal precedence. They solely connect a chance node to a value node to illustrate the existence of a bounding or excluding rule.

4.4.3 Temporal Switch

Order asymmetric components demand a relaxation of the rule that a single temporal oder has to be defined for the network. To illustrate that a decision D_{i-1} can also be taken before a decision D_i, we will introduce a temporal switch.

The switch is a small filled rectangle in the diagram that receives multiple incoming informational edges and has exactly one outgoing edge.

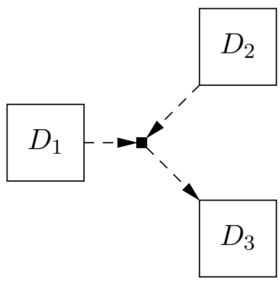


Figure 4.4.2: temporal switch

It allows an arbitrary ordering of the parent nodes. The temporal switch shown in 4.4.2 allows the ordering $D_1 \prec D_2 \prec D_3$ as well as the ordering $D_2 \prec D_1 \prec D_3$. Important is to notice that a switch of the decision nodes also switches the set of chance nodes observed directly before these decisions.

4.5 Solving the Advanced Decision Network

In this section, a method is presented to solve the advanced decision network for an asymmetric decision problem using multiple Bayesian networks. So, different networks are used to illustrate asymmetries for the different scenarios. These networks are deduced based on the introduced bounding and excluding rules. To receive the maximal expected utility the networks are parallel solved and merged until one network remains.

4.5.1 Deducing the Bayesian Networks

As mentioned earlier, the system defines a probabilistic network. If we have a symmetric decision problem, this network would be sufficient. In asymmetric decision problems however, there are scenarios in which parts of this network are not valid. Hence, different Bayesian networks are defined for these scenarios.

To deduce these networks, the bounding and excluding rules are used. Starting point is the whole network that establishes a set of nodes N. The rules are worked out in line with the temporal order. The earlier the asymmetry emerges in the temporal order; the earlier the rule is dealt with. To work out a bounding rule $\{X : Y\}$ or an excluding rule $\{X : Y\}$, the set of states defined in X are separated. Therefore, the previous set of nodes is split into two sets, one including X and one without X. The remaining nodes are part of both sets.

For the set containing X, the remaining nodes are then adjusted based on the type of rule. If it is a bounding rule, nodes that are not in Y are removed and if it is an excluding rule, the nodes in Y are removed. The next rules have to be worked out for all resulting sets.

The wildcatter oil problem includes two rules, one bounding rule $\{D_1 = n : D_2, A\}$ and one excluding rule $\{D_2 = n/A, S\}$. So, the bounding rule is attached to D_1 and the excluding rule is attached to D_2. In the temporal order, D_1 comes before D_2 and therefore we have to work out the bounding rule first.

The sets that we receive are: $\{D1 = y, R, D_2, A, S\}$ and $\{D1 = n, D_2, A\}$. Based on this result we work out the excluding rule and receive four sets of nodes and states: $\{D_1 = y, R, D_2 = y, A, S\}$, $\{D_1 = y, R, D_2 = n\}$, $\{D_1 = n, D_2 = y, A\}$ and $\{D1 = n, D_2 = n\}$.

Hence, we have four different scenarios in which a different set of nodes and states is valid. The corresponding Bayesian networks for the solution process are shown in figure 4.5.1.

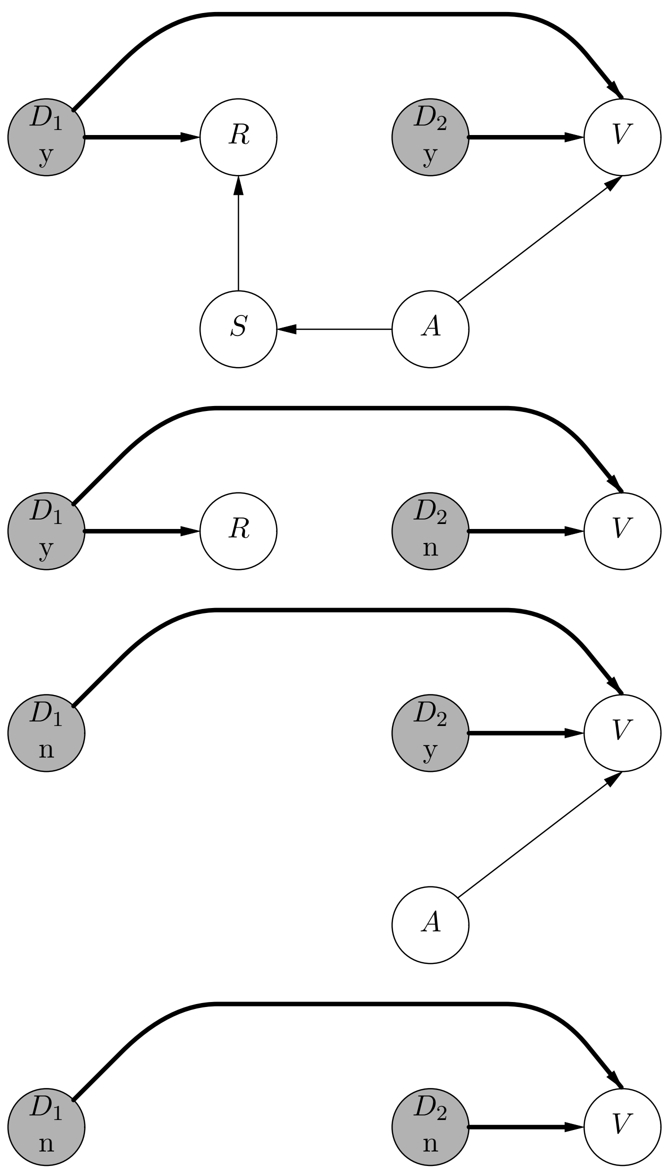


Figure 4.5.1: Multiple Bayesian networks for the wildcatter oil problem

4.5.2 Calculate the Maximal Expected Utility

To calculate the MEU, the Bayesian networks are first solved for the last decision in parallel. Therefore, we instantiate the network for all combinations of the last decision and all earlier nodes according to the temporal order. Then, the value node is queried for each instantiation. For same instantiations of earlier nodes, only the maximum value of the decision alternatives is retained.

The results become part of a new value node. This value node replaces the solved part of the network. The solved part consists of the last decision node and all later nodes according to the temporal order.

Furthermore, edges are added to the network. From every chance node, an edge to the value node is added.

After this is accomplished for all Bayesian networks, we might receive same network structures. These networks are merged by holding only the utility values that maximize the utility for an instantiation in both networks.

This is process is repeated till we receive one Bayesian network, which solely consists of the value.

So, for the wildcatter oil problem we first have to solve the four Bayesian networks for the last decision D_2. The results and the succeeding Bayesian network of the first two networks in fig 4.5.1 are shown below.

Network 1:

$$P(V = v|D_1 = y, D_2 = y, R = p) = 0.678$$

$$P(V = v|D_1 = y, D_2 = y, R = n) = 0.028$$

Network 2:

$$P(V = v|D_1 = y, D_2 = y, R = p) = 0.162$$

$$P(V = v|D_1 = y, D_2 = y, R = n) = 0.162$$

Both networks transform into the same network structure, as they are

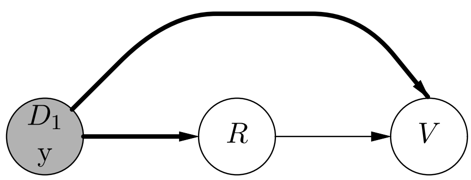


Figure 4.5.2: Resulting Bayesian network 1

solved for D_2, see figure 4.5.2. Hence, the networks are merged. The new value node holds the utilities $P(V = v|D_1 = y, R = p) = 0.678$ and $P(V = v|D_1 = y, R = n) = 0.162$.

The third and fourth network also end up in the same network structure, see figure 4.5.3.

Network 3:

$$P(v|D_1 = n, D_2 = y) = 0.319$$

Network 4:

$$P(v|D_1 = n, D_2 = n) = 0.189$$

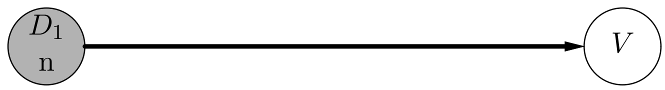


Figure 4.5.3: Resulting Bayesian network 2

The value node of this network only holds the utility $P(v|D_1 = n) = 0.319$.

In the second iteration both resulting networks transform to the value node. The result of the first network is $P(V = v|D_1 = y) = 0.37$ and $P(v|D_1 = n) = 0.319$ for the second network. Hence, the maximal expected utility is 0.37 and the maximizing decision alternative for D_1 is y.

Chapter 5

Conclusions

In this thesis, we reviewed the different theories combined in decision systems, the different features of decision trees and decision networks and various solving algorithms. Later, we introduced an advanced decision network, capable of presenting asymmetries and showed how the wildcatter oil problem is is solved with an advanced decision network using multiple Bayesian networks.

The representation of functional and structural asymmetries is achieved with rules that have to be set in the modeling process. Their creation claims that the decision maker is aware of the asymmetries and their impact on the scenarios in the underlying decision problem. If the rules are attached properly, the advanced decision network can be transformed into a decision tree representation. This can be done without additional knowledge about the decision problem. Hence, an expert that is familiar with these rules can deduce every scenario and for every scenario the nodes having impacto on the linked utilities can be determined.

The proposed solution method for the wildcatter oil problem illustrates that an asymmetric decision problem can be solved using multiple Bayesian networks. This approach however is very expensive. The networks have to be created, instantiated, solved using probabilistic inference and merged. So when we compare the number of steps needed, the solution process propsed by Shachter or the decision tree representation achieves a beter result [9].

Nevertheless, as we apply the algorithm of Shachter, we can not be sure about the presence of an edge for a given scenario. Shachter states that an edge may or may not exist. Thus, this needs to be checked while the decision problem is solved. This could be realized by holding unique data for each scenario or by linking the data to the scenarios. Yet, which data is stored or linked to an scenario can not be read out of the provided diagram. So, the representaion of an asymmetric decision problem by an regular influence diagram is incomplete. The advanced decision network facilitates a full representation of asymmetric decision problems using the temporal order and bounding and excluding rules. Now, it is possible to define the described may or may not existence of an edge based on the diagram.

Furthermore, there is the algorithm invented by Cooper. This algorithm transforms the influence diagram to a Bayesian network. In a Bayesian network, there is no may or may not existence of edges. All edge that are part of the diagram exists for each scenario. Consequently, all scenarios are treated the same way. So, the expected utilities of scenarios affected by asymmetries are distorted. Consequently, a transformed decision network can not be used to solve asymmetric decision problems. Therefore, it was shown in section 4.5.2 how an advanced decision network could be solved for an asymmetric decision problem by transforming it into multiple bayesian networks.

Bibliography

[1] David Barber. *Bayesian reasoning and machine learning*. Cambridge University Press, 2012.

[2] Gregory F Cooper. A method for using belief networks as influence diagrams. *arXiv preprint arXiv:1304.2346*, 2013.

[3] Bruce D'Ambrosio. Inference in bayesian networks. *AI magazine*, 20(2):21, 1999.

[4] Thomas Dyhre Nielsen and Finn Verner Jensen. *Bayesian networks and decision graphs*. Springer Science & Business Media, 2009.

[5] Mike Oaksford and Nick Chater. *Bayesian rationality: The probabilistic approach to human reasoning*. Oxford University Press, 2007.

[6] John Quiggin. A theory of anticipated utility. *Journal of Economic Behavior & Organization*, 3(4):323–343, 1982.

[7] Howard Raiffa. Decision analysis: introductory lectures on choices under uncertainty. 1968.

[8] Stuart Russell and Peter Norvig. Artificial intelligence: a modern approach. 1995.

[9] Ross D Shachter. Evaluating influence diagrams. *Operations research*, 34(6):871–882, 1986.

[10] Ross D Shachter. Probabilistic inference and influence diagrams. *Operations Research*, 36(4):589–604, 1988.