Optimization of information structures in influence diagrams

Topias Terho

School of Science

Thesis submitted for examination for the degree of Master of Science in Technology.

Espoo 23.5.2022

Supervisor and advisor

Prof. Ahti Salo

Aalto University
School of Science

Aalto University
School of Science

Aalto University, P.O. BOX 11000, 00076 AALTO
www.aalto.fi
Abstract of the master's thesis

Author Topias Terho	
Title Optimization of information structures in influence diagrams	
Degree programme Mathemathics and Operations Research	
Major Applied Mathematics	**Code of major** SCI3053
Supervisor and advisor Prof. Ahti Salo	
Date 23.5.2022 **Number of pages** 51+1	**Language** English

Abstract

Since mid 1980s, influence diagrams have been used widely in decision analysis. Traditionally, influence diagrams have a predetermined structure and the no-forgetting property, which means that earlier decisions can be recalled when making later decisions. The main focus in the literature on influence diagrams has been on determining the optimal decision strategy for an influence diagram with a given structure. However, the information structure of an influence diagram, i.e. what information should be acquired to support decisions, has attracted far less attention.

In this thesis, we examine what information should be available to the decision maker. We present optimization models for the information structure and the decision strategy of an influence diagram. The first optimization model enforces constraints on path probabilities, the second on local decisions and the third is based on the formulation of an extended state space. All models are tested with a variety of modified oil wildcatter problems and N-M-monitoring problem. The constraints on local decisions are clearly the fastest with all problem sizes. The constraints on path probabilities and the constraints on extended state space are also applicable to similar problems but they are clearly slower than the constraints on local decisions. The approaches worked for relatively large instances but the limitations of the methods become apparent when the size of a problem is grown.

Keywords influence diagrams, information structure, optimization models, desicion analysis

Aalto University
School of Science

Aalto-yliopisto, PL 11000, 00076 AALTO
www.aalto.fi
Diplomityön tiivistelmä

Tekijä Topias Terho	
Työn nimi Vaikutuskaavion informaatiorakenteen optimointi	
Koulutusohjelma Mathematics and Operations Research	
Pääaine Applied Mathematics	**Pääaineen koodi** SCI3053
Työn valvoja ja ohjaaja Prof. Ahti Salo	
Päivämäärä 23.5.2022 **Sivumäärä** 51+1	**Kieli** Englanti

Tiivistelmä

Vaikutuskaavioita on jo 80-luvulta asti laajasti käytetty päätösanalyysissä. Ne tarjoavat hyvän työkalun monivaiheisten epävarmuuksia sisältävien päätösongelmien jäsentämiseen ja ratkaisemiseen. Aiemmassa tutkimuksessa on erityisesti tarkasteltu päätösstrategian optimointia. Tässä diplomityössä keskitytään sen sijaan vaikutuskaavion informaatiorakenteen optimointiin. Pyrkimyksenä on optimaalisen päätösstrategian lisäksi optimoida tieto, jota tarjotaan päätöksen tueksi.

Vaikutuskaavion informaatiorakenteen optimointiin esitetään tässä työssä kolme optimointimallia. Yksi optimointimalleista asettaa rajoituksia polkutodennäköisyyksille päätöksen tueksi valitun tiedon mukaan. Toinen optimointimalli asettaa rajoituksia paikallisille päätöksille valitun tiedon mukaan. Kolmas optimointimalli hyödyntää laajennettua tilajoukkoa solmuille, jotka eivät välttämättä ole käytössä päätöksiä tehdessä.

Kaikkia malleja on testattu erinäisillä esimerkkiongelmilla. Malleilla ratkaistiin perinteinen öljynetsintäongelma useilla raporttivaihtoehdoilla, sekä niin kutsuttu N-M-monitorointiongelma. Testeissä havaittiin yhden mallin toimivan muita paremmin. Rajoitukset paikallisille päätöksille ratkaisivat lähes kaikki ongelmat nopeiten. Rajoitukset polkutodennäköisyyksille ja rajoitukset laajennetulle tilajoukolle toimivat vaihtelevasti. Kaikkia malleja pystyttiin soveltamaan useita päätöksiä ja raportteja sisältäviä ongelmiin.

Avainsanat Vaikutuskaavio, informaatiorakenne, optimointimallit, päätösanalyysi

Contents

1 Introduction

Influence diagrams were developed in mid 70s as a tool to visually present a decision problem [1; 2]. Since then, influence diagrams have been used widely in decision analysis [3]. Their combination of visual interpretability and sound mathematical theory makes them useful for presenting decision problems for non-mathematically oriented decision makers as well as for modellers and mathematicians as vehicle for capturing the key aspects of the problem. Many real life problems have been modelled with the help of influence diagrams from environmental management [4] to fighter pilot's maneuvering decisions [5]

Influence diagrams represent problems with a known structure. An important assumption has been the *no-forgetting* assumption of influence diagrams, which states that at the time of a decision, a decision maker remembers the earlier decisions as well as the information on which the decisions were based. This assumption does not hold in limited memory influence diagrams, shorthanded as LIMIDs [6]. Still, an important assumption with LIMIDs is that it is known what information can be utilized when making decisions. This makes it difficult to model situations where the availability of information depends on uncertain events or earlier decisions. These types of endogenous and exogenous uncertainties have not attracted much research.

The goal of this thesis is to build and analyse optimization models for the information structure of an influence diagram. The question that our optimization models addresses is what information should the decision maker have at various decisions to achieve maximum expected utility.

The change to the structure of the decision problem that we consider is the addition of edges to decision nodes. These edges represent the information structure of a decision problem. The addition of an edge to a decision node indicates that more information is available to the decision maker. In real-life it may not be known what information will be available when making a decision. The availability of information to support a certain decision can thus be viewed as a decision in its own right with related costs. In this situation, it is of interest to determine the optimal information such that the expected utility can be maximized. For example, a decision maker that is responsible of building a bridge can order tests that indicate if the bridge sits on solid ground or if reinforcements are needed to stabilize the bridge or not. If the decision maker has to make a decision to reinforce the ground, he can either order the test and base the decision on the result of the test; or he can avert the costs of the test and reinforce the ground anyway. Thus the information on which the decision maker bases the decision is conditional on the decision to run the test.

The availability of information to support a decision may also depend on random events. Consider for example the nuclear plant safety system where different sensors indicate the need for certain safety measures [7]. Safety measures may have to be taken instantly when sensors go off for various purposes. The information that is available to the decision maker to do a given action is given by a single sensor. However, it may be the case that some actions contain overlapping fixes. If multiple sensors go off in a short period of time, the decision maker may know what actions have been already been taken and if some expensive actions can be consequently averted. Thus the set of available information on which the decision maker bases the decisions can differ in various situations. This thesis concentrates on the deterministic modification of the information sets of decision nodes.

This thesis is constructed as follows: Chapter 2 discusses the necessary preliminary information, Chapter 3 presents and analyses three different methods for solving the optimal information structure, Chapter 4 presents some computational experiments for the analysed methods, Chapter 5 discusses potential applications and extensions for the developed frameworks and Chapter 6 concludes.

2 Background

2.1 Influence diagrams

An influence diagram is an acyclic graph $G = (N, A)$ containing nodes and directed edges [1]. A node represents an event that takes place in the decision problem. For example in the bridge example that was alluded to in the introduction, the report indicating if the ground is solid or not can be represented by a node. Nodes are divided into chance nodes C, decision nodes D and utility nodes V. Each node $i \in N = C \cup D \cup V$ in the diagram has a finite collection of possible states S_i, with the realized state being denoted as s_i. For example the possible states of the mentioned report could be $S_i = \{solid,\ not\ solid\}$, with the realized state being the result of the report. Only one state from the set of possible states can be the realized state. States can also be represented by numbers, such that $S_i = \{1, ..., n\}$, where n denotes the number of possible states. Chance nodes represent nodes such that the node $c \in C$ corresponds to a random variable X_c. Decision nodes $d \in D$ represent nodes in which a decision maker selects an alternative, which is one of states of the node. Value nodes represent the consequences of the decision problem and are typically evaluated with a utility function.

The nodes are connected via directed edges. The edges represent different relationships depending on the type of the node that sits at the end of the edge. For chance nodes, the edges represent conditional probability distribution. For decision nodes, the edges represent the information which is available for the decision maker when selecting the state from the given alternatives. For value nodes the edges represent arguments to the utility function. The set of nodes from which there is an edge to node $j \in D \cup C$ is called the *information set* of j (Denoted as $I(j)$) [8]. An information state $s_{I(j)}$ is then the collection of states of all nodes in the information set. Thus, the conditional probability distribution at a chance node c is given by $\mathbb{P}(X_c = s_c | S_{I(c)} = s_{I(c)})$. At decision nodes $j \in D$ the decision maker must select a decision alternative based on $s_{I(j)}$.

An influence diagram represents a decision problem in which a decision maker selects a decision alternative at each decision node based on the information state of the node. These decisions affect the probability distributions of chance nodes and the overall value of the system. A local decision strategy $Z_j : S_{I(j)} \mapsto S_j$ is a function from the information state to the domain of the possible states of the decision variable. A binary function $Z_j : S_{I(j)} \mapsto S_j \Leftrightarrow z(s_j | s_{I(j)}) \mapsto \{0, 1\}$ can also be used when using linear inequalities to find the optimal decision strategy [8]. The local decision strategy is injective, since there must be a decision alternative for each possible combination of states $s_{I(j)} \in S_{I(j)}$. A global decision strategy Z is a collection of local decision strategies for all decision nodes.

As an illustrative example, the influence diagram of the classic oil wildcatter problem is given in Figure 1. The diagram is a simplification of the classic problem where an oil wildcatter must decide wether to drill oil from a potential drilling site. [9] In the influence diagram decision nodes are presented as rectangles, chance nodes are presented as circles and value nodes are presented as diamonds. The edges are shown as arrows pointing from a node to another in the influence diagram. Thus, in Figure 1, the information set of the chance node *Test result* is {*Test*, *Amount of oil*}, which means that the probability distribution of the test result is conditional to the realization of *Amount of oil* (i.e. the state of the well) and the decision to run the test. Similarly, the information set of the node *Drill* is the node *Test result*. This means that the decision maker knows the realization of the node *Test result*, when making the drilling decision. The information state of the node *profit* is {*Drill*, *Amount of oil*}, which means that the utility of the decision problem depends on the drilling decision and on the amount of oil in the drilling site. The node $Test\ result$ has states $S_{\textit{Test result}} = \{Na, No, Yes\}$ and the node *Drill* has states $S_{\textit{Drill}} = \{No, Yes\}$. An example local decision strategy for the drilling decision is given in Table 1. We select a decision alternative $s_{\textit{Drill}} = Yes$ if the state in node *Test result* is Na or Yes and we select a decision alternative $s_{\textit{Drill}} = No$ if the state *Test result* is No.

Table 1: A local decision strategy $Z_{\textit{Drill}}$

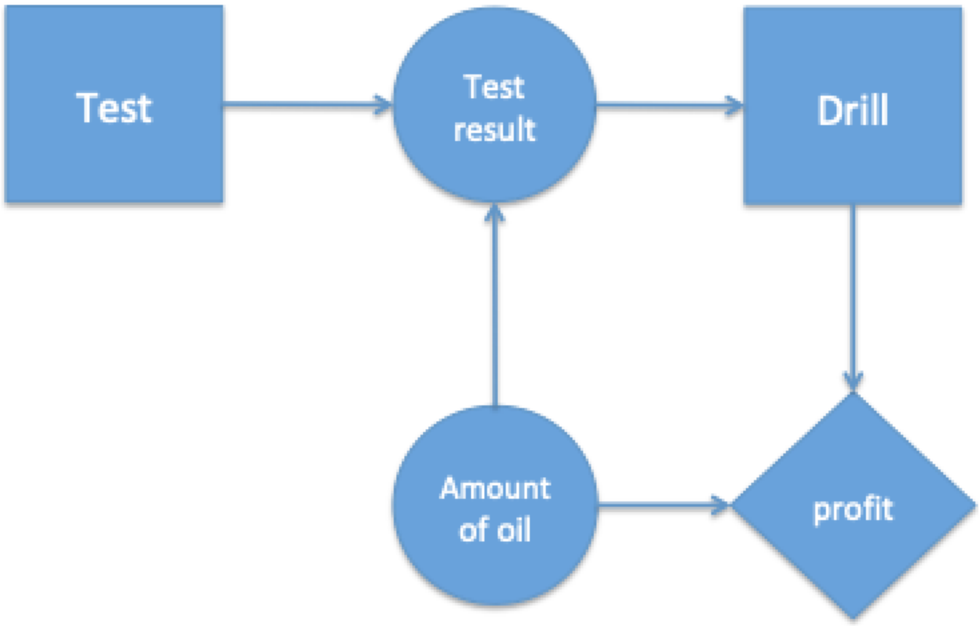
$s_{\textit{Test result}}$	$s_{\textit{Drill}}$
Na	Yes
Yes	Yes
No	No

In Decision Programming [8], a path is a sequence of states $s = (s_1, s_2, ..s_n)$ of both decision and chance nodes. Path probabilities $\pi(s)$ depend on the decision strategy so that

$$\pi(s) := \mathbb{P}(s|Z) = \prod_{j \in C} \mathbb{P}(X_j = s_j | X_{I(j)} = s_{I(j)}) \prod_{i \in D} z(s_i | s_{I(i)}). \tag{1}$$

The upper bound for the probability of path s is given by the probability of the

Figure 1: Influence diagram of the oil wildcatter decision problem



chance nodes along the path s, i.e.,

$$p(s) := \prod_{j \in C} \mathbb{P}(X_j = s_j | X_{I(j)} = s_{I(j)}) \tag{2}$$

The lower bound of a path is zero, which is obtained if the path is incompatible with the decision strategy, i.e. a path contains a decision alternative that would not be selected with the decision strategy, or the probability of a random event stated in the path is zero.

The expected utility of the decision consequences can be expressed with the path probabilities given a decision strategy Z

$$EU(Z) = \sum_{s \in S} \pi(s) U(s), \tag{3}$$

where $U(s)$ denotes the utility of the consequences for the path s.

In many influence diagrams, no-forgetting is typically assumed. This means that earlier decisions are known when making later decisions. However, there are situations where this assumption does not hold and thus a concept of <u>l</u>imited <u>m</u>emory <u>i</u>nfluence <u>d</u>iagram has been introduced [10]. LIMID is a formulation of an influence diagram where the no-forgetting assumption does not hold. An example of this type of situation is a system where multiple decision makers are responsible of decisions.

2.2 Optimal decision strategy

An optimal decision strategy is a decision alternative for each decision node such that the expected utility of decision consequences is maximized.

Definition 2.1. *Optimal decision strategy Z^* is such that $EU(Z^*) \geq EU(Z)$, for all feasible strategies Z [11]*

The optimal decision strategy is a decision alternative s_j for all $j \in D$ and for all information sets of the decision nodes, which maximizes the expected utility of decision consequences.

There are many ways to find the optimal decision strategy when the no-forgetting and acyclicity requirements are fulfilled. For example, one can evaluate an influence diagram by performing a series of node removals by removing barren nodes, replacing chance nodes by a conditional expectation and replacing decision nodes by maximizing the utility function. These actions may require arc reversals but in the end there is only the utility node, which contains the optimal decision strategy [12; 13]. Another well-known method is to transform the influence diagram to a decision tree and to solve the optimum with dynamic programming. This gives the possibility to introduce multiple value nodes to the influence diagram by combining them to a so-called super value node [14]. Influence diagrams can also be solved with techniques of Bayesian nets with the help of a framing function [15] and with multistage Monte Carlo methods [16], among others. A good overview of different solving methods is collected by Shachter and Bhattacharjya in [17].

LIMIDs are more challengeing for finding a global optimal decision strategy. It has been shown to be NP-hard to find the optimal decision strategy [18; 19]. However, there are viable methods for this as well. There are many good methods for finding the optimal decision strategy in decision problems which can be solved by applying a series of local utility maximizations. Lauritzen and Nilsson present the single policy updating algorithm, which is an iterative local utility maximization algorithm. With this algorithm, the influence diagram is first transformed to a junction tree and then the local policies are updated iteratively with a message passing algorithm.[6]. Parmentier et. al. use a mixed-integer-linear-programming (MILP) approach in a variant of a strong junction tree for the expected utility maximization [20]. Mauá and Cozman develop approximation algorithms with the help of k-neighbour search

[21]. Yuan et. al. apply a branch-and-bound search algorithm [22]. However, these approaches do not work for problems with global constraints, for example budget constraints that make some local decision strategies incompatible with some local decision strategies at other decision nodes are difficult to account for with only local computations.

These kind of problems can be solved to optimality with a Decision Programming framework consisting of a set of equality and inequality constraints [8].

$$
\begin{aligned}
\max_{z \in Z} \quad & \sum_{s \in S} \pi(s) U(s) && && (4)\\
\text{s.t.} \quad & \sum_{s_i \in S_i} z(s_i | s_{I(i)}) = 1, && \forall i \in D, \forall s_{I(i)} \in S_{I(i)} && (5)\\
& 0 \leq \pi(s) \leq p(s), && \forall s \in S && (6)\\
& \pi(s) \leq z(s_i | s_{I(i)}), && \forall s \in S, \forall i \in D && (7)\\
& z(s_i | s_{I(i)}) \in \{0, 1\}, && \forall i \in D, \forall s \in S, \forall s_{I(i)} \in S_{I(i)} && (8)
\end{aligned}
$$

2.3 Information structure

The information structure of an influence diagram $G = (N, A)$ is the set of edges A that connect the nodes. In this thesis we ask the question what is the optimal structure of an influence diagram, i.e., what information the decision maker should obtain to support decisions when the availability and accuracy of this information may be associated with costs or other decisions. Thus, the information set $I(j)$ of a decision node $j \in D$ is consequently also subject to the decision makers discretion subject to some constraints and impacts on the objective function. The edges connect nodes in the way that was presented in section 2.1. The changes that can be considered to the information structure are additions of edges and removals of edges. The addition of an edge (depending on the type of relationship that it describes) either expands the information set of a decision maker or affects the probability distribution of a random event. The removal of an edge has the reverse effect.

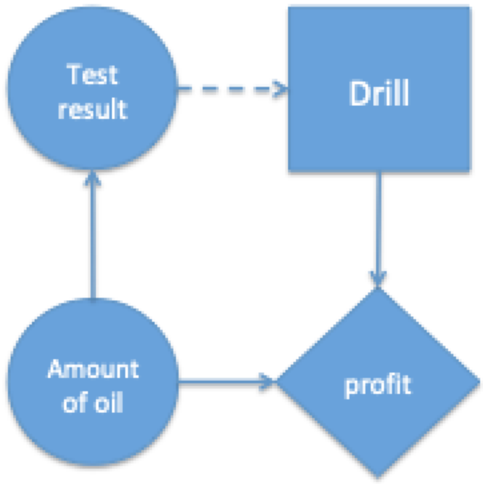
In this thesis, we only consider the addition of edges to decision nodes to establish a clear focus. The removal of edges is analogous and could be applicable with a similar approach. The revised information structure must be acyclic, meaning that arbitrary edges between two nodes of an influence diagram cannot be added. In this thesis the acyclicity requirement is accounted for by predetermining the set of available edges in a way that any combination from the set can be added to the information structure without creating a cycle into the influence diagram. It is also possible to consider information links between all nodes in the influence diagram but then the acyclicity requirement must be accounted for in a different way. One example could be to first filter out information structures which introduce a cycle to the graph. There are well established methods for detecting cycles within a directed graph. [23; 24; 25]

Previously, research on information structures has concentrated on the type 2 endogenous uncertainties, which corresponds problems where the information structure is affected by decisions [26]. Jonsbråten et al. study the so-called decision dependent uncertainties in a stochastic programming framework [27]. Xhang et al. present a robust optimization method that can deal with type 2 endogenous uncertainties and use it to solve a plant redesign problem [28]. Herrala et al. present an extension for the Decision Programming framework that is general enough to deal with endogenous uncertainties of types 1 and 2 [29], where type 1 endogenous uncertainties refer to uncertainties in decision dependent probabilities [26]. A good literature review of the endogenous uncertainties is presented in [30].

We present a complementary solution approach to Herrala et al. [29]. Their solution is to introduce decision nodes $D_{i,j}$ that correspond decisions to add edges between nodes $i \in C \cup D$. and $j \in D$. Thus, you can calculate an optimal decision strategy where you optimize the edges and the decisions. However, introducing new decision nodes increases the number of paths significantly. In this thesis, we present optimization formulations that do not add any decision nodes to the influence diagram. As Salo et al. note, the solution time grows exponentially when growing the number of paths [8]. Thus it is meaningful to come up with a solution that does not increase the number of paths. Our optimization models add constraints to situations where unavailable information is used, instead of introducing new nodes to the influence diagram and consequently increasing the number of paths.

Another solution method is to augment the states in the nodes in the information set $I(j)$. For example, in the oil wildcatter problem the realization of the test result can be set to Na (not known), when the decision maker decides not to run the test. The augmentation increases the number of possible paths, thus requiring more computations. However, if the number of added constraints stays quite low, the augmentation of states is also an option worth considering. The problem in the oil wildcatter problem is that the possible states of node $Test\ result$ are augmented and a decision node is introduced for the test result. Ideally, we would either augment the states in node $Test\ result$ or add a decision wether to run a test or not.

Figure 2: Influence diagram of the oil wildcatter problem with a conditional edge



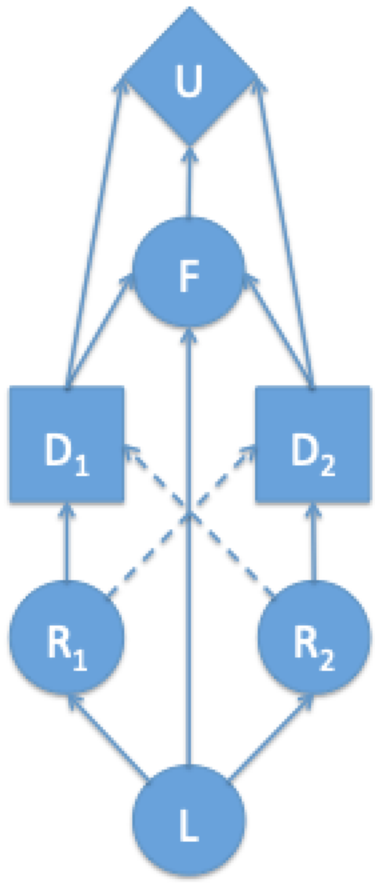
In this thesis, we examine conditional edges that are represented by dashed edges in the influence diagram. Conditional edge describes a decision to include a node to the information set of a decision node $j \in D$. Thus, the oil wildcatter problem introduced above is reduced to three nodes plus the utility node. The influence diagram is presented in Figure 2, where the dashed arrow replaces the decision wether or not to run the test.

3 Methodological development

We present three different optimizing models for the information structure and the decision strategy. These are the constraints on path probabilities, the constraints on local decisions and the constraint on extended state space. Here, we present concepts that support the optimization models in addition to the optimization models.

3.1 2-monitoring

Figure 3: Influence diagram of 2-monitoring problem



N-monitoring problem [8], with $N = 2$ demonstrates how the developed optimization models work. Assume a random load L on a structure. In the N-monitoring problem we get N independent reports R_N on the load on the structure. Then based on those reports we have the ability to do N fortification actions. However, each fortification action is decided based on a single report. The failure F of the structure is conditional on the load and on the fortification actions that have been taken. The utility U of the system depends on the failure event and the cost of the fortification actions.

Instead of settling for deciding the fortification decisions based on only a single report, we consider making the information given by the other reports available for other decisions as well. We ask what additional information the decision maker should acquire when making a fortification decision. In the influence diagram the possible changes to the information structure are indicated with dashed edges. The influence diagram of the problem is in Figure 3.

For example decision D_2 has the report R_2 at its disposal and the additional information that is considered is R_1. Table 2 presents the possible values that each node can attain. There are 6 nodes with 2 states each. Thus, the number of different paths is $2^6 = 64$ and the number of possible decision strategies is $4 * 4 = 16$. Since we are using the problem as an example for calculating the models, we present probability tables to calculate the optimal information structure and decision strategies. Tables 3 through 4 present the probabilities of s_F, s_{R_1} and s_{R_2} given their information states. The probabilities of s_L are uniformly distributed, meaning $P(s_L = big) = 0.5$ and $P(s_L = small) = 0.5$. The inputs of the utility function are the states of nodes F, D_1 and D_2. The utility of the system is calculated as $U(s_F, s_{D_1}, s_{D_2}) = t(s_F) + t(s_{D_1}) + t(s_{D_2})$, where $t(.)$ is the value given in Table 5 for the given state. The utilities are normalized so that the minimum utility of the system is 0.

Table 2: Possible values of nodes

Node	values
L	*big*, *small*
R_n	*big*, *small*
D_n	*yes*, *no*
F	*yes*, *no*

Table 3: Probability table for failure given load and each decision $\mathbb{P}(F|L, D_1, D_2)$

$s_{I(F)}$	$P(s_F = no)$	$P(s_F = yes)$
$s_L = small$, $s_{D_1} = no$, $s_{D_2} = no$	0.9	0.1
$s_L = small$, $s_{D_1} = no$, $s_{D_2} = yes$	0.96	0.04
$s_L = small$, $s_{D_1} = yes$, $s_{D_2} = no$	0.92	0.08
$s_L = small$, $s_{D_1} = yes$, $s_{D_2} = yes$	0.90	0.01
$s_L = big$, $s_{D_1} = no$, $s_{D_2} = no$	0.5	0.5
$s_L = big$, $s_{D_1} = no$, $s_{D_2} = yes$	0.8	0.2
$s_L = big$, $s_{D_1} = yes$, $s_{D_2} = no$	0.7	0.3
$s_L = big$, $s_{D_1} = yes$, $s_{D_2} = yes$	0.9	0.1

Table 4: Probability table for load given decisions and load $\mathbb{P}(s_{R_n}|s_L)$

s_L	$small$	big
s_{R_1}	$small = 0.7$ $big = 0.3$	$small = 0.3$ $big = 0.7$
s_{R_2}	$small = 0.9$ $big = 0.1$	$small = 0.1$ $big = 0.9$

Table 5: Utility function values

s_F, s_{D_1}, s_{D_2}	yes	no
$t(s_F)$	0	200
$t(s_{D_1})$	0	10
$t(s_{D_2})$	0	20

3.2 Constraints on path probabilities

Let $K(j)$ be a set of candidates from which we can extend the current information set $I(j)$ to create an augmented information set $\bar{I}(j) = I(j) \cup \{k\}, k \in K(j)$ for the decision node $j \in D$. In this thesis $K(j)$ is a predefined set of nodes from which one or more can be added to the influence diagram in combination of other nodes in $K(j)$ without creating a cycle. Specifically we can define a binary mapping $x(k, j)$.

$$x(k,j) = \begin{cases} 1, & \text{if } s_k \text{ is known at } j \in D \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

If $x(k, j) = 1, \forall k \in K(j)$, the decision maker can utilize the information given by

$s_{\bar{I}(j)}$, when selecting the decision alternative $s_j \in S_j$. If $x(k,j) = 0, \forall k \in K(j)$, the decision maker does not know s_k and the decision alternative has to be made based only on $s_{I(j)}$.

The availability of information leads to a decision which is equally good or better than the decision without the additional information. This inequality is characterized by Theorem 3.1.

Theorem 3.1. *Let $j \in D$ be a decision node with initial information set I(j) and a local decision strategy Z_j. If $\bar{I}(j) = I(j) \cup k, k \in K(j)$ and the addition of k does not involve costs, then there exists a local decision strategy Z'_j such that $EU(Z') \geq EU(Z)$*

Proof. The expected utility for the decision strategy Z is:

$EU(Z') = \sum_{s \in S} \pi(s) U(s)$

$= \sum_{s \in S} \prod_{l \in C} \mathbb{P}(X_l = s_l | X_{I(l)} = s_{I(l)}) z'(s_j | s_{I(j)}, s_k) \prod_{i \in D/j} z(s_i | s_{I(i)}) U(s)$

$\geq \sum_{s \in S} \prod_{l \in C} \mathbb{P}(X_l = s_l | X_{I(l)} = s_{I(l)}) z(s_j | s_{I(j)}) \prod_{i \in D/j} z(s_i | s_{I(i)}) U(s)$

$= EU(Z),$

The inequality follows from the fact that we can select a different decision alternative $z(s_j|s_{I(j)}, s_k) \neq z'(s_j|s_{I(j)}, s'_k), s'_k \neq s_k$. However, we also have the option to select a similar decision alternative $z(s_j|s_{I(j)}, s_k) = z'(s_j|s_{I(j)}, s'_k), s'_k \neq s_k$ as without the additional information s_k. Selecting a similar decision alternative does not change the expected utility of the system. Thus there must exist a decision strategy Z' such that $EU(Z') \geq EU(Z)$

□

Let $I \subsetneq N, I \neq \varnothing$. For a given path segment s_I, defined by states $s_i, i \in I$ we can define an extension operator $E(s_I)$,

$$E(s_I) = \{s' \in S | s'_i = s_i, \forall i \in I\}. \tag{10}$$

A complement of an extension is derived with the help of the extension operator.

$$\overline{E(s_I)} = \{s' \in S | \exists i \in I, s'_i \neq s_i\} = \cup_{i \in I} E(S_i \backslash s_i) = S \backslash E(s_I). \tag{11}$$

Apart from path segments, we can also consider extensions of sets $S'_i \subseteq S_i, \forall i \in$

$1, ..., n$.

$$E(S'_1 \times S'_2 \times ... \times S'_n) = \{s' \in S | s_i \in S'_i, \text{ For any } i \in \{1, ..., n\} \text{ such that } S'_i \neq \varnothing\} \tag{12}$$

Now, consider a decision node $j \in D$ with an information set $I(j)$ and a local decision strategy $Z_j : S_{I(j)} \mapsto S_j$. Then a *partial decision strategy* $Z_{j,\tilde{I}(j)}$ is a mapping from $S_{\tilde{I}(j)}$ to S_j, where $\tilde{I}(j) \subsetneq I(j)$ s.t.

$$s_j \in Z_{j,\tilde{I}(j)}(s_{\tilde{I}(j)}) \Longleftrightarrow \exists s_{I(j)} \in S_{I(j)} \text{ such that } Z_j(s_{\tilde{I}(j)}, s_{I(j)\setminus\tilde{I}(j)}) = s_j. \tag{13}$$

With information structures, we are interested in the situation where the decision maker may not have access to all possible information. Specifically, the partial decision strategy $Z_{j,\tilde{I}(j)}(s_{\tilde{I}(j)})$ describes a situation where the decision maker only has access to the information defined in $s_{\tilde{I}(j)}$, whereas the information given by $s_{I(j)\setminus\tilde{I}(j)}$ remains unknown.

Next, consider the expansion of the information set through a new constraint. Consider a decision node $j \in D$ with an information set $I(j)$ and a set $K(j)$. Consider the most simple case, where $|K(j)| = 1$. Thus, the information with which the decision in node j has to be made can either be $I(j)$ or $\bar{I}(j) = I(j) \cup k$, where $k \in K(j)$. The local decision strategy Z_j at node j attains values for each realization of its information state.

We thus have two possibilities for a decision strategy. A decision strategy which maps the larger information state to the decision alternatives $Z_j(s_{\bar{I}(j)})$ and a partial decision strategy $Z_{j,I(j)}$ which maps a smaller information state to the decision alternatives. If the arc between node $k \in K(j)$ and j is not added, meaning that $x(k,j) = 0$, then the partial decision strategy $Z_{j,I(j)}(s_{I(j)})$ is a function in the sense that it has to map all information states to decision alternatives such that $Z_j(s_{\bar{I}(j)}) = Z_{j,I(j)}(s_{I(j)})$. This means that the local decision strategy at j must give the same decision alternative s_j, for each $s_{I(j)} \in S_{I(j)}$. The information state $s_{I(j)}$ differs from $s_{\bar{I}(j)}$ by not having the state s_k. Thus, no matter what value of s_k we have, the decision alternative $z(s_j|s_{I(j)}, s_k)$ must stay the same. The only possibility for choosing a different decision alternative is to have a different $s_{I(j)}$. If k is is known when making a decision at j (i.e. $x(k,j) = 1$), then we allow the decision strategy to attain different values with different s_k and a similar information state $s_{I(j)}$. This can be enforced with a constraint.

$$\begin{aligned} \pi(s') \leq 1 + x(k,j) - z(s_j|s_{I(j)}, s_k), \\ \forall s_j, s_{I(j)}, s_k, \forall s' \in E(s_{I(j)}) \cap \overline{E(s_j)} \cap \overline{E(s_k)} \end{aligned} \tag{14}$$

The set $s' \in E(s_{I(j)}) \cap \overline{E(s_j)} \cap \overline{E(s_k)}$ iterates over the set of paths that differs from path $s \in S$ in nodes j and k but is the same in nodes $I(j)$. If $x(k,j) = 0$ and we have that $z(s_j|s_{I(j)}, s_k) = 1$ then the constraint enforces that the probability of a path s' with $s'_k \neq s_k$ and $s'_j \neq s_j$ and $s'_I(j) = s_I(j)$, must be 0. In other words, if the information given by s_k is not available, the decision must stay the same when the information given by s_k changes and information given by $s_{I(j)}$ stays unchanged. Conversely if $x(k,j) = 1$, then we allow the probability of the path s' to be greater than 0.

If the cardinality of the set $K(j)$ is greater than one, we can enforce the requirements by iterating over all possible states of all $k \in K(j)$

$$\pi(s') \leq 1 + x(k,j) - z(s_j|s_{\bar{I}(j)\backslash k}, s_k), \forall s_j, s_{\bar{I}(j)}, s_k, \forall k \in K(j), \forall s' \in E(s_{\bar{I}(j)\backslash k}) \cap \overline{E(s_j)} \cap \overline{E(s_k)}. \tag{15}$$

Thus far the analysis of the information structures has relied on a decision to share information from specified nodes. More importantly, the requirement is that the information set of a decision node is decided beforehand. However, it may be of interest to represent problems, in which earlier events influence the realization of the information sets of decision nodes. Following the notations of [29], we can define a distinguishability set $D_{k,j} \in C \cup D$ and and a distinguishability condition $F : S_{D_{k,j}} \mapsto 0, 1$, such that $F(S_{D_{k,j}}) = 1 \iff x(k,j) = 1$. The difference is that now the realization is conditional to the states of earlier nodes. We can modify the notation and introduce a mapping $x_{k,j}(s)$ that tells if for path s the edge (k,j) exists. We denote the set of paths for which this is true as $S_{D_{k,j}} \subseteq S$ Thus the constraints on path probabilities can be rewritten as

$$\pi(s') \leq 1 + x_{k,j}(s') - z(s_j|s_{I(j)}, s_{k_j}), \forall s' \in E(s_{I(j)}) \cap \overline{E(s_j)} \cap \overline{E(s_{k_j})} \cap \overline{S_{D_{k,j}}}. \tag{16}$$

The constraint states that for all situations, where the information set without the additional information s_{k_j} stays the same, the additional information s_{k_j} changes and similarly the distinguishability condition changes, we cannot change the decision if we have no access to the additional information.

3.3 Cost of adding edges

The expansion of the information set can increase the expected utility of the system. To select the edges that grow the expected utility the most, we optimize the information structure of the problem. If we do not limit the number of edges that can be added to the influence diagram, the information structure with the maximum amount of information would be selected. The only restriction that could limit the number of edges is the acyclicity of the influence diagram. Thus the problem would reduce to finding an optimal subset of potential edges such that a cycle would not be created. However, the acyclicity requirement is accounted for by predetermining the set of potential edges.

A penalty for each added edge is instead used to represent the cost of acquiring information. Consider for example two independent scout patrols, which have an option of contacting the other patrol. The contact increases the probability of getting caught and thus the decision wether or not to contact affects the expected utility of the system. This is not considered in the initial formulation of the problem.

The penalty can be expressed in many ways. In some cases, one can quantify the cost of adding information and subtract it from the utility function. This could be the case, for example, if the added wear and tear would be considered for the communication equipment for the scout patrols. For example, if a very fragile and expensive equipment would need to be replaced after 100 uses, then the penalty term to the utility function would be $\frac{1}{100}p$, where p denotes the price of the equipment. In some cases the penalty term cannot be added to the utility function, because some decisions can affect the probability distributions of some chance events, which then in turn affect the utility of the system.

3.4 Optimization models with constraints on path probabilities

Adding the constraints on path probabilities to the optimization model introduced in [8], we get the following model:

$$
\begin{aligned}
\max_{x\in X}\max_{z\in Z} \quad & \sum_{s\in S} \pi(s)U(s) && && (17)\\
\text{s.t.} \quad & \sum_{s_j\in S_j} z(s_j|s_{\bar{I}(j)}) = 1, && \forall j\in D, \forall s_{\bar{I}(j)}\in S_{\bar{I}(j)} && (18)\\
& 0\le \pi(s)\le p(s), && \forall s\in S && (19)\\
& \pi(s)\le z(s_j|s_{\bar{I}(j)}), && \forall s\in S, \forall j\in D && (20)\\
& z(s_j|s_{\bar{I}(j)})\in\{0,1\}, && \forall j\in D, \forall s\in S, \forall s_{\bar{I}(i)}\in S_{\bar{I}(i)} && (21)\\
& x(k,j)\in\{0,1\}, && \forall j\in D, k\in K(j) && (22)\\
& \pi(s')\le 1+x(k,j)-z(s_j|s_{I(j)},s_k), && \forall s_j, s_{I(j)}, s_k, \forall s'\in E(s_{I(j)})\cap\overline{E(s_j)}\cap\overline{E(s_k)} && (23)
\end{aligned}
$$

We optimize over all decision strategies $z\in Z$ and over all possible information structures $x\in X$. Here X denotes all combinations of binary variables $x(k,j)$, for all $j\in D$. Now let $j=D_1$ and $K(j)=R_2$ in the 2-monitoring problem. Thus $X := x(R_i,D_j)\in\{0,1\}$ such that $i,j\in\{1,2\}, i\neq j$. Consider a decision strategy in Tables 6 and 7, in which the decision in D_1 is informed by the report R_2. The decision at D_1 is different when s_{R_1} is *big* and s_{R_2} is either *small* or *big*. Thus the decision strategy requires that the edge between nodes R_2 and D_1 exists.

To demonstrate how the constraints on path probabilities work, assume that the information structure is specified by $x(R_2,D_1)=0$ and $x(R_1,D_2)=1$, meaning that there is no edge between nodes R_2 and D_1 and there is an edge between nodes R_1 and D_2. Consider paths in Table 8. Both paths s^1 and s^2 are compatible with the given decision strategy as they fulfill the constraint (20). In constraint (23) we have that $z(s_j|s_{I(j)},s_k)=1, s\in s^1,s^2$. We also assumed that $x(R_2,D_1)=0$. Thus the right side of the constraint (23) becomes $1+0-1=0$ at $j=D_1$. Now if we examine path s^2, we can see that $s^2_{D_1}\neq s^1_{D_1}$, $s^2_{I(D_1)}=s^1_{I(D_1)}$ and $s^2_k\neq s^1_k$. Thus we see that $s^2\in E(s^1_{I(D_1)})\cap\overline{E(s^1_{D_1})}\cap\overline{E(s^1_k)}$. Thus it should hold according to constraint (23) that $\pi(s^2)\le 0 \Longrightarrow 0.032\le 0$, which is not the case. Thus decision strategy in Tables 6 and 7 is not compatible with the information structure $x(R_2,D_1)=0$. However, if we change a part of the information structure to $x(R_2,D_1)=1$ when $x(R_1,D_2)$ stays the same, then the constraint holds and the decision strategy is compatible with the information structure.

Table 6: A local strategy Z_{D_1}

s_{R_1}	s_{R_2}	s_{D_1}
big	*big*	*yes*
small	*big*	*no*
small	*small*	*no*
big	*small*	*no*

Table 7: A local decision strategy Z_{D_2}

s_{R_2}	s_{D_2}
big	*yes*
small	*no*

Table 8: Illustrative paths

	s_L	s_{R_1}	s_{R_2}	s_{D_1}	s_{D_2}	s_F	$\pi(s)$
s^1	*big*	*big*	*small*	*no*	*yes*	*no*	0.028
s^2	*big*	*big*	*big*	*yes*	*yes*	*no*	0.032

Next we present a model for the optimal information structure of a given decision node. We introduce a penalty matrix R such that the cost of adding an edge (k, j) is given by $R(k, j)$. The cost is subtracted from the expected utility of the system if the edge is added to the information structure. The optimal information structure of a given decision node j is the solution of the following optimization problem:

$$\max_{x\in X}\max_{z\in Z} \quad \sum_{s\in S}\pi(s)U(s) - \sum_{k\in K(j)} x(k,j)R(k,j) \tag{24}$$

$$\text{s.t.} \quad \sum_{s_j\in S_j} z(s_j|s_{\bar{I}(j)}) = 1, \qquad \forall j\in D, \forall s_{\bar{I}(j)}\in S_{\bar{I}(j)} \tag{25}$$

$$0 \le \pi(s) \le p(s), \qquad \forall s\in S \tag{26}$$

$$\pi(s) \le z(s_j|s_{\bar{I}(j)}), \qquad \forall s\in S, \forall j\in D \tag{27}$$

$$z(s_j|s_{\bar{I}(j)}) \in \{0,1\}, \qquad \forall j\in D, \forall s\in S, \forall s_{\bar{I}(j)}\in S_{\bar{I}(j)} \tag{28}$$

$$x(k,j) \in \{0,1\}, \qquad \forall j\in D, k\in K(j) \tag{29}$$

$$\pi(s') \le 1 + x(k,j) - z(s_j|s_{\bar{I}(j)\setminus k}, s_k), \qquad \forall s_j, s_{\bar{I}(j)\setminus k}, s_k, \forall k\in K(j), \tag{30}$$

$$\forall s'\in E(s_{I(j)})\cap\overline{E(s_j)}\cap\overline{E(s_k)} \tag{31}$$

Finally, we present a model which optimizes the information structure of an influence diagram by considering multiple decision nodes in conjunction.

$$\max_{x\in X}\max_{z\in Z} \quad \sum_{s\in S}\pi(s)U(s)-\sum_{j\in D}\sum_{k\in K(j)}x(k,j)R(k,j) \tag{32}$$

$$\text{s.t.} \quad \sum_{s_j\in S_j} z(s_j|s_{\bar{I}(j)})=1, \qquad \forall j\in D, \forall s_{\bar{I}(j)}\in S_{\bar{I}(j)} \tag{33}$$

$$0\le \pi(s)\le p(s), \qquad \forall s\in S \tag{34}$$

$$\pi(s)\le z(s_j|s_{\bar{I}(j)}), \qquad \forall s\in S, \forall j\in D \tag{35}$$

$$z(s_j|s_{\bar{I}(j)})\in\{0,1\}, \qquad \forall j\in D, \forall s\in S, \forall s_{\bar{I}(j)}\in S_{\bar{I}(j)} \tag{36}$$

$$x(k,j)\in\{0,1\}, \qquad \forall j\in D, k\in K(j) \tag{37}$$

$$\pi(s')\le 1+x(k,j)-z(s_j|s_{\bar{I}(j)\setminus k},s_k), \qquad \forall j\in D, \forall k\in K(j), \forall s_j, s_{\bar{I}(j)\setminus k}, s_k, \tag{38}$$

$$\forall s'\in E(s_{I(j)})\cap\overline{E(s_j)}\cap\overline{E(s_k)} \tag{39}$$

3.5 Constraints on local decisions

Another way to represent the problem are constraints on local decisions. Consider decision nodes $j\in D$ and their augmented information sets $\bar{I}(j)$. A *decision set* is a set of decision nodes and their augmented information sets $\overline{D}=\cup_{j\in D}\{j\}\bigcup\cup_{j\in D}\bar{I}(j)$. A *decision path* is then a sequence of states $s_{\overline{D}}\in S_{\overline{D}}$. An *active decision path* is s.t. the decision strategy maps the augmented information states to the corresponding decisions in the decision path $s_{\overline{D}}$. We can define a binary mapping $y(s_{\overline{D}})\in\{0,1\}$ as

$$\begin{aligned} y(s_{\overline{D}}) &\le \frac{1}{|D|}\sum_{j\in D} z(s_j|s_{\bar{I}(j)}) \\ y(s_{\overline{D}}) &\ge \sum_{j\in D} z(s_j|s_{\bar{I}(j)})-|D|+1, \end{aligned} \tag{40}$$

where the states s_j and $s_{\bar{I}(j)}$ correspond with the decision path $s_{\overline{D}}$. The decision path $s_{\overline{D}}$ is active if $y(s_{\overline{D}})=1$.

The active decision path is thus a path s such that the augmented information set of node j coincides with the path. Formally this is a set $\{s\in S|s_{\overline{D}}=s'_{\overline{D}}\}$. However, the actual information set might be smaller than the augmented information set. We can define a binary variable $\xi(k)$ through

$$\begin{aligned}\xi(k) &\leq \frac{1}{\Xi(k)} \sum_{\{j\in D | k\in \overline{I}(j)\}} x(k,j) \\ \xi(k) &\geq \Xi(k) - \sum_{\{j\in D | k\in \overline{I}(j)\}} x(k,j) + 1,\end{aligned} \tag{41}$$

where $\Xi(k) = |\{j \in D | k \in \bar{I}(j)\}|$ The mapping $\xi(k)$ is thus 1 if and only if the node k is known in all decision nodes that it is a candidate for. Consequently, $\xi(k)$ gets the value zero if k is not known at every decision that it is a candidate for. Formally, $\xi(k) = 0$ if there exists $j \in D$ s.t. $k \in \bar{I}(j)$ and $x(k,j) = 0$.

Of interest are nodes $k \in K(j), j \in D$ such that $\xi(k) = 0$. These are the nodes that are in the augmented information set of some decision node $j \in D$ and therefore their states are a part of the active decision path but they are not a part of the information that the decision maker can utilize. Nodes $k \in K(j), j \in D$ such that $\xi(k) = 1$ are then included in all possible information sets and in all cases the decision maker knows the state of the node.

Let $\underline{K}$ be a set of nodes k s.t. $\xi(k) = 0$. For an active decision path $s_{\overline{D}}$ we require that if the information set that the decision maker can utilize is smaller than the augmented information set, then the decision strategy cannot depend on the nodes that are not available. This corresponds to the situation where the decision strategy $z(s_j | s_{\overline{I}(j)\setminus s_k}, s_k)$ stays the same for all states $s_k \in S_k$. This condition can be enforced through

$$\frac{1}{|D|} \sum_{j\in D} z(s_j | s'_{\overline{I}(j)\setminus I(j)}, s_{I(j)}) \geq y(s_{\overline{D}}), \forall s_{\overline{D}} \in S_{\overline{D}}, \forall s'_{\underline{K}} \in S_{\underline{K}}. \tag{42}$$

This approach is attractive in the sense that instead of full paths it suffices to inspect active decision paths.

However, a better approach is to limit the decisions made based on the selected information set and the active decision path for each decision node $j \in D$. Above a decision path was introduced to cover all decision nodes and their maximal information sets. However, the decisions in different decision nodes are made based on their maximal information set and thus it suffices to analyse the corresponding *local decision sets* $\overline{D_j} = \{j\} \cup \bar{I}(j)$ and *local decision paths*

$$s_{\overline{D_j}} \in S_{\overline{D_j}} = S_j \cup S_{\bar{I}(j)}. \tag{43}$$

Thus for each $s_k, \forall k \in K(j)$, we have to make sure that the decision s_j stays the same for different s_k if $x(k,j) = 0$. We can enforce this with the following constraint

$$|z(s_j|s'_k, s_{\bar{I}(j)\setminus k}) - z(s_j|s_k, s_{\bar{I}(j)\setminus k})| \leq x(k,j), \forall j \in D, \forall k \in K(j), \forall s \in S_{\overline{D_j}}, \forall s'_k \neq s_k. \tag{44}$$

An advantage of this approach is that the number of constraints stays fairly low even though the number of paths grows. The number of constraints is thus proportional to the cardinality of the augmented information state.

3.6 Optimization models with constraints on local decisions

Adding the constraints on local decisions to the Decision Programming framework, we get the following optimization model:

$$\max_{x \in X} \max_{z \in Z} \quad \sum_{s \in S} \pi(s) U(s) - \sum_{j \in D} \sum_{k \in K(j)} x(k,j) R(k,j) \tag{45}$$

$$\text{s.t.} \quad \sum_{s_j \in S_j} z(s_j | s_{\bar{I}(j)}) = 1, \qquad \forall j \in D, \forall s_{\bar{I}(j)} \in S_{\bar{I}(j)} \tag{46}$$

$$0 \leq \pi(s) \leq p(s), \qquad \forall s \in S \tag{47}$$

$$\pi(s) \leq z(s_i | s_{\bar{I}(j)}), \qquad \forall s \in S, \forall i \in D \tag{48}$$

$$z(s_j | s_{\bar{I}(j)}) \in \{0,1\}, \qquad \forall j \in D, \forall s \in S, \forall s_{\bar{I}(j)} \in S_{\bar{I}(i)} \tag{49}$$

$$x(k,j) \in \{0,1\}, \qquad \forall j \in D, k \in K(j) \tag{50}$$

$$|z(s_j|s'_k, s_{\bar{I}(j\setminus k)}) - z(s_j|s_k, s_{\bar{I}(j)\setminus k})| \leq x(k,j) \quad \forall j \in D, \forall k \in K(j), \forall s \in S_{\overline{D_j}}, \forall s'_k \neq s_k \tag{51}$$

The optimization is carried over the binary variables $X := \{x(k,j) | j \in D, k \in K(j)\}$ and over all possible decision strategies. In the 2-monitoring example, the local decision paths for the decision nodes are $\overline{D}_{D_1} = \{D_1, R_1, R_2\}$ and $\overline{D}_{D_2} = \{D_2, R_1, R_2\}$. Moreover, $K(D_1) = \{R_2\}$ and $K(D_2) = \{R_1\}$. Consider for example the decision strategies in Tables 9 and 10. Assume that the information structure is specified by $x(R_2, D_1) = 0$ and $x(R_1, D_2) = 1$. Then consider the path presented in Table 11 and especially the local decision path for $j = D_1$, which is $s^1_{D_{D_1}} = (s_{R_1} = big, s_{R_2} = small, s_{D_1} = no)$.

Consider the constraint (51) and select $j = D_1$ and $k = R_2$. Then $s = s^1_{D_{D_1}}$. We must consider states s'_{R_2} that are different from the specified state in the local decision path. There exist only one option for this, which is $s'_{R_2} = big$. The local decision strategy in 9 evaluated at the local decision path $s^1_{D_{D_1}}$ gives $z(s^1_{D_1}|s^1_{R_1}, s^1_{R_2}) = 1$, since the decision alternative $s^1_{D_1} = no$ is selected for $s^1_{R_1} = big$ and $s^1_{R_2} = small$. Thus, the path is compatible with the decision strategy. Now for the different s'_k it holds that $z(s^1_{D_1}|s^1_{R_1}, s'_{R_2}) = 0$, since the decision alternative $s^1_{D_1} = no$ is not selected for $s^1_{R_1} = big$ and $s^1_{R_2} = big$. Since $x(R_2, D_1) = 0$, the constraint (51) does not hold. Below the constraint is calculated for these parameters.

$$|z(s^1_{D_1}|s'_{R_2}, s^1_{R_1}) - z(s^1_{D_1}|s^1_{R_2}, s^1_{R_1})| \leq x(R_2, D_1) \tag{52}$$

$$\Rightarrow |0 - 1| \leq 0. \tag{53}$$

Thus, the decision strategy is incompatible with the given information structure. If we set $x(R_2, D_1) = 1$, then the constraint is fulfilled and the decision strategy is compatible with the information structure. Alternatively we can adopt a local decision strategy given in Table 12 for D_1 and then the constraint would be fulfilled with the assumed information structure of $X(R_2, D_1) = 0$ and $X(R_1, D_2) = 1$.

Table 9: A local decision strategy Z_{D_1}

s_{R_1}	s_{R_2}	s_{D_1}
big	*big*	*yes*
big	*small*	*no*
small	*big*	*no*
small	*small*	*no*

Table 10: A local decision strategy Z_{D_2}

s_{R_1}	s_{R_2}	s_{D_2}
big	*big*	*yes*
big	*small*	*yes*
small	*big*	*no*
small	*small*	*no*

Table 11: Illustrative path

	s_L	s_{R_1}	s_{R_2}	s_{D_1}	s_{D_2}	s_F
s^1	*big*	*big*	*small*	*no*	*yes*	*no*

Table 12: An alternative local decision strategy $Z^*_{D_1}$

s_{R_1}	s_{R_2}	s_{D_1}
big	big	no
big	small	no
small	big	yes
small	small	yes

3.7 Constraints on extended state space

Another way to represent the problem is to extend the state space of some nodes. Assume that $j \in D$ has an information set $I(j)$ and a set $K(j)$. We enumerate the state space of each $i \in I(j)$ as $S_i = \{1, ..., n\}$. Then we extend the state space of each $k \in K(j)$ to include a *zero state* $s_k = 0$, which represents the situation where the information of node k is not available to support the decision at j. We define the augmented state space for a node as

$$S_k^\circ = \{0, 1, ..., n\}, \forall j \in D, k \in K(j). \tag{54}$$

Now instead of paths or local decisions, we have to analyse the local decisions on extended information states. In previous sections, a local decision strategy was a decision alternative for each realization of the information state of a decision node.

Now we cannot necessarily select a decision alternative for each realization of the information state due to the fact that some nodes in $K(j)$ can be only in state $s_k = 0$. Thus, we have to select the decision alternative for the information state that is compatible with the information structure. For this purpose we introduce a zero-extension operator $E^* : S_{\bar{I}(j)} \mapsto S^\circ_{\bar{I}(j)}, j \in D$

$$E^*(s_{\bar{I}(j)}) = \{s' \in S^\circ_{\bar{I}(j)} | s'_i = s'_i, s'_k \in \{0, s_k\}, \forall i \in \bar{I}(j) \backslash K, K \subseteq K(j), k \in K\} \tag{55}$$

Assume that $j \in D$, $\bar{I}(j) = \{a, b\}$ and $K(j) = \{b\}$. Assume also an information state $s_{\bar{I}(j)} = (1, 1)$. Then the zero-extension operator for $s_{\bar{I}(j)}$ gives

$$E^*(s_{\bar{I}(j)}) = \{(1, 1), (1, 0)\}. \tag{56}$$

We require that exactly one decision alternative and one alternative from the set of extended information states are selected.

$$\sum_{s_j \in S_j} \sum_{s^\circ \in E^*(s_{\bar{I}(j)})} z(s_j|s^\circ) = 1, \forall s_{\bar{I}(j)} \in S_{\bar{I}(j)} \tag{57}$$

The selected alternative from the set of extended information states depends on the selected information structure. When $x(k,j) = 0$, we need to select a decision alternative for an extended information state, where $s_k = 0$. When $x(k,j) = 1$ we need to select a decision alternative for all information states, where $s_k > 0$. These requirements can be enforced through

$$\sum_{s_j \in S_j} \sum_{\{s' \in S^\circ_{\bar{I}(j)} | s'_k > 0\}} z(s_j|s') \leq Mx(k,j), \forall k \in K(j) \tag{58}$$

and

$$\sum_{s_j \in S_j} \sum_{\{s' \in S^\circ_{\bar{I}(j)} | s'_k = 0\}} z(s_j|s') \leq M[1 - x(k,j)], \forall k \in K(j), \tag{59}$$

where M is a large constant. In this thesis we set M to be the number of paths for the influence diagram, due to the fact that the sum on the left side of the constraints can be quite different depending on the selected information structure. We must select a decision alternative for each realization of states of nodes in the information set of the decision node. The number of different combinations for $s_{I(j)}$ is given by $\prod_{p \in I(j)} |S_p|$. If the nodes in $K(j)$ are not known when making a decision at j, then the number of different combinations for which we select a decision alternative is again $\prod_{p \in I(j)} |S_p|$. This is because the only possible value for s_k is the zero value. However, if the information given by a node $k \in K(j)$ is made available when making a decision at j, then we must also select a decision alternative for the different combinations of $\{s_k\} \cup \{s_{I(j)}\}$. This corresponds to $|S_k| \prod_{p \in I(j)} |S_p|$. Thus it follows that

$$\sum_{s_j \in S_j} \sum_{s' \in S^\circ_{\bar{I}(j)}} z(s_j|s') = \prod_{p \in I(j)} |S_p| \prod_{\{k \in K(j) | x(k,j) = 1\}} |S_k|. \tag{60}$$

Thus, the constant M is needed to ensure that that the sum on the left in constraints (58) and (59) stays below the right side of the constraint in cases, where $x(k,j) = 1$. It suffices to select a large enough M. Since it holds that

$$\prod_{p \in I(j)} |S_p| \prod_{\{k \in K(j) | x(k,j) = 1\}} |S_k| \leq |S|, \tag{61}$$

selecting M to be the number of paths guarantees that the constraints (58) and (59) hold.

We also have the constraint on path probabilities

$$\pi(s) \leq \sum_{s^\circ \in E^*(s_{\bar{I}(j)})} z(s_j|s^\circ), \forall s \in S, \forall j \in D \tag{62}$$

The mutual exclusivity of the information states ensures that the sum in constraint (62) has at most one positive term. This proposition is proved as follows.

Theorem 3.2. *Let $j \in D$ and $s \in S$. Assume that constraints (58) and (59) hold and that $K(j)$ is a nonempty set. Then the sum $\sum_{s^\circ \in E^*(s_{\bar{I}(j)})} z(s_j|s^\circ)$ has at most one positive term.*

Proof. Assume that there is a decision s_j and two distinct information states $s'_{\bar{I}j}, s''_{\bar{I}(j)} \in E^*(s_{I\bar{(j)}}), s'_{\bar{I}(j)} \neq s''_{\bar{I}(j)}$ such that $z(s_j|s'_{\bar{I}(j)}) = z(s_j|s''_{\bar{I}(j)}) = 1$. By the definition of E^*, there exists $k \in K(j)$, such that $s'_k = s_k$ and $s''_k = 0$. Assume that $x(k,j) = 0$. Then according to constraint (58) $z(s_j|s') \leq \sum_{s_j \in S_j} \sum_{\{s^\circ \in S^\circ_{\bar{I}(j)} | s^\circ_k = s^\circ_k\}} z(s_j|s^\circ) \leq 0$, which is a contradiction to the assumption. Similarly, if $x(k,j) = 1$ then according to constraint (59) $z(s_j|s'') \leq \sum_{s_j \in S_j} \sum_{\{s^\circ \in S^\circ_{\bar{I}(j)} | s^\circ_k = 0\}} z(s_j|s^\circ) \leq 0$, which again is a contradiction to the assumption. Thus, there can be at most one positive term in the summation of (62). □

3.8 Optimization model with constraints on extended state space

Adding the constraints on extended state space to the Decision Programming framework gives the following optimization model

$$\max_{x\in X}\max_{z\in Z} \quad \sum_{s\in S}\pi(s)U(s)-\sum_{j\in D}\sum_{k\in K(j)} x(k,j)R(k,j) \tag{63}$$

$$\text{s.t.} \quad \sum_{s_j\in S_j}\sum_{s^\circ\in E^*(s_{\bar{I}(j)})} z(s_j|s^\circ)\le 1, \qquad \forall j\in D, s_{\bar{I}(j)}\in S_{\bar{I}(j)} \tag{64}$$

$$0\le\pi(s)\le p(s), \qquad \forall s\in S \tag{65}$$

$$\pi(s)\le\sum_{s^\circ\in E^*(s_{\bar{I}(j)})} z(s_j|s^\circ), \qquad \forall s\in S,\forall j\in D \tag{66}$$

$$z(s_i|s_{\bar{I}(i)})\in\{0,1\}, \qquad \forall i\in D,\forall s\in S,\forall s_{\bar{I}(i)}\in S_{\bar{I}(i)} \tag{67}$$

$$\sum_{s_j\in S_j}\sum_{\{s'\in S^\circ_{\bar{I}(j)}|s'_k>0\}} z(s_j|s')\le Mx(k,j), \qquad \forall j\in D,k\in K(j) \tag{68}$$

$$\sum_{s_j\in S_j}\sum_{\{s'\in S^\circ_{\bar{I}(j)}|s'_k=0\}} z(s_j|s')\le M[1-x(k,j)], \quad \forall j\in D,k\in K(j) \tag{69}$$

Notice that constraint (64) is changed from equality to inequality compared to the cosntraint that was presented in previous subsection. This is possible because the change enables a polyhedron as the feasible area. The optimum of a polyhedron is found at an extremum point and thus the change does not affect the result. Instead it makes the optimization problem convex and thus quicker to compute. This requires also that the target function, which is the utility of the decision consequences added to the costs of the selected information structure, is scaled such that the minimum value of the utility function is greater than zero.

In the 2-monitoring example, assume that the information structure is given by $x(R_1, D_2) = 1$ and $x(R_2, D_1) = 0$. The number of paths in the 2-monitoring problem is $2^6 = 64$ and thus we set $M = 64$. Consider the local decision strategies Z_{D_1} and Z_{D_2} presented in Tables 13 and 14. The local decision strategy at node D_1 gets the value 1 only when $s_{R_2} = 0$. Thus we get that the sum of local decision strategies at $j = D_1$ in constraint (68) is 0. Conversely the sum of local decision strategies at $j = D_1$ in constraint (69) is 2. The specified information structure states that $x(R_2, D_1) = 0$, so both constraints are fulfilled and the decision strategy is compatible with the information structure. At decision node D_2 we get that the sum of local decision strategies in constraint (68) is 4 and the sum of local decision strategies in constraint (69) is 0. Thus when we substitute $x(R_1, D_2)$ and $M = 64$ to the constraints, we get that the decision strategy is compatible with the information structure.

Table 13: A local decision strategy Z_{D_1}

s_{R_1}	s_{R_2}	s_{D_1}
big	0	*yes*
small	0	*no*

Table 14: A local decision strategy Z_{D_2}

s_{R_1}	s_{R_2}	s_{D_2}
big	*big*	*yes*
small	*big*	*no*
big	*small*	*no*
small	*small*	*no*

3.9 Analysis of computational requirements

We have presented three approaches for the structure of the influence diagram. Here, we analyse the computational requirements for solving the problem. The evaluation of an influence diagram can be a computationally demanding task. Even the most simple diagrams may require a lot of computational power [32]. If the complexity of the diagram grows, so does the required computational power. For example, the solution of limited memory influence diagrams are in the worst case evaluated in exponential time [18].

However, the Decision Programming framework, which is the basis of this thesis computes the optimal decision strategy quite efficiently although the problem is NP-complete. As noted in [8], the sharing of reports in N-monitoring example increases the number of analysed paths considerably but can still be computed in a reasonable time when the number of instances is 4. However, to optimize the structure of the influence diagram using the Decision Programming framework, the size of the problem grows fast when introducing new conditional edges. In what follows we discuss the number of constraint that each optimization model adds to the decision model.

The number of constraints on path probabilities are proportional to the number of paths in the influence diagram. For each conditional edge $(k, j), j \in D, k \in K(j)$ there are constraints for each state s_j and $s_{\bar{I}(j)}$ and for each path $s' \in E(s_{I(j)}) \cap \overline{E(s_j)} \cap \overline{E(s_k)}$. Thus the number of constraints per conditional edge (k, j) depends on the size of the extension, which is given by $\prod_{i \in N \setminus j, \bar{I}(j)} |S_i|$ and the size of the state space of nodes in $\bar{I}(j)$ and j. The size of the problem grows quite significantly when the number of paths grows and we introduce more conditional edges to the problem. Thus, the number of constraints on path probabilities (constraint (39)) is

$$\sum_{j \in D} \sum_{k \in K(j)} |S_j||S_{\bar{I}(j)}| \prod_{i \in N \setminus \{j \cup \bar{I}(j)\}} |S_i|. \tag{70}$$

The constraints on local decisions can handle larger problems better than the constraints on path probabilities can. Instead of building constraints for full paths it uses the augmented information sets of the decision nodes. For each conditional edge (k, j), the number of constraints depends on the size of the local decision path $|S_{D_j}|$ and the size of the state space of node k. Each s_{D_j} and each $s'_k \neq s_k$, for the s_k stated in the local decision path s_{D_j} is constrained to follow the information structure. Thus the number of constraints per distinct local decision path s_{D_j} is $|S_k| - 1$, since the path contains only one realization from S_k. The number of constraints on local decisions is then given by

$$\sum_{j \in D} \sum_{k \in K(j)} |S_{D_j}|(|S_k| - 1). \tag{71}$$

Constraints on the extended state space introduce far less constraints to the optimization model. With this method, the constraints do not grow when new nodes are added to the information set of a decision node. For each $k \in K(j)$, $j \in D$ we have only two constraints. The number of extended state space constraints is then given by

$$\sum_{j \in D} \sum_{k \in K(j)} 2 \tag{72}$$

However, the extended state space constraint method increases the state space of nodes in $K(j), j \in D$. This grows the number of decision strategy variables which have to be optimized, which is not the case with the other two methods. Thus, smaller number of added constraints does not imply that the optimization model is more efficient.

The number of extended state space constraints is not proportional to the size of the state space of the nodes in the influence diagram unlike the other methods. This suggests that when the state space of nodes in the influence diagram grows, the constraints on extended state space are superior to the other methods. The constraints on path probabilities and the constraints on local decisions are quite similar to each other. However, a significant difference is that the constraints on path probabilities are set for the extensions of path segments instead of local decision paths. This suggests that when the influence diagram contains multiple decision nodes, which have relatively small information sets, the constraints on local decisions are superior to the constraints on path probabilities. However, when the problem consists of only one decision node and it's information set, then the difference between the number of constraints is smaller.

Explicit enumeration is a good comparison for all optimization methods. In explicit enumeration, one would iterate over all possible information structures and all possible decision strategies and select the one with the best expected utility. To calculate the expected utility of a single decision strategy and an information structure, one would need to consider all possible information structures and all possible decision strategies. For each information structure and decision strategy, there is a need to calculate the utility and the probability of each path. Explicit enumeration then is proportional to the number of paths, to the number of possible decision strategies and to the number of possible information structures. Since the decision strategy variables and information structure variables are binary variables, we would need to calculate the utility and probability of a path $|S|2^{|Z|}2^{|X|}$ times.

4 Computational experiments

This section examines the computational performance of the models. All computations were performed using Julia 1.6.2 in Aalto University's jupyter environment. The optimization models were solved using the JuMP framework with GLPK-optimizer. All used codes can be accessed via this link

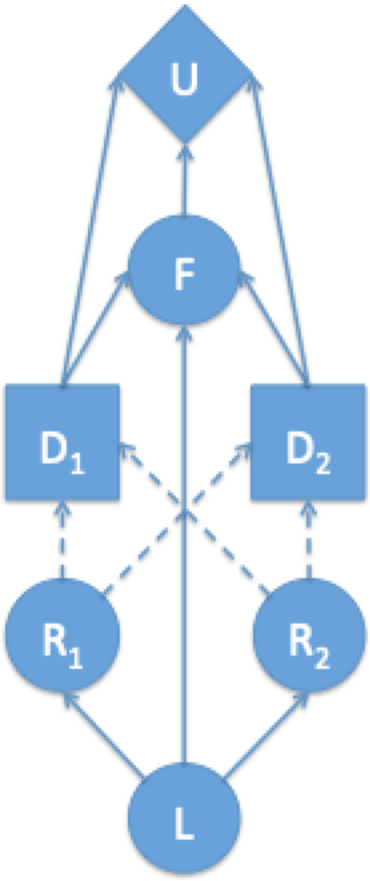
4.1 N-M-monitoring problem

The N-M-monitoring problem is similar to the 2-monitoring problem presented above with the exception that now there are N reports and M decisions. In addition, all edges between reports and decisions belong to the set $K(j), \forall j \in D$. The influence diagram of the problem is in Figure 4. The probabilities between the reports and the load as well as between the decisions and the failure event are randomly generated. For the load probabilities a random number between 0 and 1 is generated using the rand-function of julia language. The probability of load being in state *high* is the generated number and the probability of load being in state *low* is 1 minus the generated number. Similarly, all the report probabilities are generated using the rand-function. This time two random numbers between 0 and 1 are generated using the rand-function.

The first of the generated numbers is used as the probability of a true positive report if it lies between 0.5 and 1. If the generated number is less than 0.5, then the probability of a true positive report is 1 minus the generated number. Similar procedure is utilized for the second generated random number as the probability of a true negative report. Similarly, the probability of failure given the load and fortification actions is created using two random numbers between 0 and 1. The first generated number is used as the probability of failure given that the load is in state *high*. The probability of no failure when the load is in state *high* is thus 1 minus the random number.

The second random number is then the probability of failure when the load is in state *low*. Since the failure event also depends on the fortification actions, a special denominator is created for each iteration of the possible states of action nodes. This random number is constrained on being slightly greater than 1. Thus the probability of failure given that the load is in state *high* and the probability of no failure given that the state of load is *low* are divided by the denominator. This way we get unique probabilities for all realizations of failure node's information state. The costs of reports are random numbers between 0 and 10 and the costs of fortification actions are the same as the randomly generated denominators for the failure probabilities. The utility of the system is 0 minus the fortification costs if the system collapses, and 100 minus the fortification actions if the system does not collapse. The utilities are normalized such that the minimum utility is 0. All problems were solved 10 times with different randomly generated starting values and the average of these solve times is presented.

Figure 4: Influence diagram of the N-M-monitoring problem, where N = 2 and M = 2



The N-M-monitoring problem was solved using the constraints on path probabilities, local decisions and extended states. The models were solved ten times for each N-M pair and Table 16 shows the solution times for each optimization model. The problem was solved by relaxing the binary requirement of the decision strategy variables and instead constraining them to lie between 0 and 1. As noted in [8] this also leads to optimality.

Table 15: Solution times for different N and M in seconds

N	M	path prob.	local decisions	extended states
1	1	0.000195	0.000178	0.000218
2	1	0.000517	0.000435	0.000788
2	2	0.00238	0.00132	0.00304
3	2	0.00882	0.00463	0.01133
3	3	0.04316	0.01256	0.02985
4	3	0.15849	0.04726	0.07990
4	4	0.76935	0.13819	0.25490
5	4	3.93886	0.66613	2.66954
5	5	19.3005	1.81897	5.03522

The solution times grow quite fast and already when $N = 5$ and $M = 5$, the optimal solution takes several seconds to find. However, the computation time of the models is still quite reasonable. The results are quite consistent. Constraints on local decisions are the fastest, whereas the constraints on path probabilities and the constraints on extended state space are the slowest depending on problem size. On smaller instances the extended state space constraints are slower than the constraints on path probabilities whereas on bigger instances the constraints on extended state space are quicker than the constraints on path probabilities. This suggests that the computation time of constraints on path probabilities grows faster than the one of constraints on extended state space.

4.2 Oil wildcatter problem

Consider the traditional oil wildcatter problem presented in section 2.1 with the exception that the wildcatter has multiple expert reports to choose from. The influence diagram of the modified oil wildcatter problem where the number of possible reports is 3 is in Figure 5. We solve the problems of different sizes and compare the solution times that the different optimization models give. We assume that the probability of the existence of oil is 0.15, which then means that the probability of no oil is 0.85. The probabilities of individual reports given the existence of oil are created as follows: First, four numbers between 0 and 1 are randomly created using the rand-function of julia language. These numbers are used as parameters for two different normal distributions. Two of them are used as means and two as variances. Then two normal distributions are generated using these values, and 100 observations are sampled from the normal distributions. The normal distribution with the lower mean is taken to be the test statistic distribution of test results for test where there are no oil in the ground. The normal distribution with the higher mean is then taken to be the test statistic distribution for test results where there is oil in the ground. Using the randomly sampled values, a cutoff point is estimated s.t. the amount of false negative results and false positive results is as small as possible. Then by using this cutoff point, a false positive rate and a false negative rate are calculated. These rates are then used as the probabilities for achieving a false negative test result and a false positive test result. Thus, the probability of a true positive test is one minus false negative. Costs of tests are randomly created as integers between 0 and 50 and the highest costs are associated to the test that gives the lowest false positive rate. The cost of drilling is 300 and the potential payoff is 7000. The problem is randomly generated and solved twenty times for each problem size. Each problem is solved by all three methods and an average of solution times is reported. Table 16 gives the computation times for the different solution methods for $N \in \{1..6\}$.

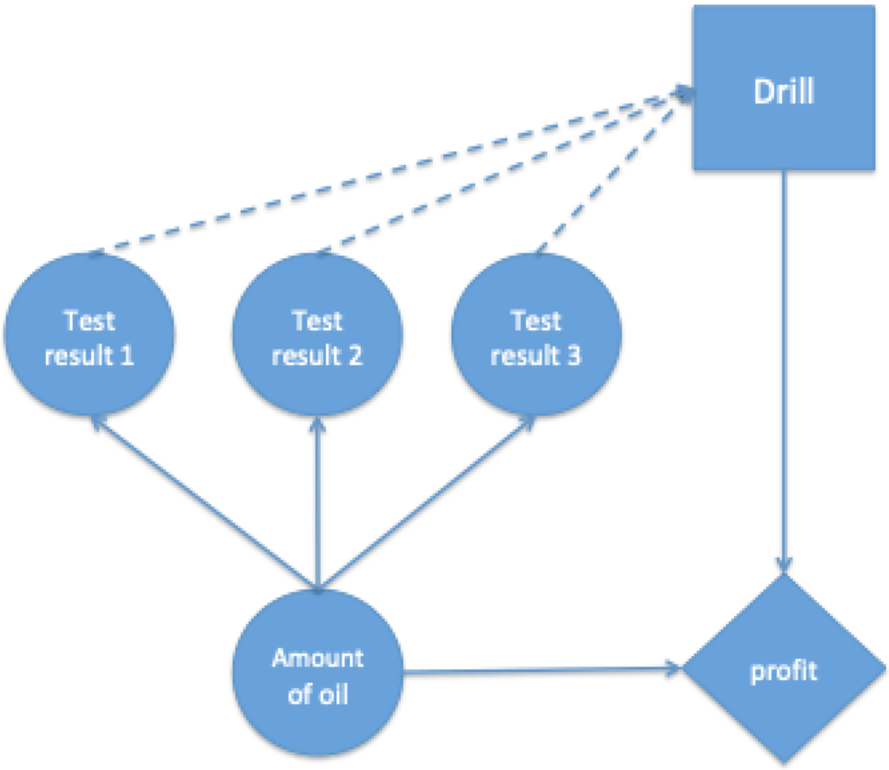
Table 16: Solution times for different N in seconds

N	path prob.	local decisions	extended states	Avg number of selected edges
1	0.000186	0.000183	0.000188	0.25
2	0.000602	0.000518	0.000661	0.75
3	0.002399	0.002267	0.003676	1.00
4	0.01003	0.01243	0.01801	1.28
5	0.07974	0.18375	0.14647	1.53
6	0.99904	30.7098	7.73159	1.8

The solution times of the three methods are quite similar to each other in smaller instances. The constraints on local decisions are slightly quicker than the other models in small problem sizes but when the problem size grows, the constraints on local decisions slow down relative to the others. The constraints on path probabilities is the fastest method in bigger instances as the average time to optimality takes approximately a second on average, whereas the other methods take several seconds. This is unexpected considering that the number of constraints for the path probabilities is the highest of the three. The number of added edges seems quite reasonable. Approximately a third of the possible reports were selected to support the decisions.

We can also test the optimization models when relaxing the binary requirement of the decision variable z. The relaxed oil wildcatter problem is solved with $N \in \{1, ..., 6\}$

Figure 5: Influence diagram of the modified oil wildcatter problem



and the average solution times are highlighted in Table 17. The results are quite similar as to the N-M-monitoring example in previous section. Constraints on local decisions are the fastest in all problem sizes. Now however, the constraints on extended state space are quicker than the constraints on path probabilities in smaller problem sizes whereas the constraints on path probabilities are faster than the constraints on extended state space in bigger problem sizes. The relaxation has a significant impact to the solution times. Especially in bigger instances the impact is big. Solution with the constraints on local decisions took approximately 1000 times longer without the relaxation.

Table 17: Solution times for different N with relaxation

N	path prob.	local decisions	extended states	avg no of edges
1	0.000152	0.000112	0.000142	0.25
2	0.000479	0.000265	0.000367	0.65
3	0.001792	0.000773	0.001724	1.075
4	0.00762	0.00243	0.00830	1.275
5	0.02345	0.01175	0.06242	1.7
6	0.11159	0.04725	0.49111	1.575

4.3 Extended oil wildcatter problem with a perfect report

Consider again the extended oil wildcatter problem. The exception to the previous subsection is that now there exists a perfect report that indicates the existence of oil with certainty. In addition, there are $n-1$ imperfect reports, which are much cheaper than the perfect report. When demonstrating the results, the perfect report is always denoted as R_n, where n is the size of the problem. The goal of this exercise is to determine how many imperfect but uncorrelated reports are needed to replace the perfect report.

The parameters of the problem are modified as follows. The potential payoff when drilling for oil is 11000. This payoff is achieved when the decision to drill is *yes* and there is oil in the ground. The cost of drilling is 1000 and the cost of the perfect report is 400. The probability of oil is again 0.15 whereas the probability of no oil is then 0.85. The imperfect reports are equal to each other and the probability of a false negative test result is the same as the probability of a false positive test result. This means that the probability of the report indicating oil in the ground given that there is oil in the ground is the same as report indicating no oil when there isn't oil in the ground. Table 18 shows some solutions for the problem. The results show that the imperfect reports can be quite useful when combining them with other uncorrelated imperfect reports. Imperfect reports can also be more useful outright as we can see in the last row of the table. The model has only chosen to run one report, which indicates the existence of oil correctly with a probability of 0.9. The cost of the report is so small compared to the perfect report that it gives better expected utility than using the perfect report.

Table 18: Results

N	$P(s_{R_i} = \text{yes} \mid s_O = \text{yes})$	Cost	Added edges
3	0.8	20	(R_3, D)
4	0.8	20	(R_1, D), (R_2, D), (R_3, D)
4	0.75	20	(R_4, D)
5	0.75	20	(R_1, D), (R_2, D), (R_3, D), (R_4, D)
3	0.9	50	(R_2, D)

4.4 Order of tests

Consider another variant of the oil wildcatter problem. The wildcatter has the ability to perform tests for the potential drilling site but this time the subject of interest is the order of the tests. The assumption is that the wildcatter can perform a single test at a time and then must choose wether to drill, run another test or not to drill. Once a drilling decision has been made, the wildcatter cannot change it anymore.

The analyzed variant is quite small. The wildcatter has only two tests that can be utilized. Thus the wildcatter must first make a decision wether to run one of the possible tests. Then a decision can be made to wait for the result of another test or to drill straight away based on the result of the first test. The influence diagram of the problem is presented in Figure 6.

The state space of decision in period 1 is now $\{yes, no, wait\}$, where wait means that the decision is made in period 2. If decision alternatives *yes* or *no* are already agreed in period 1, they must stay the same in period 2 as well. Thus, path probabilities that contain violations of this principle must be set to 0. In addition, we are limited to one test per period. If we opt to do no testing in period 1, then we cannot run both tests in period 2. In addition we assume that if a test is selected in period 1, the result of that test is also available when making a decision in period 2. These requirements can be enforced through separate constraints.

$$x(j, D_1) \leq x(j, D_2), \forall j \in \{Test1, Test2\} \tag{73}$$

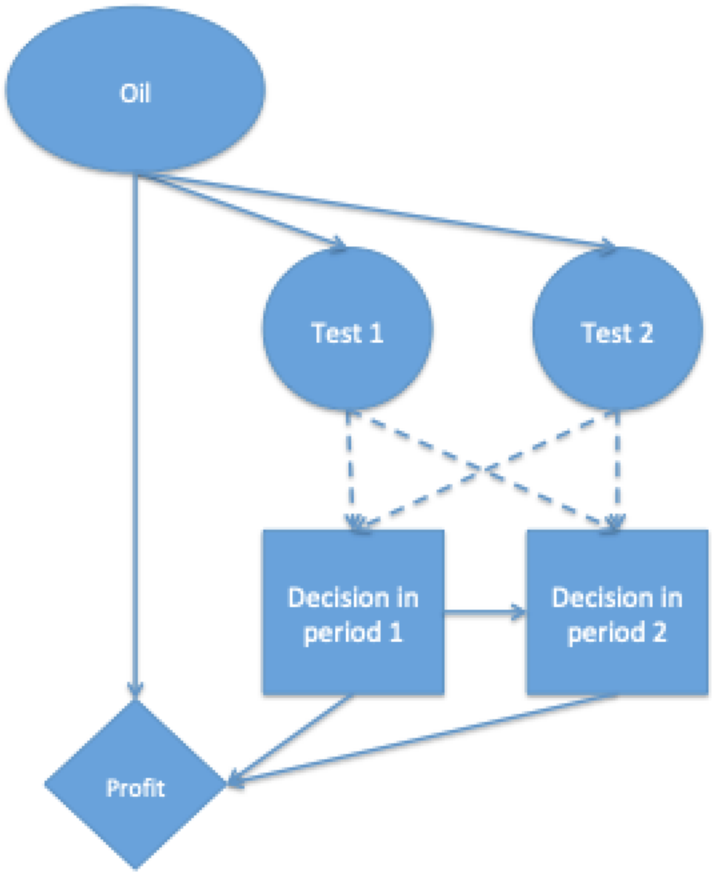
$$x(Test1, D_1) + x(Test2, D_1) \leq 1 \tag{74}$$

$$x(Test1, D_2) + x(Test2, D_2) \leq 1 + x(Test1, D_1) + x(Test2, D_1) \tag{75}$$

The problem was calculated with tests specified in Table 19.

Table 19: Test probabilities

Figure 6: Influence diagram of the order of tests problem



O	T	P(T$_1$|O)	P(T$_2$|O)
yes	yes	0.5	0.93
yes	no	0.5	0.07
no	yes	0.07	0.5
no	no	0.93	0.5

The costs of both tests are 25 and the utility if oil is found from the ground is 20000. The cost of drilling is 1000. The cost of selecting the option $s_{D_1} = wait$ is 100. The cost of waiting simulates the time value of money [31]. With these parameters, an optimal decision strategy and optimal information structure are calculated. The decision strategy is presented in Tables 20 and 21. The optimal information structure is presented in Table 22. Thus, the optimal information structure is to run test 2 in period one and then test 1 in period 2. If test 2 indicates that there is oil in the ground, we decide to drill immediately. If the test shows that there is no oil in the ground, we will instead order the second test and decide on drilling in the second period. The expected utility of the system is 2400,78.

This example also highlights a significant shortcoming of the framework, which is that the information structure cannot be state dependent. If the first test shows that there is oil in the ground, we already decide to drill in period 1. Then we couldn't change the decision in period 2 anymore due to the presented constraints. However, the optimal information structure states that test 1 is ran in period 2 no matter the result of test 2 and the decision in period 1. Therefore, this example highlights a problem, where the tests have to be prebooked and they cannot be ordered on the fly.

Table 20: Optimal local decision strategy $Z^*_{D_1}$

s_{T_1}	s_{T_2}	s_{D_1}
yes	yes	yes
no	yes	yes
yes	no	wait
no	no	wait

Table 21: Optimal local decision strategy $Z^*_{D_2}$

s_{D_1}	s_{T_1}	s_{T_2}	s_{D_2}
wait	yes	yes	yes
wait	no	yes	yes
wait	yes	no	yes
wait	no	no	no

Table 22: Optimal information structure

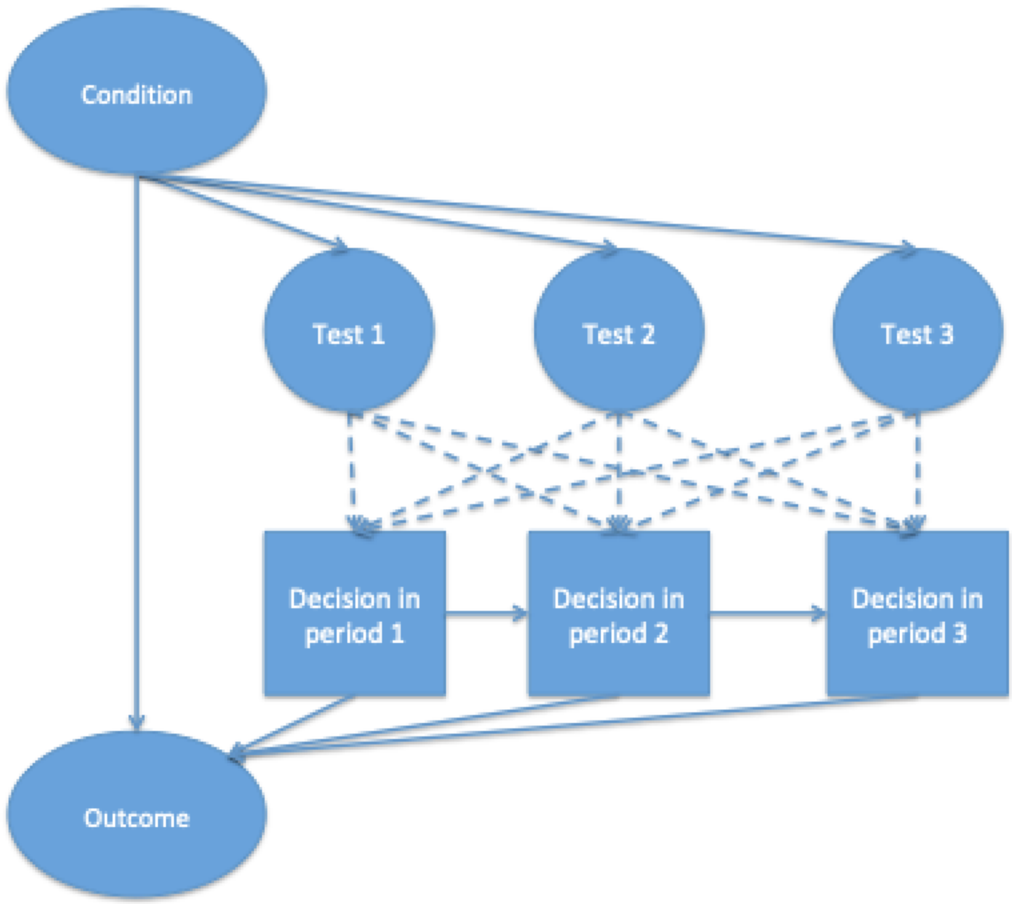
$x(Test1, D_1)$	0
$x(Test2, D_1)$	1
$x(Test1, D_2)$	1
$x(Test2, D_2)$	1

5 Possible applications and extensions

The extension of influence diagrams to alternative information structures has many potential applications. Posing the question of what information should be provided to the decision maker is valuable in certain situations. We have applied the models successfully to the N-M-monitoring problem and to modified oil wildcatter problems. The given framework gives intriguing possibilities for these kind of sequential decision problems, where there are multiple parallel decisions to be made. The given framework could be utilized to consider the order of decisions. For example in the case of the N-M-monitoring problem a decision D_1 could be expanded to different stages, where we would choose from three different options $S_{D_1} = \{yes, no, wait\}$. This way we could analyse sequential decisions $D_1^1 ... D_1^n$ in contrast to the sequential decisions $S_2^1 ... D_2^n$. We would then be able to analyse potential edges between decision pairs (D_1^i, D_2^{i+1}) and thus get the optimal ordering of the decisions by optimizing the information structure. The difference to the order of tests example in section 4.4 is that we would consider edges between various decision nodes. It would also be of value to consider some cases where the order of the decision is somewhat random but the decision maker nevertheless would know the realizations of previous decisions.

One obvious application area is health care resource optimization. As the resources for health care are limited, the testing decisions for different diseases mean that there may not be enough resources left for other tests [33]. The test results are available for the doctors if a test decision is made for a certain disease. Thus it is a natural application area for the presented frameworks. Consider for example the resource allocation problem discussed in [34], which presents utilitarian and egalitarian decision models for the testing decisions of coronary heart disease. A utilitarian approach for health care resource allocation is to maximize the aggregate population health, whereas the egalitarian approach is to minimize health differences between population segments. [35] An influence diagram representation for the optimal information structure is presented in Figure 7. Here, chance nodes represent tests that can be run in each period and the decisions represent treatment decisions that have to be made based on the tests that have been done. If the treatment is started in period 1 or 2 then it needs to be continued also in period 3. The limited resources could be represented by limiting the number of added edges for example to three with the following constraint: $\sum_{j \in D, k \in K(j)} x(k, j) \leq 3$. Influence diagrams are already a useful tool in medical decision analytics [36] but as presented above, our framework gives a lot of flexibility to the problem layout. The same approach could be also used in contingent portfolio programming developed in [37].

Figure 7: Influence diagram of the healthcare resource allocation problem



In this thesis, the Decision Programming framework has been extended to situations where the addition of an edge is deterministic and thus the structure of the influence diagram is defined at the onset. The framework given by Herrala et. al. can handle also situations, where the existence of edges depends on some random events. It would be of value to extend the Decision Programming framework to handle also uncertainties that affect to the information structure of the problem.

6 Conclusions

The thesis has presented three models for optimizing the information structure of an influence diagram. Two of the models rely on constraints that force decisions to be equal in cases where information that is not known changes and other information stays the same. The optimization models differ in the extent on which they enforce constraints. The third optimization model forces the decisions to be made without the information of nodes that are not available.

The constraints on path probabilities analyse full paths and thus the number of constraints grows quite fast as a function of the nodes in the decision problem. For example, adding barren nodes to the influence diagram does not change the optimal decision strategy or the optimal information structure in any way, but the number of paths grows and thus the number of constraints also grows. The constraints on local decisions analyse only augmented information sets and their combinations so the number of constraints grows when the augmented information sets grow.

The constraints on extended state space introduce a zero-state which indicates that the information given by a node with the zero-state may not be known. Thus a decision strategy is only created for an information state that is compatible with the information structure.

All methods were tested with example problems. The methods perform quite similarly in small problems. When the size of the problem grows, the differences between the models grow. The constraints on local decisions are superior to the other models in all problem sizes and types. Constraints on extended state space and constraints on path probabilities were quite similar to each other. In N-M-monitoring problem the constraints on path probabilities were faster in smaller instances whereas the constraints on extended state space were faster in larger instances. In the modified oil wildcatter problem the order of the two models was reversed. The constraints on extended state space was faster in smaller instances whereas the constraints on path probabilities were faster on larger instances. The approaches are applicable to problems with fairly many decision nodes and multiple different information structures but there are limitations. In N-M-monitoring problem, the maximum problem size that was solved using the introduced models was $N = 5$ and $M = 5$. The modified oil wildcatter problem was solved with a maximum of 6 reports. The relaxation of the decision strategy variables has a significant impact on the computational requirement. The N-M-monitoring problem was already impossible to solve with $N = 3$ and $M = 3$ without the relaxation. The contribution can be quite valuable in decision problems with multiple decisions that give more information on some target. As evidenced by the extended oil wildcatter problem that was solved above, the optimization models can handle multiple testing decisions. Now all solved models had only two states per node. However, when the initial state space of the nodes grows, then the constraints on extended state space could work better than the other models.

All models assume that the information structure can be decided before the chance events and decisions represented by nodes in the influence diagram take place. Contrary to the assumption, the information structure may not be fully known beforehand. An important addition to the decision model would be a non-deterministic view of the realization of the information structure.

References

[1] Howard, R.A., Matheson, J.E. *Influence diagrams* Decision Analysis, 2005, vol. 2:3, pp. 127–143. https://doi.org/10.1287/deca.1050.0020

[2] Olmsted, M. 1983. *On representing and solving decision problems* (PhD Dissertation). Stanford University, Stanford. CA.

[3] Keefer, D.L., Kirkwood, C.W., Corner, J.L. *Perspective on decision analysis applications, 1990–2001.* Decision Analysis, 2004, vol. 1:1, pp. 5-24

[4] Varis, O., Kettunen, J., Sirviö, H. *Bayesian influence diagrams in complex environmental management including observational design* Computational Statistics & Data Analysis. 1990, vol 9, pp. 77-91.

[5] Virtanen, K., Raivio, T., Hämäläinen, R.P. *Modeling pilot's sequential maneuvering decisions by a multistage influence diagram*, Journal of Guidance, Control and Dynamics, 2004, vol 27:4, pp.665-677

[6] Lauritzen, S.L., Nilsson, D *Representing and solving decision problems with limited information* Management Science, 2001, vol 47:9, pp. 1235-1251, https://doi.org/10.1287/mnsc.47.9.1235.9779

[7] Mancuso, A., Compare, M., Salo, A., Zio, E. *Portfolio optimization of safety measures for the prevention of time-dependent accident scenarios.* Reliability Engineering & System Safety, 2019, vol 190, 106500

[8] Salo, A., Andelmin, J., Oliveira, F *Decision programming for mixed-integer multi-stage optimization under uncertainty* European Journal of Operational Research, 2022, vol. 299:2 pp. 550-565.

[9] Raiffa, H. *Decision analysis.* 1968, Addison-Wesley, Reading, Mass

[10] Zhang, L. L., Qi, R., Poole, D. *A computational theory of decision networks.* International Journal of Approximate Reasoning, 1994, vol 11, pp. 83–158.

[11] Kjærulff, U.B., Madsen, A.L. *Bayesian networks and influence diagrams: A guide to construction and analysis*, 2013, 2nd ed, New York, NY : Springer, ISBN 978-1-4614-5104-4

[12] Shachter, R.D. *Evaluating influence diagrams* Operations Research, 1986, vol 34:6, pp. 871-882

[13] Shachter, R. D. *Probabilistic inference and influence diagrams.* Operations Research, 1988, vol. 36, pp.589-604.

[14] Tatman, J.A., Shachter, R.D. *Dynamic programming and influence diagrams* IEEE Transactions on Systems, Man, and Cybernetics, 1990 vol. 20:2, pp. 365-379, https://doi.org/10.1109/21.52548.

[15] Qi, R., Poole, D. *A new method for influence diagram evaluation* Computational Intelligence, 1995, vol. 11:3, pp. 498-528

[16] Charnes, J. M., Shenoy, P. P. *Multistage Monte Carlo method for solving influence diagrams using local computation*, Management Science, 2004, vol. 50:3, pp. 405-418

[17] Shachter, R.D., Bhattacharjya, D. *Solving Influence Diagrams: Exact Algorithms.* Wiley Encyclopedia of Operations Research and Management Science, 2010

[18] Mauá, D. D., de Campos, C. P., Zaffalon, M. *Solving limited memory influence diagrams.* Journal of Artificial Intelligence Research, 2012, vol 44. pp. 97-140

[19] Mauá, D. D., de Campos, C. P., Zaffalon, M. *On the complexity of solving polytree-shaped limited memory influence diagrams with binary variables* Journal of Artificial Intelligence Research, 2013, vol 205. pp. 30-38

[20] Parmentier, A., Cohen, V., Leclére, V., Obozinski, G., Salmon, J. *Integer programming on the junction tree polytope for influence diagrams.* INFORMS Journal on Optimization, 2020, vol 2:3, pp. 209–228.

[21] Mauá, D. D., Cozman, F. G. *Fast local search methods for solving limited memory influence diagrams.* International Journal of Approximate Reasoning, 2016, vol. 68, pp. 1235–1251.

[22] Yuan, C., Wu, X., Hansen, E.A. *Solving multistage influence diagrams using branch-and-bound search* Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence, UAI, 2010, pp. 691-700

[23] Bender, M. A., Fineman, J. T., Gilbert, S,. Tarjan, R. E. *A new approach to incremental cycle detection and related problems*, ACM Transactions on Algorithms, 2015, vol 12:2, pp. 1-22

[24] Weinblatt, H. *A new search algorithm for finding the simple cycles of a finite directed graph.* Journal of the ACM, 1972, vol. 19:1, pp.43-56

[25] Szwarcfiter, J. L., Lauer, P.E. *A search strategy for the elementary cycles of a directed graph* BIT Numerical Mathematics, 1976, vol. 16, pp. 192-204

[26] Hellemo, L., Barton, P.I., Tomasgard, A. *Decision-dependent probabilities in stochastic programs with recourse*, Computational Management Science, 2018, vol. 15:(3-4), pp. 369-395

[27] Jonsbråten, T. W., Wets, R.J.B., Woodruff, D.L *A class of stochastic programs with decision dependent random elements*, Annals of Operations Research, 1998, vol. 82, pp. 83-106

[28] Zhang, Q., Feng, W. *A unified framework for adjustable robust optimization with endogenous uncertainty* AIChe journal, 2020, vol. 66:12

[29] Herrala, O., Ekholm, T., Oliveira, F. *Solving decision problems with decision-dependent probabilities and information structures*, manuscript

[30] Apap, R. M., Grossmann, I. E. *Models and computational strategies for multistage stochastic programming under endogenous and exogenous uncertainties.* Computers & Chemical Engineering, 2017, vol.103, pp. 233-274.

[31] Whitman D.L., Terry R.E. *Fundamentals of Engineering Economics and Decision Analysis*, 2012, 1st ed., Morgan Claypool, ISBN 9781608458653

[32] Cabañas R., Madsen A.L., Cano A., Gómez-Olmedo M. *On SPI for evaluating influence diagrams.* In: Laurent A., Strauss O., Bouchon-Meunier B., Yager R.R. (eds) Information Processing and Management of Uncertainty in Knowledge-Based Systems. IPMU 2014. Communications in Computer and Information Science, vol 442. Springer, Cham.

[33] Hunink, M., Glasziou, P., Siegel, J. E., Weeks, J. C., Pliskin, J. S., Elstein, A. S., *Decision making in health and medicine: Integrating evidence and values*, Cambridge, UK: Cambridge University Press, 2001.

[34] Hynninen, Y., Vilkkumaa, E., Salo, A. *Operationalization of utilitarian and egalitarian objectives for optimal allocation of health care resources.* Decision Sciences, 2020, vol. 52:5

[35] Rawls, J. *A Theory of Justice.* Cambridge, MA: Harvard University Press. 1971

[36] Nease, R. F., Shachter, R. D., Owens, D. K. *Use of influence diagrams to structure medical decisions*, Medical Decision Making, 1997, vol. 17:3, pp. 241–262.

[37] Gustafsson, J., Salo, A. *Contingent portfolio programming for the management of risky projects.* Operations Research, 2005, vol.53, pp. 946-956.