# Efficient Calculation of Optimal Decisions in Graphical Models

Marie Lilleborge

Master of Science in Physics and Mathematics
Submission date: June 2012
Supervisor: Håkon Tjelmeland, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

## Preface

This project report is my Master's thesis, and represents 20 weeks of work. Writing this thesis is the last part of the Master of Science in Industrial Mathematics program at the Norwegian University of Science and Technology. The first years of studies, I tried not to think about the fact that I would have to write a Master's thesis during my tenth and last semester. I have loved being a student, and I have very much enjoyed each and every one of the last five years. Especially, I have loved this Spring semester. Working with this project and this thesis, it turned out to be maybe the best semester of all. I have learned a lot, both of statistical knowledge, but also a lot about writing a report.

First and foremost, I would like to thank my adviser Professor Håkon Tjelmeland for the invaluable cooperation, for each of our many discussions, and for lots of inspiration and constructive feedback. I am very thankful for all the time he has spent on this project.

I would also like to thank "Matteland", and with that, all of my classmates, for creating a great place to work, but also space and time for much needed breaks. Then, I would like to thank the Norwegian Computing Center for hopes and dreams for the future. My mother very much deserves a special thanks for her continuous support, as does my grandmother for being the most original person I know. Also, a big thanks to my dad, for telling his 16 year old daughter that girls are not able to do Mathematics.

At last, I would like to thank Black Sabbath, longboarding, Thin Lizzy, cross country skiing, coffee, unicycling, Led Zeppelin, Rubik's cube and the wonderful spring in Trondheim for making each of my days special.

Trondheim, June 2012
Marie Lilleborge

## Abstract

We present a method for finding the optimal decision on Random Variables in a graphical model. Upper and lower bounds on the exact value for each decision are used to reduce the complexity of the algorithm, while we still ensure that the decision chosen actually represents the exact optimal choice. Since the highest lower bound value is also a lower bound on the value of the optimal decision, we rule out any candidate with an upper bound of lower value than the highest lower bound. By this strategy, we try to reduce the number of candidates to a number we can afford to do exact calculations on.

We generate five Bayesian Networks with corresponding value functions, and apply our strategy to these. The bounds on the values are obtained by use of an available computer program, where the complexity is controlled by an input constant. We study the number of decisions accepted for different values of this input constant. From the first Network, we learn that the bounds does not work well unless we split the calculations into parts for different groups of the nodes. We observe that this splitting works well on the next three Networks, while the last Network illustrates how the method fails when we add more edges to the graph. We realize that our improved strategy is successful on sparse graphs, while the method is unsuccessful when we increase the density of edges among the nodes.

**Sammendrag**

Vi presenterer en metode for å finne optimal beslutning på tilfeldige variable i grafiske modeller. Øvre og nedre skranker for eksakt verdi blir brukt for å redusere metodens kompleksitet, samtidig som det sørger for at beslutningen som velges til slutt faktisk er den optimale. Da den høyeste nedre skranken også er en nedre skranke for verdien av den optimale beslutningen, kan vi forkaste enhver kandidat med lavere øvre skranke enn den høyeste nedre skranken. Slik prøver vi å redusere antallet kandidater til et antall vi klarer å regne eksakt på.

Vi genererer fem Bayesianske nettverk med tilhørende verdifunksjoner, og tester metoden på disse. Skrankene blir funnet ved hjelp av et tilgjengelig dataprogram, hvor kompleksiteten styres av brukeren. Vi har variert denne, og sett på hvor mange kandidater vi klarer å forkaste. Av det første nettverket ser vi at skrankene fungerer dårlig dersom vi ikke splitter utregningene i deler for ulike grupper av nodene. Fra neste tre nettverkene, ser vi at oppsplittingen fungerer godt, samtidig som det siste nettverket viser hvordan metoden slutter å fungere når vi legger til flere kanter på grafen. Vi ser derfor at resultatet blir bra for grafer med relativt få koblinger mellom nodene, men når tettheten av disse øker, ser vi at metoden slutter å fungere.

# Contents

# 1 Introduction

Continuously, we try our best to find the best decision for something. Usually, we want do "the right thing", but even with the best intentions, our decisions can lead to outcomes we don't like. This is because we usually don't control all surroundings that will influence the result of our decision, when set into action. That means that we don't know for sure what will be the end result of acting according to some decision. Then, we want to reason about how good a decision is on average. To handle this problem, we set up a probabilistic model for the surroundings, and associate Random Variables to the sources of influence to our end result. On these Random Variables, we construct a Joint Probability Distribution.

The Joint Probability Distribution associates a probability to each event concerning the surroundings, that is, some number telling how often we would expect this collection of criteria on the surroundings to be true. But the Joint Probability Distribution also contains more information about how the Random Variables depend on each other in a probabilistic setting. So, when setting up such a probabilistic model, we have to be able to encode all such information. In general, a causal model is both the easiest to set up, and the easiest to work with after. That means setting up a marginal probability distribution for each Random Variable to depend on the other Random Variables having a direct influence on the first one, and merging all these marginals to the Joint Probability Distribution.

To visualize these dependencies, it is usually convenient to set it all up in a graphical model. That means drawing a circle to represent each Random Variable, and put arrows between them to encode dependencies. In a directed graphical model, the arrows have a direction, and correspondingly, in an undirected graphical model, arrows are undirected. For the directed causal model, we would set an arrow to point from a Random Variable to another if the first one has direct influence on the other. After specifying the graphical model, that is, the circles and arrows, the graphical model encodes some structure on the Joint Probability Distribution. This structure usually also makes it easier to specify the numbers we need to have the full Joint Probability Distribution.

Given the graphical probabilistic model, we can start reasoning about the effect of our decision; what could happen, and how often it would. To figure out what is actually the best decision, we have to be able to associate a value to each realization. That is, a combination of our decision and an assignment to the unknown surroundings should correspond to some number representing the value. These values are then a function depending both on our deterministic decision and several unknown Random Variables. And then, with our probabilistic graphical model in hand, we can find the expected value of a decision. Comparing the expected values for different decisions, we choose the decision with the highest expected value, and call this the optimal decision.

As an example of application, we have a search for hydrocarbons, as in Martinelli et al. (2011). The Random Variables represent different places where hydrocarbons might be present. The structure of the graph is specified by where hydrocarbons can be generated, and where it can flow. That is, the probabilistic model and the Joint Probability Distribution is set up by looking at some geological data. We would like to expect as high a gross income as possible, and our decision would be which areas to check. The value function would represent the gross income of a realization, that is, the sum over incomes for findings, minus both development costs for the findings and costs for checking all of the areas we decided to.

The general problem described above is exactly what we try to solve in this report. First, in Section 2, we introduce theory about such probabilistic models, where we focus on Bayesian Networks, which is a type of directed graphical models. In Section 3, we present the representation of functions that will be used throughout the report, and also, in Section 4, we present an algorithm that will be used when we calculate the expected value. Then, we give a mathematical formulation of our problem in Section 5, and a mathematical formulation of the solution is given in Section 6. Sections 7, 8 and 9 present information and the actual results from tests of our solution method, and Section 10 discusses the complexity of the algorithms. Finally, Section 11 provides some closing remarks.

## 2 Bayesian Networks

From Cormen et al. (2009), we have the following definition of a directed graph.

**Definition 2.1.** *A directed graph $G$ is a pair $(V, E)$, where $V$ is a finite set and $E$ is a binary relation on $V$. The set $V$ is called the vertex set of $G$, and its elements are called vertices or nodes. The set $E$ is called the edge set of $G$, and its elements are called edges.*

We will refer to the elements in $V$ as the nodes in $G$, and represent them as circles in our figures. As described in Definition 2.1, the elements in $E$ are ordered pairs of nodes. That is, if $e = (X_i, X_j) \in E$, there is an edge $e$ from node $X_i$ to node $X_j$. This edge $e$ will be represented by an arrow from node $X_i$ to node $X_j$ in our figures. Correspondingly, an undirected graph has

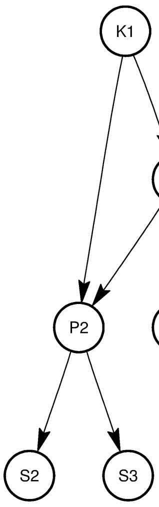

Figure 1: An example graph $G = (V, E)$. All graphs in this report are drawn by use of the MATLAB Biograph tool.

undirected edges. That is, the elements in $E$ are unordered pairs of nodes, and can therefore be viewed as if there was a directed edge in both directions. From now on, when the term graph is used, we assume that the graph is a directed one, unless otherwise specified.

A visualization of an example graph can be seen in Figure 1. Observe that in this graph,

$$V = \{\ K_1,\ P_1,\ P_2,\ S_1,\ S_2,\ S_3\ \}$$

and

$$E = \{\ (K_1, P_1),\ (K_1, P_2),\ (P_1, P_2),\ (P_1, S_1),\ (P_2, S_2),\ (P_2, S_3)\ \}.$$

A Bayesian Network is a way to express conditional independence assumptions among a set of Random Variables by use of a graph. Each Random Variable is represented as a node $X_i$, and arrows between the nodes encode a possible conditional dependence relationship. In this discussion, we will not distinguish between the Random Variable and its corresponding node. In graph theory the terminology for nodes at the different end points of an edge is as follows.

**Definition 2.2.** *If there is an edge $e = (X_j, X_k)$ from node $X_j$ to node $X_k$, node $X_j$ is a parent of node $X_k$, and node $X_k$ is a child of node $X_j$.*

If there are no edges $e$ from any node to node $X_j$, we say that $X_j$ is a root node. If there are no edges $e$ to any node from node $X_j$, we say that $X_j$ is a leaf node. Also, a path is a list of edges, such that the end node at each edge is the start node of the next. Then, the path describes a way to traverse the graph from the first start node to the last end node by just following the directed edges in the graph. If there is a path along the directed edges $E = \{e_i\}_i$ from node $X_j$ to node $X_k$, we say that $X_j$ is an ancestor of $X_k$, and $X_k$ is an descendant of $X_j$. Letting $2^V$ denote the power set of $V$, this introduces the following functions.

$$\begin{aligned} \mathrm{Pa} &: V \to 2^V \qquad \text{such that} \qquad \mathrm{Pa}(X_i) = \{X_k \in V \mid (X_k, X_i) \in E\} \\ \mathrm{Ch} &: V \to 2^V \qquad \text{such that} \qquad \mathrm{Ch}(X_i) = \{X_k \in V \mid (X_i, X_k) \in E\} \end{aligned}$$

That is, $\mathrm{Pa}(X_i)$ is the set of nodes $X_k$ such that $X_k$ is a parent of $X_i$ and $\mathrm{Ch}(X_i)$ is the set of nodes $X_k$ such that $X_k$ is a child of $X_i$. Correspondingly, we also let $\mathrm{Anc}(X_i)$ denote the set of ancestors of $X_i$ and $\mathrm{Des}(X_i)$ denote the set of descendants of $X_i$.

With some graph terminology in hand, we present the following definition of a Bayesian Network from Russell and Norvig (2003).

**Definition 2.3.** *A Bayesian Network is a graph, consisting of a set of nodes $V = \{X_i\}_{i=1}^n$ and a set of directed edges $E = \{e_i\}_{i=1}^{n_e}$ between pairs of the nodes. It is required that the graph has no directed cycles, i.e. it is a Directed Acyclic Graph. In addition, each node $X_i$ has a set of Local Probability Distributions $P(X_i|Pa(X_i))$ associated with it.*

That is, for each assignment of $\mathrm{Pa}(X_i)$, $P(X_i|\mathrm{Pa}(X_i))$ is a probability distribution for the Random Variable $X_i$.

The nodes and the edges specify what is called the topology of the network. Since the topology of the Bayesian Network constitutes a Directed Acyclic Graph, there is a topological ordering of the nodes. That is, there exists a bijective numbering of the nodes

$$\ell \; : \; \{X_i\}_{i=1}^n \;\to\; \{1, \cdots, n\}$$

such that for any edge $e = (X_j, X_k)$ in the network, we have $\ell(X_j) < \ell(X_k)$. Also, this means that for any nodes $X_j, X_k$ with $\ell(X_j) < \ell(X_k)$, there is no directed path from $X_k$ to $X_j$.

Given a Bayesian Network with its topology and the sets of Local Probability Distributions, one uniquely determines the full Joint Probability Distribution over all the Random Variables represented in the network by the formula

$$\mathbb{P}(X_1, \cdots, X_n) = \prod_{i=1}^{n} P\left(X_i | \mathrm{Pa}(X_i)\right). \tag{2.1}$$

## 2.1 Mathematical Results

Notice that, given a subset $A_j$ of the Random Variables, we can divide the probability distribution into

$$\mathbb{P}(X_1, \cdots, X_n) = \prod_{X_k \in A_j} P\left(X_k | \mathrm{Pa}(X_k)\right) \prod_{X_k \notin A_j} P\left(X_k | \mathrm{Pa}(X_k)\right).$$

Note that we have introduced the short hand $P(X_j|\mathrm{Pa}(X_j))$ for the more correct expression $P(X_j = x_j | X_k = x_k \ \forall X_k \in \mathrm{Pa}(X_j))$. Given a Random Variable $X_j$, we let $C_j \subseteq \mathrm{Ch}(X_j)$, and extend the definition of the function Anc to sets $C_j$ of nodes such that

$$\mathrm{Anc}(C_j) = \bigcup_{X_k \in C_j} \mathrm{Anc}(X_k).$$

Now, let

$$A_j \ = \ C_j \cup \mathrm{Anc}(C_j) \cup \mathrm{Anc}(X_j) \cup \{X_j\}, \tag{2.2}$$

and also

$$A_j^c = \{X_1, \cdots, X_n\} \backslash A_j.$$

Notice that if $C_j \neq \varnothing$,

$$\mathrm{Anc}(X_j) \cup \{X_j\} \subseteq \mathrm{Anc}(C_j).$$

We are going to prove that for any assignment of the Random Variables in $A_j$,

$$\sum_{X_i \in A_j^c} \prod_{X_k \in A_j^c} P\left(X_k | \mathrm{Pa}(X_k)\right) = 1. \tag{2.3}$$

Without loss of generality, assume that the numbering $i$ of the nodes $X_i$ is such that

$$A_j = \{X_1, X_2, \cdots, X_{m-1}\}, \qquad A_j^c = \{X_m, X_{m+1}, \cdots, X_n\},$$

and that $X_1, X_2, \cdots, X_n$ is actually a topological ordering of all the Random Variables $\{X_1, \cdots, X_n\}$ in the Bayesian Network. Then, given an assignment of the Random Variables in $A_j$,

$$\begin{aligned}
\sum_{X_i \in A_j^c} \prod_{X_k \in A_j^c} P\left(X_k | \mathrm{Pa}(X_k)\right) &= \sum_{X_m} \cdots \sum_{X_{n-1}} \sum_{X_n} \prod_{X_k \in A_j^c} P\left(X_k | \mathrm{Pa}(X_k)\right) \\
&= \sum_{X_m} \cdots \sum_{X_{n-1}} \sum_{X_n} \prod_{k=m}^{n} P\left(X_k | \mathrm{Pa}(X_k)\right) \\
&= \sum_{X_m} \cdots \sum_{X_{n-1}} \prod_{k=m}^{n-1} P\left(X_k | \mathrm{Pa}(X_k)\right) \sum_{X_n} P\left(X_n | \mathrm{Pa}(X_n)\right) \\
&= \sum_{X_m} \cdots \sum_{X_{n-1}} \prod_{k=m}^{n-1} P\left(X_k | \mathrm{Pa}(X_k)\right) \\
&\cdots \\
&= \sum_{X_m} P\left(X_m | \mathrm{Pa}(X_m)\right) \\
&= 1,
\end{aligned}$$

since for each $k$, and any assignment of the Random Variables in $\mathrm{Pa}(X_k)$, $P(X_k|\mathrm{Pa}(X_k))$ is a probability distribution for the Random Variable $X_k$, and thus sums to one. The fact that $P(X_k|\mathrm{Pa}(X_k))$ does not depend on any variable $X_i$ for which $X_i$ is succeding $X_k$ in a topological ordering, is used to move products outside the innermost sum. That is, in a topological ordering, a given node is succeeding all of its ancestors. The Local Probability Distribution $P(X_k|\mathrm{Pa}(X_k))$, varies only with the different assignments to $X_k$'s parents, which is a subset of $X_k$'s ancestors. Therefore, this Local Probability Distribution cannot depend on a node $X_i$ succeeding the node $X_k$ in the topological ordering, in this setting denoted by $k < i$.

By Bayes Rule,

$$\mathbb{P}(X_j|\mathrm{Pa}(X_j)) = \frac{\mathbb{P}(X_j, \mathrm{Pa}(X_j))}{\mathbb{P}(\mathrm{Pa}(X_j))}. \tag{2.4}$$

Let

$$C_j = \varnothing,$$

and note that this implies that

$$A_j = \mathrm{Anc}(X_j) \cup \{X_j\}.$$

Define

$$\begin{aligned}\tilde{A}_j &= A_j \backslash \mathrm{Pa}(X_j) \backslash \{X_j\} \\ &= (\mathrm{Anc}(X_j) \cup \{X_j\}) \backslash (\mathrm{Pa}(X_j) \cup \{X_j\}) \\ &= \mathrm{Anc}(X_j) \backslash \mathrm{Pa}(X_j).\end{aligned}$$

For any assignment of the Random Variables in $\mathrm{Pa}(X_j)$, the denominator in (2.4) can be expressed as

$$\begin{aligned}\mathbb{P}(\mathrm{Pa}(X_j)) &= \sum_{X_i \in (\mathrm{Pa}(X_j))^c} \mathbb{P}(X_1, \cdots, X_n) \\ &= \sum_{X_{i_2} \in \tilde{A}_j \cup \{X_j\}} \sum_{X_{i_1} \in A_j^c} \prod_{X_k \in A_j} P(X_k|\mathrm{Pa}(X_k)) \prod_{X_k \in A_j^c} P(X_k|\mathrm{Pa}(X_k)) \\ &= \sum_{X_{i_2} \in \tilde{A}_j \cup \{X_j\}} \prod_{X_k \in A_j} P(X_k|\mathrm{Pa}(X_k)) \left( \sum_{X_{i_1} \in A_j^c} \prod_{X_k \in A_j^c} P(X_k|\mathrm{Pa}(X_k)) \right) \\ &= \sum_{X_i \in \tilde{A}_j \cup \{X_j\}} \prod_{X_k \in A_j} P(X_k|\mathrm{Pa}(X_k)) \\ &= \sum_{X_i \in \tilde{A}_j} \sum_{X_j} P(X_j|\mathrm{Pa}(X_j)) \prod_{X_k \in \mathrm{Anc}(X_j)} P(X_k|\mathrm{Pa}(X_k)) \\ &= \sum_{X_i \in \tilde{A}_j} \prod_{X_k \in \mathrm{Anc}(X_j)} P(X_k|\mathrm{Pa}(X_k)) \left( \sum_{X_j} P(X_j|\mathrm{Pa}(X_j)) \right) \\ &= \sum_{X_i \in \tilde{A}_j} \prod_{X_k \in \mathrm{Anc}(X_j)} P(X_k|\mathrm{Pa}(X_k)),\end{aligned}$$

where we have used (2.3) for $C_j = \varnothing$. Correspondingly, for any assignment of the Random Variables in $\mathrm{Pa}(X_j) \cup \{X_j\}$,

$$\begin{aligned}\mathbb{P}(X_j, \mathrm{Pa}(X_j)) &= \sum_{X_i \in (\{X_j\} \cup \mathrm{Pa}(X_j))^c} \mathbb{P}(X_1, \cdots, X_n) \\ &= \sum_{X_{i_2} \in \tilde{A}_j} \sum_{X_{i_1} \in A_j^c} \prod_{X_k \in A_j} P(X_k|\mathrm{Pa}(X_k)) \prod_{X_k \in A_j^c} P(X_k|\mathrm{Pa}(X_k)) \\ &= \sum_{X_{i_2} \in \tilde{A}_j} \prod_{X_k \in A_j} P(X_k|\mathrm{Pa}(X_k)) \left( \sum_{X_{i_1} \in A_j^c} \prod_{X_k \in A_j^c} P(X_k|\mathrm{Pa}(X_k)) \right)\end{aligned}$$

$$\begin{aligned}
&= \sum_{X_i \in \tilde{A}_j} \prod_{X_k \in A_j} P(X_k|\mathrm{Pa}(X_k)) \\
&= \sum_{X_i \in \tilde{A}_j} P(X_j|\mathrm{Pa}(X_j)) \prod_{X_k \in \mathrm{Anc}(X_j)} P(X_k|\mathrm{Pa}(X_k)) \\
&= P(X_j|\mathrm{Pa}(X_j)) \sum_{X_i \in \tilde{A}_j} \prod_{X_k \in \mathrm{Anc}(X_j)} P(X_k|\mathrm{Pa}(X_k)) \\
&= P(X_j|\mathrm{Pa}(X_j)) \cdot \mathbb{P}(\mathrm{Pa}(X_j)).
\end{aligned}$$

That is, the Local Probability Distributions $P(X_k|\mathrm{Pa}(X_k))$ equals the Conditional Probability Distributions $\mathbb{P}(X_k|\mathrm{Pa}(X_k))$, since (2.4) reduces to

$$\mathbb{P}(X_j|\mathrm{Pa}(X_j)) = \frac{\mathbb{P}(X_j, \mathrm{Pa}(X_j))}{\mathbb{P}(\mathrm{Pa}(X_j))} = P(X_k|\mathrm{Pa}(X_k)).$$

Thus, we can write (2.1) as

$$\mathbb{P}(X_1, \cdots, X_n) = \prod_{i=1}^{n} \mathbb{P}(X_i|\mathrm{Pa}(X_i)). \tag{2.5}$$

Then again, let $A_j$ be any set as in (2.2), and $h(X_1, \cdots, X_n)$ be a function whose value only depends on the assignment of the variables in $A_j$. We know that

$$\begin{aligned}
\mathbb{E}h &= \sum_{\text{All RVs}} h(X_1, \cdots, X_n)\mathbb{P}(X_1, \cdots, X_n) \\
&= \sum_{X_{i_1} \in A_j} \sum_{X_{i_2} \in A_j^c} h(X_1, \cdots, X_n)\mathbb{P}(X_1, \cdots, X_n) \\
&= \sum_{X_{i_1} \in A_j} \sum_{X_{i_2} \in A_j^c} h(X_1, \cdots, X_n) \prod_{X_k \in A_j} \mathbb{P}(X_k|\mathrm{Pa}(X_k)) \prod_{X_k \in A_j^c} \mathbb{P}(X_k|\mathrm{Pa}(X_k)).
\end{aligned}$$

Since $h$ only depends on variables in $A_j$, we could write $h = h(X_{A_j})$, and get

$$\begin{aligned}
\mathbb{E}h &= \sum_{X_{i_1} \in A_j} \sum_{X_{i_2} \in A_j^c} h(X_{A_j}) \prod_{X_k \in A_j} \mathbb{P}(X_k|\mathrm{Pa}(X_k)) \prod_{X_k \in A_j^c} \mathbb{P}(X_k|\mathrm{Pa}(X_k)) \\
&= \sum_{X_{i_1} \in A_j} h(X_{A_j}) \prod_{X_k \in A_j} \mathbb{P}(X_k|\mathrm{Pa}(X_k)) \left( \sum_{X_{i_2} \in A_j^c} \prod_{X_k \in A_j^c} \mathbb{P}(X_k|\mathrm{Pa}(X_k)) \right) \\
&= \sum_{X_{i_1} \in A_j} h(X_{A_j}) \prod_{X_k \in A_j} \mathbb{P}(X_k|\mathrm{Pa}(X_k)). \qquad (2.6)
\end{aligned}$$

Actually,

$$\prod_{X_k \in A_j} \mathbb{P}(X_k|\text{Pa}(X_k)) = \mathbb{P}(X_{A_j}),$$

which, in fact, is easily proved by (2.6) by letting $h(X_1, \cdots, X_n)$ be the indicator function for some assignment of the Random Variables in $A_j$. However, the main observation from (2.6), is that to find the expected value $\mathbb{E}h$, it is sufficient to use the probability distribution we get by ignoring all the factors in (2.5) which are Conditional Probability Distributions for the Random Variables in $A_j^c$. That is, letting $\mathbb{P}(X_{A_j})$ denote the product in the above equation, we get

$$\mathbb{E}h(X_{A_j}) = \sum_{A_j} h(X_{A_j})\mathbb{P}(X_{A_j}). \tag{2.7}$$

## 2.2 Conditional Independence

As mentioned, each edge in the graph encodes a possible conditional dependence relationship between two nodes. That is, the edges in the graph determines the factors in the formula (2.5), which in turn allows for certain conditional dependence relationships. Whether a set of variables actually are conditionally dependent of each other, is determined by the conditional probability distributions $\{\mathbb{P}(X_i|\text{Pa}(X_i))\}$.

We let $X \perp Y$ denote that two Random Variables $X, Y$ are independent, i.e.

$$\mathbb{P}(X = x) = \mathbb{P}(X = x|Y = y) \qquad \forall x, y.$$

An equivalent statement is

$$\mathbb{P}(X = x, Y = y) = \mathbb{P}(X = x) \cdot \mathbb{P}(Y = y) \qquad \forall x, y.$$

Also, the symbol | indicates that we are discussing a conditional probability distribution, that is, X $\perp$ Y | Z means

$$\mathbb{P}(X = x, Y = y|Z = z) = \mathbb{P}(X = x|Z = z) \cdot \mathbb{P}(Y = y|Z = z) \qquad \forall x, y, z.$$

Note that this means that $X \perp Y$ and $X \perp Y|\ \varnothing$ are equivalent statements.

Correspondingly, we let $X \not\perp Y$ denote that the graph does not encode that the Random Variables $X, Y$ are independent. Also, $X \not\perp Y|\ Z$ means that the graph does not encode that $X$ and $Y$ are independent conditional on $Z$. That does not mean, however, that there are no set of conditional probability distributions that makes them independent. In fact, if we let $X$ or $Y$ be a constant, they will always be conditionally independent, but their joint distribution can still be represented on a graph where they are connected by edges. The edges just allow for Random Variables not to be independent.

As an example, we study the graph in Figure 1. There are algorithms for finding all conditional independence relationships, see for example the Bouncing Ball Algorithm in Jordan (n.d.). However, this will not be covered in our scope. To introduce some intuition, we will present some of the possible conditional independence relationships from the graph in Figure 1, and give a brief, non rigorous discussion of what they mean for a given example.

We could look at the nodes in Figure 1 as persons, and let the edges indicate where messages could be sent. Say, $K_1$ suddenly wants to have a dinner party, and on his phone he only has the phone numbers to $P_1$ and $P_2$. Correspondingly, $P_1$ can only communicate a message to $P_2$ and $S_1$, and $P_2$ to $S_2$ and $S_3$. Say, $K_1$ would want all to come, but he can only say this to $P_1$ and $P_2$. However, through them all persons can be reached, but $P_1$ and $P_2$ would only forward the message if they are going themselves. The Random Variable for each node would here be a Boolean variable $\{\textbf{True}, \textbf{False}\}$ indicating whether the person is attending the dinner party or not. If $K_1$ is having the party for sure, he would be a constant Random Variable, having value **True** with probability 1. Otherwise, his node could be associated with some probability distribution indicating a probability for him inviting to a dinner party or not. In that case, assuming all telephones are working perfectly, and all persons are going if they get the message, we know that all Random Variables are either **True** or all Random Variables are **False**. Thus, we cannot have

$$S_1 \perp S_2,$$

$$K_1 \perp S_1$$

nor

$$S_2 \perp S_3.$$

However, if we know the value of $P_2$, then $S_1$ and $S_2$ are independent, written

$$S_2 \perp S_3 \mid P_2.$$

Then, allow for the telephones not to be perfect, that is, not all messages sent will be received. For each message sent, whether it is received or not is assumed to be independent of whether any other sent message is received. For example, the statement

$$S_1 \perp S_2 \mid P_2$$

is explained by the fact that knowing whether $P_2$ is going or not leaves only what we non rigorously can call independent randomness for the impact on the values of $S_1$ and $S_2$, respectively.

Note that any $\not\perp$ statement can be proved by a counter example, while the explanations for the $\perp$ statements above just serves as describing examples. The above examples can be summarized in the following list of conditional independence statements for the graph in Figure 1.

- $K_1 \not\perp S_1$
- $K_1 \perp S_1 \mid P_1$
- $S_2 \not\perp S_3$
- $S_2 \perp S_3 \mid P_2$
- $S_1 \not\perp S_2$
- $S_1 \perp S_2 \mid P_1$
- $S_1 \perp S_2 \mid P_2$
- $S_1 \perp S_2 \mid P_1, P_2$
- $S_1 \not\perp S_2 \mid K_1$

Note that this is just a small excerpt of all the possible conditional independence statements we could deduce from Figure 1. To find more of them, or for a more rigorous check, the reader is referred to either the Bouncing Ball Algorithm in Jordan (n.d.), or to do the more tedious work of checking by use of Bayes Rule on the general Joint Probability Distribution (2.5) for the given graph.

In general, there are two standard conditional independence relations that are characteristic for all Bayesian Networks. They are to be found in the following two Theorems from Russell and Norvig (2003), and assume a Joint Probability Distribution where each assignment of the Random Variables $X_1, \cdots, X_n$ has a positive probability,

$$\mathbb{P}(X_1, \cdots, X_n) > 0.$$

**Theorem 2.1.** *A Random Variable in a Bayesian Network is conditionally independent of its non-descendants, given its parents.*

To state the last theorem, we have to make the following definition as in Russell and Norvig (2003).

**Definition 2.4.** *The Markov Blanket of a Node in a Bayesian Network is defined to be the set of its neighboring nodes, that is, its parents, its children and the parents of its children.*

**Theorem 2.2.** *A Random Variable in a Bayesian Network is conditionally independent of all other nodes in the network, given its Markov Blanket, as in Definition 2.4*

Theorems 2.1 and 2.2 can be proved by applying Bayes rule on the Joint Probability Distribution as in (2.5).

## 2.3 Markov Random Fields

In a Bayesian Network, a Directed Acyclic Graph is used to represent some dependence properties of the Joint Probability Distribution for the Random Variables. But the same Joint Probability Distribution (2.5) can also be represented on an undirected graph. A Markov Random Field is an undirected graphical model to represent probabilistic relationships. That is, as for a Bayesian Network, each node still represents a Random Variable, but in a Markov Random Field the edges are undirected. Thus, we cannot talk about a node's parents nor its children, descendants or ancestors. Instead we have the following definition from Bishop (2006).

**Definition 2.5.** *A clique is a subset of the nodes in the undirected graph such that there exists an (undirected) edge between all pairs of nodes in the subset. Furthermore, a maximal clique is a clique for which it is not possible to include any other nodes in the graph without it ceasing to be a clique.*

A potential function $\psi$ on a maximal clique is a non negative function on the Random Variables in the clique. By letting $\Lambda$ denote a maximal clique, and $X_\Lambda$ denote the Random Variables in it, we can define a probability distribution on an undirected graph as the product of all potential functions $\psi_\Lambda(X_\Lambda)$, as in

$$\mathbb{P}(X_1, \cdots X_n) = \frac{1}{C} \prod_{\Lambda} \psi_\Lambda(X_\Lambda), \tag{2.8}$$

where $C$ is the normalization constant that ensures it to be a probability distribution. The undirected graph and the corresponding probability distribution is then a Markov Random Field.

As for Bayesian Networks, there are also conditional independence statements that are true in general for Markov Random Fields. The following is from Bishop (2006), and assumes a Joint Probability Distribution where

$$\mathbb{P}(X_1, \cdots, X_n) > 0$$

for any assignment to the Random Variables $X_1, \cdots, X_n$.

**Theorem 2.3.** *For three sets of nodes $A, B, D$, we have*

$$A \perp B \mid D$$

*if all possible (undirected) paths between a node in $A$ and a node in $B$ pass through one or more nodes in $D$.*

From this, we can deduce that a Random Variable is conditionally independent of any subset of the other Random Variables given its neighbors. Thus, the Markov Blanket of a Random Variable $X_i$ in a Markov Random Field, is the set of nodes $X_j$ which shares an edge $e = \{X_i, X_j\} \in E$ with $X_i$.

Given a Bayesian Network, we would like to find a corresponding Markov Random Field. For the same conditional independence properties to hold, we need the Markov Blanket for each Random Variable to be the same in both representations. Hence, when going from the Bayesian Network representation to the Markov Random Field representation, edges must be added between any two nodes that have common children. In fact, according to Bishop (2006), in addition to replacing all directed edges in the original Bayesian Network by undirected ones, this is all we need to do. The process of adding the extra edges between any pair of parents of the same child is called "moralization", and is also referred to as "marrying the parents". The resulting undirected graph is called the moral graph, and is also the corresponding undirected graphical representation of the probability distribution.

For the Joint Probability Distribution, note from (2.5) and (2.8), that we need to have

$$\frac{1}{C} \prod_{\Lambda} \psi_\Lambda(X_\Lambda) = \prod_{i=1}^{n} \mathbb{P}(X_i | \text{Pa}(X_i)). \tag{2.9}$$

As an example, we study the directed graph in Figure 1. Note that no moralization has to be done, since $K_1$ and $P_1$ are the only nodes with common children, and they already share an edge. Thus, the moral graph of Figure 1 has the same nodes and the same edges, except that now the edges are undirected ones. For this graph, the Bayesian Network tells us that

$$\mathbb{P}(K_1, \cdots, S_3) = \mathbb{P}(K_1)\mathbb{P}(P_1|K_1)\mathbb{P}(S_1|P_1)\mathbb{P}(P_2|K_1, P_1)\mathbb{P}(S_2|P_2)\mathbb{P}(S_3|P_2).$$

The corresponding moral graph tells us that

$$\mathbb{P}(K_1, \cdots, S_3) = \frac{1}{C}\psi_1(P_1, S_1)\psi_2(P_2, S_2)\psi_3(P_2, S_3)\psi_4(K_1, P_1, P_2).$$

Comparing the two above equations, we observe from (2.9) that we can choose

$$\begin{aligned}
\psi_1(P_1, S_1) &= C_1\mathbb{P}(S_1|P_1)\\
\psi_2(P_2, S_2) &= C_2\mathbb{P}(S_2|P_2)\\
\psi_3(P_2, S_3) &= C_3\mathbb{P}(S_3|P_2)\\
\psi_4(K_1, P_1, P_2) &= C_4\mathbb{P}(K_1)\mathbb{P}(P_1|K_1)\mathbb{P}(P_2|K_1, P_1)
\end{aligned}$$

for some constants $C_1, C_2, C_3, C_4$ such that $C = \frac{1}{C_1C_2C_3C_4}$. The general case is similar, and we can always choose all $C_i$s to be 1, and thus get $C = 1$. From now on, we will assume this choice.

In general, note that moralization ensures that for any node $X_i$, there exists a maximal clique $\Lambda$ in the moral graph such that

$$\{X_i\} \cup \mathrm{Pa}(X_i) \subseteq \Lambda.$$

Thus, we observe from (2.9), that for any maximal clique $\Lambda$, there is a non empty subset $\lambda \subseteq \Lambda$ such that

$$\psi_\Lambda = \prod_{X_i \in \lambda} \mathbb{P}(X_i|\mathrm{Pa}(X_i)).$$

Now, let us study a subset of the Random Variables in Figure 1. Let $\tilde{G} = (\tilde{V}, \tilde{E})$, where

$$\tilde{V} = \{\ P_2,\ S_2,\ S_3\ \} \subset V \quad \text{and} \quad \tilde{E} = \{\ (P_2, S_2),\ (P_2, S_3)\ \} \subset E$$

for $G = (V, E)$ in Figure 1. A visualization of $\tilde{G}$ can be found in Figure 2, and the corresponding moral graph can be found in Figure 3. Observe that the maximal cliques are

$$\Lambda_1 = \{\ S_2,\ P_2\ \} \quad \text{and} \quad \Lambda_2 = \{\ S_3,\ P_2\ \}.$$

Correspondingly, the potential functions must be of the form

$$\psi_1 = \Phi_1(P_2)\mathbb{P}(S_2|P_2) \quad \text{and} \quad \psi_2 = \Phi_2(P_2)\mathbb{P}(S_3|P_2),$$

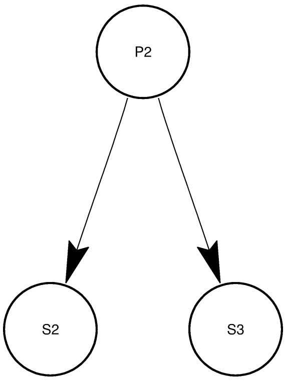

Figure 2: The graph $\tilde{G}$ created by a subset of the nodes in Figure 1.

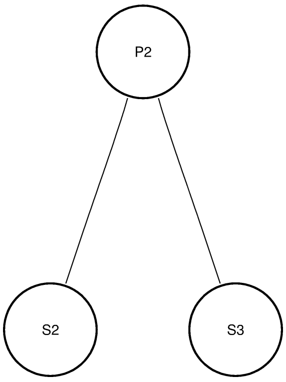

Figure 3: A visualization of the moral graph for $\tilde{G}$, where $\tilde{G}$ is the graph visualized in Figure 2. Note that no extra edges had to be added to obtain the moral graph.

where $\Phi_1(P_2)\Phi_2(P_2) = \mathbb{P}(S_2)$, and we can choose either $\Phi_1(P_2) \equiv 1$ or $\Phi_2(P_2) \equiv 1$.

From this example, we note that the choice of $\lambda$ might not be unique for each maximal clique $\Lambda$. In any case, the $\lambda$s define a partition of all the nodes $X_1, \cdots, X_n$. That is, the $\lambda$s are pairwise disjoint and their union contains all nodes in the graph.

## 3 Pseudo-Boolean functions

In several applications, we are working with a set of variables that can take on two values. In this section, we will present some theory about functions on such a set. As in Hammer and Rudeanu (1968), we begin by the following definition.

**Definition 3.1.** *The set $B_2 = \{0, 1\}$ together with the operations of disjunction, conjunction and negation is called the Boolean algebra.*

That is, the operations mentioned is defined as follows.

**Definition 3.2.** *Negation, denoted by $^-$, is associated with the function $1 - x$, and we get the following formulas*

$$\bar{0} = 1 \qquad \textit{and} \qquad \bar{1} = 0.$$

**Definition 3.3.** *Conjunction, denoted by $\cdot$, works like usual multiplication. That is*

$$0 \cdot 0 = 0 \qquad 0 \cdot 1 = 0 \qquad 1 \cdot 0 = 0 \qquad 1 \cdot 1 = 1.$$

**Definition 3.4.** *Boolean disjunction is denoted by $\cup$, and is defined by the formulas*

$$0 \cup 0 = 0 \qquad 0 \cup 1 = 1 \qquad 1 \cup 0 = 1 \qquad 1 \cup 1 = 1.$$

Also, we let the term Boolean variable denote a variable that can take values in the set $B_2$. A Boolean function $f$ is then a function

$$f : B_2^n \to B_2,$$

that is, a function whose value and $n$ arguments are all Boolean variables. This concept of a Boolean function is then expanded to the following.

**Definition 3.5.** *A pseudo-Boolean function $f$ is a function*

$$f : B_2^n \to \mathbb{R},$$

*that is, a function from an n-tuple of bivalent variables to the real numbers.*

Note that this definition allows us to look at the space of Boolean functions as a subspace of all pseudo-Boolean functions.

The following result is from Hammer and Rudeanu (1968).

**Theorem 3.1.** *Every pseudo-Boolean function may be written as a polynomial, which is linear in each variable, and which, after the reduction of the similar terms, is uniquely determined up to the order of the sums and products.*

That is, let $X = [X_k]_{k=1}^n$ be a vector of Boolean variables and $f$ be a pseudo-Boolean function. Then, there exists a collection $T$ of sets $\lambda \subseteq \{1, \cdots, n\}$ of indexes and a corresponding set of coefficients $\{\alpha^\lambda\}_{\lambda \in T}$ such that

$$f(X) = \sum_{\lambda \in T} \alpha^\lambda \prod_{k \in \lambda} X_k. \tag{3.1}$$

In fact, as mentioned in Theorem 3.1, there might be several choices of $T$. $T$ can always be the power set of $\{1, \cdots, n\}$, and in general

$$T \subseteq 2^{\{1, \cdots, n\}}.$$

As an example, we look at a function $h$ acting on the variables in Figure 1,

$$h(K_1, P_1, P_2, S_1, S_2, S_3) = \begin{cases} 1 & \text{if } S_2 = 0, S_3 = 0, \\ e & \text{if } S_2 = 1, S_3 = 0, \\ e^2 & \text{if } S_2 = 0, S_3 = 1, \\ e + e^2 - 1 & \text{if } S_2 = 1, S_3 = 1. \end{cases} \tag{3.2}$$

For this function, we can let

$$T = \{\ \varnothing,\ \{S_1\},\ \{S_2\}\ \},$$

and write $h$ as

$$h(K_1, P_1, P_2, S_1, S_2, S_3) = 1 + (e - 1)S_2 + (e^2 - 1)S_3.$$

Theorem 3.1 is easily proved by a calculation of the coefficients $\{\alpha^\lambda\}_{\lambda\in T}$ for a general function, which can be done recursively as follows. First, for simplicity, assume $T = 2^{\{1,\cdots,n\}}$, and let

$$v_\Lambda = [v_k]_{k=1}^n \qquad \text{such that} \qquad v_k = 1 \Leftrightarrow k \in \Lambda.$$

That is, we assume $T$ to be the power set of all indexes appearing in $S$, and $v_\Lambda$ to be a vector where all entries with indexes in $\Lambda$ are on, and all other entries are off. This allows us to set

$$\alpha^\varnothing = f(v_\varnothing) = f(\vec{0}),$$

and for increasing size of $\Lambda$, set

$$\alpha^\Lambda = f(v_\Lambda) - \sum_{\lambda \subsetneq \Lambda} \alpha^\lambda.$$

After determining the full set of coefficients $\{\alpha_\lambda\}$, the choice of $T$ can be changed by not including all $\lambda$s where $\alpha^\lambda = 0$.

In Hammer and Rudeanu (1968), several other representations of a pseudo-Boolean function, also using combinations of the operations in Definition 3.2 and Definition 3.4 are presented. In the following, we will assume the representation from (3.1).

From Theorem 3.1, we also deduce that any such positive valued function $f(X)$ can be represented as the exponential of a pseudo-Boolean function. That is,

$$f(X) = \exp\left(\sum_{\lambda\in S} \beta^\lambda \prod_{k\in\lambda} X_k\right). \tag{3.3}$$

As an example, the function $h$ in (3.2) can be written as

$$h(K_1, P_1, P_2, S_1, S_2, S_3) = \exp\left(S_2 + 2S_3 + S_2S_3(\ln(e + e^2 - 1) - 3)\right).$$

In the following, we will continue to write $\alpha^\lambda$ for coefficients of the pseudo-Boolean representation (3.1), and $\beta^\lambda$ for coefficients of the pseudo-Boolean representation of the logarithm of the given function as in (3.3). Correspondingly, $T$ will denote the subsets we are summing over in the pseudo-Boolean representation (3.1), and $S$ for the exponential version (3.3).

## 3.1 A graph structure for coefficients

When a computer program is given a pseudo-Boolean representation of a function, or when calculating the coefficients of that representation for a given function, a structure for storing coefficients of type $\alpha^\lambda$ is needed. Also, expressions of type

$$\sum_{\lambda \subsetneq \Lambda} \alpha^\lambda$$

appear frequently in the formulas, so given $\Lambda$, its coefficient and the coefficients of subsets of $\Lambda$ should be easily accessed. This is easily solved by a graph structure. The graph is a set of nodes, with one root node corresponding to $\alpha^\varnothing$, and all other nodes divided into layers depending on the size of $\Lambda$ for the coefficient $\alpha^\Lambda$ the node corresponds to. Let $N$ denote the set of indexes for the variables the given function $f$ depends on. For simplicity, assume $N = \{1, 2, \cdots, n\}$, and also that we study the full representation

$$f(X) = \sum_{\lambda \subseteq N} \alpha^\lambda \prod_{k \in \lambda} X_k.$$

Then, for each $\Lambda \subseteq N$, in the coefficient graph, the corresponding node has all subsets of $\Lambda$ as ancestors. Especially, the parents of that given node are the nodes corresponding to the sets in

$$\{\ \lambda \subset \Lambda\ :\ |\lambda|\ =\ |\Lambda| - 1\ \} = \{\ \lambda\ :\ \lambda = \Lambda \backslash \{m\},\ m \in \Lambda\ \}\,.$$

That is, to obtain a child of a given node $\lambda$, one augments $\lambda$ with an $m \in N \backslash \lambda$. Correspondingly, each parent is found by removing one of the elements in $\lambda$. Such a coefficient graph with $n = 4$ and four layers, can be seen in Figure 4.

As an example, given $\Lambda = \{3, 4, 7\}$ and $n = 7$, we observe

$$\mathrm{Pa}(\Lambda) = \{\ \{4, 7\},\ \{3, 7\},\ \{3, 4\}\ \}$$

and correspondingly,

$$\mathrm{Ch}(\Lambda) = \{\ \{1, 3, 4, 7\},\ \{2, 3, 4, 7\},\ \{3, 4, 5, 7\},\ \{3, 4, 6, 7\}\ \}\,.$$

An illustration of the node $\Lambda = \{3, 4, 7\}$ and its parents and children is to be found in Figure 5. We let layer $r$ denote the set of nodes of distance $r$ from the root node, so that all children of a given node are in the same layer, and similarly for all its parents.

For a further description of how we have implemented this graph structure, the reader is referred to Appendix A.

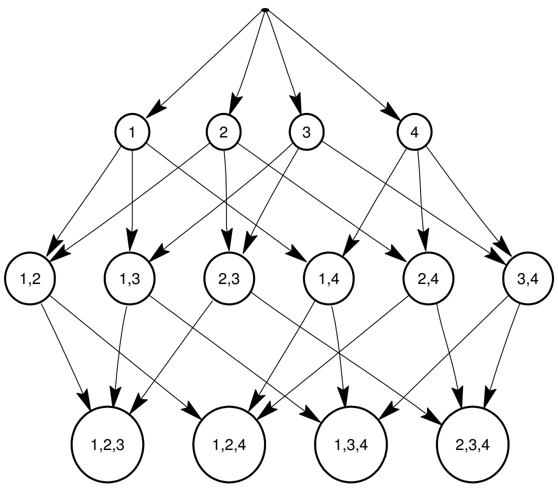

Figure 4: An example coefficient graph of four layers including the root node, with $n = 4$. That is, coefficient nodes corresponding to $|\lambda| \leqslant 3$ are represented. The arrows are from parents to children.

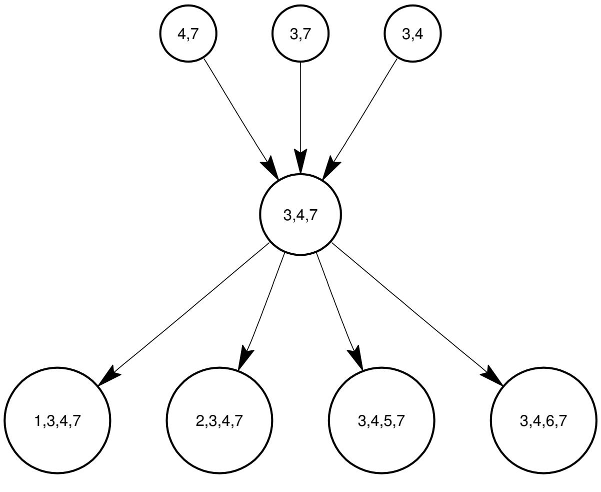

Figure 5: An illustration of the example showing parents and children for the node $\Lambda = \{3, 4, 7\}$ when $n = 7$.

## 3.2 Converting between exponential and linear form

Given a set $T$ of index sets $\lambda$, and the set coefficients $\{\alpha^\lambda\}_{\lambda\in T}$ to describe a function $f(X)$ of the form

$$f(X) = \sum_{\lambda\in T} \alpha^\lambda \prod_{k\in\lambda} X_k,$$

we want to find a corresponding set $S$, and the set of coefficients $\{\beta^\lambda\}_{\lambda\in S}$ to write $f(X)$ as

$$f(X) = \exp\left(\sum_{\lambda\in S} \beta^\lambda \prod_{k\in\lambda} X_k\right).$$

First, note that in general $S \neq T$. As an example, note for $a, b \in \mathbb{R}$, we get the following pair of equivalent representations

$$1 + aX_1 + bX_2 = \exp\left(X_1 \ln(1+a) + X_2 \ln(1+b) + X_1X_2 \ln\frac{1+a+b}{(1+a)(1+b)}\right).$$

That is, in general, the exponential representation is full even though the linear representation is sparse. Correspondingly, the linear representation might be full for a function with sparse exponential representation.

For simplicity, assume $S$ to be the power set of all indexes appearing in $T$, that is,

$$S = 2^{\{j \,:\, \exists\lambda\in T \text{ such that } j\in\lambda\}}.$$

We get, for $\Lambda \in S$,

$$\beta^\Lambda = \ln\left(\sum_{\lambda\in T \,:\, \lambda\subseteq\Lambda} \alpha^\lambda\right) - \sum_{\lambda\subsetneq\Lambda} \beta^\lambda.$$

This introduces an algorithm for calculating the $\beta^\lambda$s in order of increasing size of $\lambda$. Correspondingly, for the converse case, we have for $\Lambda \in T$

$$\alpha^\Lambda = \exp\left(\sum_{\lambda\in S \,:\, \lambda\subseteq\Lambda} \beta^\lambda\right) - \sum_{\lambda\subsetneq\Lambda} \alpha^\lambda.$$

## 3.3 Probability Distribution for a Bayesian Network

Given a Bayesian Network with nodes $X = [X_i]_{i=1}^n$, we have the full Joint Probability Distribution as in (2.5), after specifying the conditional probabilities

$\mathbb{P}\left(X_i|\text{Pa}(X_i)\right)$ for each node $X_i$. As in Section 6, we might want to represent the natural logarithm of the full Joint Probability Distribution as a pseudo-Boolean function. Observe that

$$\ln \mathbb{P}(X_1, \cdots, X_n) = \sum_{i=1}^{n} \ln \mathbb{P}\left(X_i|\text{Pa}(X_i)\right),$$

where for each $i$, $\mathbb{P}\left(X_i|\text{Pa}(X_i)\right)$ is a function of the Random Variables in $\{X_i\} \cup \text{Pa}(X_i)$. Thus, if we know the pseudo-Boolean representation of $\ln \mathbb{P}\left(X_i|\text{Pa}(X_i)\right)$ for each $i$, their sum equals the natural logarithm of the full Joint Probability Distribution.

Given an index $i$, we write the pseudo-Boolean representation of the conditional probability distribution $\mathbb{P}\left(X_i|\text{Pa}(X_i)\right)$ as

$$\ln \mathbb{P}\left(X_i|\text{Pa}(X_i)\right) = \sum_{\lambda \in S_i} \gamma_i^\lambda \prod_{X_k \in \lambda} X_k,$$

where

$$S_i = 2^{\{j \ | \ X_j \in \{X_i\} \cup \text{Pa}(X_i)\}}.$$

Note that here, the full representation is chosen, that is, we sum over the power set of $\{X_i\} \cup \text{Pa}(X_i)$. Also, from here, we let the coefficients of the pseudo-Boolean representation of probability distributions be denoted by $\gamma$s, in fact for the conditional distribution $\mathbb{P}\left(X_i|\text{Pa}(X_i)\right)$, we use $\{\gamma_i^\lambda\}_\lambda$.

Now, we assume that the nodes represent binary Random Variables, that is $X_i \in \{0, 1\}$. Observe, that the coefficients $\gamma_i^\lambda$ can be calculated recursively, in order of increasing size of $\lambda$. In fact, first let

$$\gamma_i^\varnothing = \ln \mathbb{P}\left(X_i = 0 | X_k = 0 \ \forall X_k \in \text{Pa}(X_i)\right).$$

Then, for any subset $\Lambda \subseteq \text{Pa}(X_i)$,

$$\gamma_i^\Lambda = \ln \mathbb{P}\left(X_i = 0 | X_k = 1 \ \forall X_k \in \Lambda \text{ and } X_j = 0 \ \forall X_j \in \text{Pa}(X_i) \backslash \Lambda\right) - \sum_{\lambda \subsetneq \Lambda} \gamma_i^\lambda.$$

Similarly, for any set $\Lambda = \{X_i\} \cup \Psi$ where $\Psi \subseteq \text{Pa}(X_i)$,

$$\gamma_i^\Lambda = \ln \mathbb{P}\left(X_i = 1 | X_k = 1 \ \forall X_k \in \Psi \text{ and } X_j = 0 \ \forall X_j \in \text{Pa}(X_i) \backslash \Psi\right) - \sum_{\lambda \subsetneq \Lambda} \gamma_i^\lambda.$$

### 3.3.1 An example Network

This example will illustrate the Joint Probability Distribution for the tests presented in Sections 8 and 9. We have a Bayesian Network over Random Variables that are either on or off, corresponding to the values 1 and 0, respectively. Assume we are given a probability $p_{X_i}$ for each root node $X_i$, and then let

$$p_{X_i} = \mathbb{P}(X_i = 1) = 1 - \mathbb{P}(X_i = 0).$$

This implies the following formula for the root nodes $X_i$,

$$\ln \mathbb{P}(X_i) = \ln(1 - p_{X_i}) + X_i(\ln p_{X_i} - \ln(1 - p_{X_i})),$$

indicating

$$\begin{aligned} \gamma_i^{\varnothing} &= \ln(1 - p_{X_i}) \\ \gamma_i^{\{i\}} &= \ln p_{X_i} - \ln(1 - p_{X_i}). \end{aligned}$$

For each edge $e_{X_i,X_j}$ from node $X_i$ to node $X_j$, assume we are given two probabilities, $p^0_{X_i,X_j}$ and $p^1_{X_i,X_j}$, respectively. In fact, let

$$p^0_{X_i,X_j} = \mathbb{P}(X_j \text{ receives signal from } X_i | X_i = 0)$$

and

$$p^1_{X_i,X_j} = \mathbb{P}(X_j \text{ receives signal from } X_i | X_i = 1).$$

Whether a child $X_j$ receives a signal from one of its parents, say $X_i$, is assumed to be conditionally independent of whether a signal is received in any other pair of child and parent, given the value of the parent $X_i$. Assume that for any non root node, its value is 1 if it has received a signal from at least one of its parents, and 0 otherwise.

Given a node $X_j$ and an assignment $X_i = x_i$ to each of its parents $X_i \in \text{Pa}(X_j)$, we let

$$\text{Pa}(X_j) = \{X_i \mid i \in \Lambda \cup \Psi\},$$

where $\Lambda \cup \Psi$ is a decomposition of $X_j$s parents such that

$$x_i = \begin{cases} 1 & \text{if } X_i \in \Lambda, \\ 0 & \text{if } X_i \in \Psi. \end{cases}$$

Observe that

$$\begin{aligned}\ln \mathbb{P}(X_j = 0|\mathrm{Pa}(X_j)) &= \ln\left(\prod_{i\in\Lambda}\left(1-p^1_{X_i,X_j}\right)\prod_{i\in\Psi}\left(1-p^0_{X_i,X_j}\right)\right)\\ &= \sum_{i\in\Lambda}\ln\left(1-p^1_{X_i,X_j}\right)+\sum_{i\in\Psi}\ln\left(1-p^0_{X_i,X_j}\right),\end{aligned}$$

which implies

$$\gamma_j^{\Lambda} = \sum_{i\in\Lambda}\ln\left(1-p^1_{X_i,X_j}\right)+\sum_{i\in\Psi}\ln\left(1-p^0_{X_i,X_j}\right)-\sum_{\lambda\subsetneq\Lambda}\gamma_j^{\lambda} \tag{3.4}$$

and

$$\gamma_j^{\Lambda\cup\{j\}} = \ln\left(1-\prod_{i\in\Lambda}\left(1-p^1_{X_i,X_j}\right)\prod_{i\in\Psi}\left(1-p^0_{X_i,X_j}\right)\right)-\sum_{\lambda\subsetneq\Lambda\cup\{j\}}\gamma_j^{\lambda\cup\{j\}}.$$

Observe from (3.4) that for a non root node $X_j$, we get

$$\gamma_j^{\varnothing} = \sum_{X_i\in\mathrm{Pa}(X_j)}\ln(1-p^0_{X_i,X_j}),$$

and if $X_k\in\mathrm{Pa}(X_j)$,

$$\begin{aligned}\gamma_j^{\{k\}} &= \ln(1-p^1_{X_k,X_j})+\sum_{X_i\in\mathrm{Pa}(X_j)\setminus\{X_k\}}\ln(1-p^0_{X_i,X_j})-\gamma_j^{\varnothing}\\ &= \ln(1-p^1_{X_k,X_j})-\ln(1-p^0_{X_k,X_j}).\end{aligned}$$

But then also, for $X_k, X_\ell\in\mathrm{Pa}(X_j)$ where $k\neq\ell$, (3.4) reduces to

$$\gamma_j^{\{k,\ell\}} = 0.$$

In fact, we get

$$\gamma_j^{\Lambda} = 0 \text{ when } |\Lambda|\geqslant 2$$

by induction on the size of $\Lambda$.

As an example, we let $p^0_{X_i,X_j} = \epsilon$ for all edges $e_{X_i,X_j}$ and calculate the representation of the corresponding conditional distribution for some of the variables in Figure 1. For example, we get

$$\ln\mathbb{P}(K_1) = \ln(1-p_{K_1})+K_1(\ln p_{K_1}-\ln(1-p_{K_1})),$$

$$\ln \mathbb{P}(P_1|K_1) = \ln(1-\epsilon) + K_1 \ln \frac{1-p_{K_1,P_1}}{1-\epsilon} + P_1 \ln \frac{\epsilon}{1-\epsilon} + K_1 P_1 \ln \frac{(1-\epsilon)p_{K_1,P_1}}{(1-p_{K_1,P_1})\epsilon}.$$

and

$$\ln \mathbb{P}(S_1|P_1) = \ln(1-\epsilon) + P_1 \ln \frac{1-p_{P_1,S_1}}{1-\epsilon} + S_1 \ln \frac{\epsilon}{1-\epsilon} + P_1 S_1 \ln \frac{(1-\epsilon)p_{P_1,S_1}}{(1-p_{P_1,S_1})\epsilon}.$$

Note that the formulas for $\ln \mathbb{P}(S_2|P_2)$ and $\ln \mathbb{P}(S_3|P_2)$ are similar to those for $\ln \mathbb{P}(P_1|K_1)$ and $\ln \mathbb{P}(S_1|P_1)$.

## 4 Approximate Forward-Backward Algorithm

From Tjelmeland and Austad (2011), we have an approximate forward-backward algorithm for probability distributions of the form

$$\mathbb{P}(X_1, \cdots, X_n) = \frac{1}{C} \exp\left(\sum_{\lambda \in S} \gamma^\lambda \prod_{k \in \lambda} X_k\right)$$

That is, the algorithm is constructed to sum out Random Variables, and return the value of the sum. This is useful for example when working with Markov Random Fields, where the normalization constant $C$ might be unknown. From Section 2.3, we recall that

$$C = \sum_{X_1, \cdots, X_n} \exp\left(\sum_{\lambda \in S} \gamma^\lambda \prod_{k \in \lambda} X_k\right).$$

Another application is for calculating the expected value $\mathbb{E}h$ for some function $h$ on a set of Random Variables. Let

$$h(X_1, \cdots, X_n) = \exp\left(\sum_{\lambda \in S} \beta^\lambda \prod_{k \in \lambda} X_k\right).$$

Then,

$$\begin{aligned} \mathbb{E}h &= \sum_{X_1, \cdots, X_n} h(X_1, \cdots, X_n)\mathbb{P}(X_1, \cdots, X_n) \\ &= \sum_{X_1, \cdots, X_n} \exp\left(\sum_{\lambda \in S} (\beta^\lambda + \gamma^\lambda) \prod_{k \in \lambda} X_k\right). \end{aligned}$$

Also, note that if $h$ is a function which depends on more variables than our Random Variables $X_1, \cdots, X_n$, say

$$h = h(X_1, \cdots, X_n, X_{n+1}, \cdots X_{n+m}),$$

we would have

$$h(X_1, \cdots, X_{n+m}) = \exp\left(\sum_{\lambda \in S} \beta^\lambda \prod_{k \in \lambda} X_k\right).$$

for a collection $S$ of index sets $\lambda$ where some of the indexes $k \in \lambda$ might have $k > n$. We get

$$\begin{aligned}\mathbb{E}h &= \sum_{X_1, \cdots, X_n} h(X_1, \cdots, X_{n+m}) \mathbb{P}(X_1, \cdots, X_n) \\ &= \sum_{X_1, \cdots, X_n} \exp\left(\sum_{\lambda \in S} (\beta^\lambda + \gamma^\lambda) \prod_{k \in \lambda} X_k\right),\end{aligned}$$

which corresponds to looking for a function $C(X_{n+1}, \cdots X_{n+m})$. Observe how these three problems are essentially the same, and thus we can focus on how to find the normalization constant $C$.

The exact forward-backward algorithm works by summing out one variable at the time. That is, assume we want to calculate

$$\begin{aligned}C &= \sum_{X_n, X_{n-1} \cdots, X_1} \exp\left(\sum_{\lambda \in S} \gamma^\lambda \prod_{k \in \lambda} X_k\right) \\ &= \sum_{X_n} \cdots \sum_{X_2} \sum_{X_1} \exp\left(\sum_{\lambda \in S} \gamma^\lambda \prod_{k \in \lambda} X_k\right) \\ &= \sum_{X_n} \cdots \sum_{X_2} \left(\sum_{X_1} \exp\left(\sum_{\lambda \in S} \gamma^\lambda \prod_{k \in \lambda} X_k\right)\right).\end{aligned}$$

First, $X_1$ is summed out, to get

$$\sum_{X_1} \exp\left(\sum_{\lambda \in S} \gamma^\lambda \prod_{k \in \lambda} X_k\right) = \exp\left(\sum_{\lambda \in \hat{S}} \hat{\gamma}^\lambda \prod_{k \in \lambda} X_k\right)$$

for some $\hat{S}$ and some collection of coefficients $\{\hat{\gamma}^\lambda\}_{\lambda \in \hat{S}}$. Note that

$$X_1 \notin \lambda \qquad \forall \lambda \in \hat{S}.$$

After this step,

$$C = \sum_{X_2 \cdots, X_n} \exp\left( \sum_{\lambda \in \hat{S}} \hat{\gamma}^\lambda \prod_{k \in \lambda} X_k \right).$$

Continuing like this, we finally end up with some $\bar{S}$ and some collection of coefficients $\{\bar{\gamma}^\lambda\}_{\lambda \in \bar{S}}$ such that

$$C = \sum_{X_n} \exp\left( \sum_{\lambda \in \bar{S}} \bar{\gamma}^\lambda \prod_{k \in \lambda} X_k \right).$$

We observe that in this stepwise procedure, each step is essentially the same. Therefore, it is sufficient to consider how to calculate

$$C^1(X_2, \cdots, X_n) = \sum_{X_1} \exp\left( \sum_{\lambda \in S} \gamma^\lambda \prod_{k \in \lambda} X_k \right).$$

Observe that for any $\dot{S} \subseteq S$,

$$\sum_{\lambda \in S} \gamma^\lambda \prod_{k \in \lambda} X_k = \sum_{\lambda \in \dot{S}} \gamma^\lambda \prod_{k \in \lambda} X_k + \sum_{\lambda \in S \backslash \dot{S}} \gamma^\lambda \prod_{k \in \lambda} X_k.$$

Now, let

$$\dot{S} = \{\lambda \in S \mid 1 \in \lambda\}$$

Then,

$$\begin{aligned} C^1(X_2, \cdots, X_n) &= \sum_{X_1} \exp\left( \sum_{\lambda \in \dot{S}} \gamma^\lambda \prod_{k \in \lambda} X_k + \sum_{\lambda \in S \backslash \dot{S}} \gamma^\lambda \prod_{k \in \lambda} X_k \right) \\ &= \sum_{X_1} \exp\left( \sum_{\lambda \in \dot{S}} \gamma^\lambda \prod_{k \in \lambda} X_k \right) \exp\left( \sum_{\lambda \in S \backslash \dot{S}} \gamma^\lambda \prod_{k \in \lambda} X_k \right) \\ &= \exp\left( \sum_{\lambda \in S \backslash \dot{S}} \gamma^\lambda \prod_{k \in \lambda} X_k \right) \sum_{X_1} \exp\left( \sum_{\lambda \in \dot{S}} \gamma^\lambda \prod_{k \in \lambda} X_k \right). \end{aligned}$$

Thus, essentially, what needs to be calculated is

$$\sum_{X_1} \exp\left( \sum_{\lambda \in S \text{ s.t. } 1 \in \lambda} \gamma^\lambda \prod_{k \in \lambda} X_k \right). \tag{4.1}$$

Let

$$\text{Neigh}(X_i, S) = \{X_j \mid \exists \lambda \in S \text{ such that } \{i, j\} \subseteq \lambda\}.$$

Then, $\text{Neigh}(X_1, S)$ is the set of nodes with indexes appearing in (4.1). In fact, if we look at

$$\frac{1}{\tilde{C}} \sum_{\lambda \in S} \gamma^\lambda \prod_{k \in \lambda} X_k$$

as the probability distribution for a Markov Random Field, then, $\text{Neigh}(X_1, S)$ is the set of neighboring nodes of $X_1$. In a visualization of this Markov Random Field, this would be illustrated by an undirected arrow between each pair of neighbors. Also, this implies that for this Markov Random Field $G = (V, E)$, there is an edge $e$ between the nodes $X_i$ and $X_j$, denoted

$$e = \{X_i, X_j\} \in E$$

for each node $X_j \in \text{Neigh}(X_i, S)$. Recall from Section 2.3, that we can only have conditional dependencies between nodes $X_i$ and $X_j$ given all other Random Variables if $X_i$ and $X_j$ are neighbors.

Calculating (4.1) has a complexity of

$$\mathcal{O}(2^{|\text{Neigh}(X_i, S)|}).$$

Tjelmeland and Austad (2011) presents an algorithm which finds an approximate value of the sum with a complexity of $\mathcal{O}(2^\nu)$, for some input constant $\nu$. That is, if

$$|\text{Neigh}(X_i, S)| \leqslant \nu,$$

the exact value of the sum is calculated. Otherwise, if

$$|\text{Neigh}(X_i, S)| > \nu,$$

the approximation

$$f \equiv \sum_{X_1} \exp\left(\sum_{\lambda \in S} \gamma^\lambda \prod_{k \in \lambda} X_k\right) \approx \sum_{X_1} \exp\left(\sum_{\lambda \in \tilde{S}} \tilde{\gamma}^\lambda \prod_{k \in \lambda} X_k\right) \equiv \tilde{f}$$

is made for some $\tilde{S}$ and some collection of coefficients $\{\tilde{\gamma}^\lambda\}_{\lambda \in \tilde{S}}$ such that

$$|\text{Neigh}(X_1, \tilde{S})| \leqslant \nu.$$

The approximating function $\tilde{f}$ for the function $f$ is chosen to minimize the error sum of squares

$$SSE(f, \tilde{f}) = \sum_X \Big( f(X) - \tilde{f}(X) \Big)^2 .$$

Now, the operation of summing out a Random Variable has a complexity which is bounded by $\mathcal{O}(2^\nu)$ for the input constant $\nu$.

Also, in Tjelmeland and Austad (2011), a couple of possible ways to calculate bounds on the error are discussed. These bounds together with the approximate value of the sum, gives upper and lower bounds on the exact value. What is referred to as "Optimal bounds for pseudo-Boolean functions" in Tjelmeland and Austad (2011) is already implemented, and will be used for the tests later in this report. This computer program, implemented by Tjelmeland and Austad, will be referred to as "The AMRF Programs", where AMRF is an abbreviation for Approximate Markov Random Field. For a short user oriented presentation of "The AMRF Programs", the reader is referred to Appendix B.

## 5 Mathematical Formulation of the Problem

Given a graphical model with $n$ nodes numbered from 1 to $n$, we assume that each node $i$ is associated with a Boolean Random Variable $X_i$. Following the notation from Tjelmeland and Austad (2011), let

$$N = \{1, 2, \cdots, n\},$$

and also, let

$$\Omega = \{0, 1\}^n$$

be the set of all Boolean vectors of length $n$. The vector

$$X = [X_i]_{i=1}^n,$$

containing all the Random Variables $X_i$ in our graphical model, is then a Random Variable in the probability space

$$(\Omega, 2^\Omega, \mathbb{P}),$$

where $\mathbb{P}$ is the Joint Probability Distribution on the graph.

Observe that the function

$$\dot{g} \;:\; \Omega \;\rightarrow\; 2^N \qquad \text{such that} \qquad \dot{g}(v) = \{\; j \mid v_j = 1 \;\}$$

is one-to-one and onto, and thus defines a bijection between vectors in $\Omega$ and subsets of $N$. This allows us to associate $\Omega$ with the power set $2^N$, and talk about their corresponding elements. That is, a given subset of $N$ contains the indexes of the 1s in the corresponding vector in $\Omega$.

**Definition 5.1.** *A decision is a deterministic Boolean vector $Z$ of length $n$, that is, with each entry corresponding to one of the Random Variables $X_i$ in our graphical model.*

Let $u$ be an integer, and also, let $\Gamma \subseteq N$ be the indexes of a subset of all nodes in the graph. Define

$$\Omega^{\Gamma,u} = \left\{ Z \in \Omega : \quad Z_i = 0 \ \forall i \notin \Gamma, \quad \text{and} \quad \sum_{i=1}^{n} Z_i \leqslant u \right\},$$

to be our decision space. That is, for a decision $Z$, we only allow the entries $Z_i$ with index $i \in \Gamma$ to take on the value 1, i.e. all other entries have to be 0. Also, no more than $u$ entries are allowed to have the value 1. Note that $\Omega^{\Gamma,u} \subseteq \Omega$.

Given a function

$$f \ : \ \Omega^{\Gamma,u} \ \rightarrow \ \mathbb{R},$$

we look for a decision $Z^* \in \Omega^{\Gamma,u}$ that maximizes the value of $f(Z)$. That is,

$$Z^* = \arg \max_{Z \in \Omega^{\Gamma,u}} \{f(Z)\} .$$

We will refer to such a decision $Z^*$ as the optimal decision according to the function $f$ .

Assume we want to look for nodes $i$ where $X_i = 1$, and that it is not possible to observe all nodes. In fact, assume that there is only a subset of the nodes for which it is possible to observe the value of $X_i$, and also that there is an upper limit of how many nodes we can observe. We let the value of the entries $Z_i$ in the decision vector $Z$ denote whether we are going to check the value of the corresponding Random Variable, as in

$$Z_i = \begin{cases} 1 & \text{if we observe the value of } X_i, \\ 0 & \text{if we don't observe the value of } X_i. \end{cases} .$$

Then, $\Gamma \subseteq N$ contains the indexes of the nodes that are observable, and $u$ is the maximum number of nodes we are allowed to check. Also, for our graphical model which encodes probabilistic information about the unknown variables in $X$, we assume

$$f(Z) = \mathbb{E}h(X,Z) = \int_\Omega h(X,Z)\mathbb{P}(dX)$$

for some function

$$h \ : \ \Omega \times \Omega^{\Gamma,u} \ \to \ \mathbb{R}.$$

Let

$$c \ : \ \Omega^{\Gamma,u} \to \ \mathbb{R}^+$$

represent the cost of a decision. That is, $c(Z)$ is the cost associated with observing $X_i$ for each $i$ such that $Z_i = 1$. Also, let

$$w \ : \ \Omega \times \Omega^{\Gamma,u} \to \ \mathbb{R}^+$$

be the income of a decision for a specific assignment to the Random Variables in the graph. That is $w(X,Z)$ is the income associated with the observations according to the decision $Z$ if the Random Variables had the assignment as in $X$. Now, we let $h(X,Z) = w(X,Z) - c(Z)$ and get

$$\begin{aligned} f(z) &= \mathbb{E}\left(\text{profit}|_{\text{following } Z}\right) \\ &= \mathbb{E}\left(w(X,Z) - c(Z)\right) \\ &= \mathbb{E}w(X,Z) - c(Z) \\ &= \sum_X w(X,Z)\mathbb{P}(X) - c(Z). \end{aligned}$$

This is the function $f$ for which we want to find the optimal decision $Z^*$, and we will refer to this as the optimal decision $Z^*$, and assume this choice of $f$ to be implicit.

As in Martinelli et al. (2011), the graphical model could be a Directed Acyclic Graph which denotes the causal conditional distributions for the presence of hydrocarbons in certain areas. We would let $X_i = 1$ denote presence of hydrocarbons in the area associated with node $X_i$, and correspondingly, $X_i = 0$ would denote absence of hydrocarbons. The nodes with indexes in $\Gamma$ would represent the Potential Drilling Sites, while the other nodes are used to impose geological causal dependencies. Then, $u$ is the maximum number of nodes we are allowed to drill. Also, $c(Z)$ would be the cost of drilling the Potential Drilling Sites with a nonzero entry in $Z$. Correspondingly, $w(X,Z)$ would be the net income after the finds of hydrocarbons according to $X$ and $Z$. That is, we would find hydrocarbons in node $X_i$ if and only if $X_i = Z_i = 1$.

# 6 Mathematical Formulation of the Solution

Typically, it is too time consuming to calculate the exact value of $f(Z)$ for all $Z$s in our decision space $\Omega^{\Gamma,u}$. That is why we have to look at approximations. Since we still want to make sure that we actually find the optimal decision, we look at upper and lower bounds for the value of $f(Z)$ for each $Z$. An algorithm for finding upper and lower bounds for the normalization constant $C$ of a given probability distribution

$$\mathbb{P}(X_1, \cdots, X_n) = \frac{1}{C} \exp\left( \sum_{\lambda \in S} \gamma^{\lambda} \prod_{k \in \lambda} X_k \right),$$

as in Tjelmeland and Austad (2011), is already implemented. This will be used as a "Black Box" for finding upper and lower bounds on the value of $f(Z) = \mathbb{E}h(X, Z)$ for a given $Z$. Recall from Section 4, that finding the expected value of a function $h$ on some Random Variables $X$ is essentially the same problem as finding the normalization constant for a distribution $P(X)$ that is not normalized. Letting some set of upper and lower bounds be implicit, this lets us define the functions

$$f^+ \ : \ \Omega^{\Gamma,u} \ \rightarrow \ \mathbb{R}$$

and

$$f^- \ : \ \Omega^{\Gamma,u} \ \rightarrow \ \mathbb{R}.$$

We let $f^+(Z)$ denote the upper bound for the expected profit, and correspondingly, $f^-(Z)$ denote the lower bound.

Given a set of bounds, we have to use the values of $f^+(Z)$ and $f^-(Z)$ for all $Z$s to separate out as few candidates for the optimal decision as possible. By the definition of the optimal decision $Z^*$, we know that

$$f(Z^*) \geqslant f(Z) \qquad \forall \ Z \in \Omega^{\Gamma,u}.$$

Also, for the upper and lower bounds, $f^+(Z)$ and $f^-(Z)$, respectively, we know that

$$f^-(Z) \leqslant f(Z) \leqslant f^+(Z) \qquad \forall \ Z \in \Omega^{\Gamma,u}. \tag{6.1}$$

From the two above equations, we deduce that

$$f(Z^*) \geqslant f^-(Z) \qquad \forall \ Z \in \Omega^{\Gamma,u},$$

which again implies

$$f(Z^*) \geqslant \max_{Z \in \Omega^{\Gamma,u}} f^-(Z).$$

Since (6.1) is true for any decision $Z \in \Omega^{\Gamma,u}$, more specifically we also have

$$f^-(Z^*) \leqslant f(Z^*) \leqslant f^+(Z^*),$$

which finally leaves us with

$$f^+(Z^*) \geqslant \max_{Z \in \Omega^{\Gamma,u}} f^-(Z).$$

Thus, we know that any decision $\tilde{Z}$ with

$$f^+(\tilde{Z}) < \max_{Z \in \Omega^{\Gamma,u}} f^-(Z).$$

is not a candidate for the optimal decision. This also tells us that if we could find a $\dot{Z}$ such that

$$f^-(\dot{Z}) \geqslant f^+(Z) \qquad \forall\ Z \in \Omega^{\Gamma,u} \backslash \{\dot{Z}\},$$

we would know that $Z^* = \dot{Z}$. In general, the upper and lower bounds are not tight enough to separate out only one candidate for $Z^*$, but hopefully we would be able to exclude a significant fraction of the possible decisions $Z$. Note that it is sufficient to observe that

$$f^+(\tilde{Z}) < f^-(\dot{Z}).$$

for some decisions $\tilde{Z}, \dot{Z} \in \Omega^{\Gamma,u}$ to rule out $\tilde{Z}$ as a candidate for the optimal decision.

Now, we have a plan for how to deal with a set of evaluations of the upper and lower bounds. Our focus will now be on how to obtain such bounds. As defined in Section 5, we assume that we are given a cost function $c(Z)$ and an income function $w(X, Z)$. Recall from Section 3, that the cost function can be represented as a pseudo-Boolean function.

$$c(Z) = \sum_{\lambda \in T_{cost}} \alpha_{cost}^{\lambda} \prod_{k \in \lambda} Z_k, \tag{6.2}$$

where $T_{cost}$ is some subset of the power set $2^{\Omega^{\Gamma,\nu}}$. Since we only observe the values of $X$ in the nonzero entries of $Z$, it is natural to assume that $w$ is a function of the scalar product $X \cdot Z$, where $(X \cdot Z)_i = X_i \cdot Z_i$. Then, we can write $w$ as a pseudo-Boolean function of the following form,

$$w(X, Z) = \sum_{\lambda \in T_{inc}} \alpha_{inc}^{\lambda} \prod_{k \in \lambda} X_k \cdot Z_k. \tag{6.3}$$

Since we have assumed the income function to be a positive function $w(X, Z) > 0$ of the decision $Z$ and the Random Variables $X$, we also let $S_{inc}$ denote a collection of sets of indexes and $\{\beta_{inc}^{\lambda}\}_{\lambda \in S_{inc}} \subset \mathbb{R}$ denote coefficients such that

$$w(X, Z) = \exp\left(\sum_{\lambda \in S_{inc}} \beta_{inc}^{\lambda} \prod_{k \in \lambda} X_k \cdot Z_k\right). \tag{6.4}$$

Recall that Section 3.2 presented an algorithm for how to convert between the representations in (6.3) and (6.4). As in Section 3.3, we also write the probability distribution of the Random Variables $X$ in our graphical model as

$$\mathbb{P}(X) = \exp\left(\sum_{\lambda \in S} \gamma^{\lambda} \prod_{k \in \lambda} X_k\right). \tag{6.5}$$

We introduce a variable $\tilde{X}$, which holds information from both the Random Variable $X$ and the decision $Z$. In fact, we let $\tilde{X}$ be a vector of length $2n$, where the first $n$ entries are the entries in $X$, and the next $n$ entries are the entries in $Z$. That is, for the Random Variable $X$ and a decision $Z$, element-wise we define

$$\tilde{X}_i = \begin{cases} X_i & \text{if } i \leqslant n, \\ Z_{i-n} & \text{if } i > n. \end{cases} \tag{6.6}$$

Observe for a decision $Z$, $\tilde{X}(X, Z)$ is a Random Variable where

$$\mathbb{P}\left(\tilde{X} = [x, z]\right) = \begin{cases} \mathbb{P}(X = x) & \text{if } z \neq Z, \\ 0 & \text{if } z = Z, \end{cases}$$

for Boolean vectors $x, z$ of length $n$. Thus, implicitly letting the choice of $Z$ deterministically fill out the last $n$ entries of $\tilde{X}$, and also letting $\tilde{X}_k(X, Z)$ denote the $k$th entry of $\tilde{X}(X, Z)$, (6.5) can be rewritten as

$$\mathbb{P}\left(\tilde{X}(X, Z)\right) = \exp\left(\sum_{\lambda \in S} \gamma^{\lambda} \prod_{k \in \lambda} \tilde{X}_k(X, Z)\right),$$

where the collection $S$ of index sets $\lambda$ is unchanged.

Correspondingly, (6.4) can be translated to

$$w(X,Z) = \exp\left(\sum_{\lambda\in S_{inc}} \beta_{inc}^{\lambda} \prod_{k\in\lambda} \left(\tilde{X}_k(X,Z)\cdot\tilde{X}_{n+k}(X,Z)\right)\right).$$

We define

$$\theta \ : \ 2^N \ \to 2^{\{1,\cdots,2n\}} \qquad \text{such that} \qquad \theta(\lambda) = \lambda \cup \{j+n \mid j\in\lambda\},$$

and let

$$\tilde{S}_{inc} = \{\theta(\lambda) \mid \lambda \in S_{inc}\}.$$

Also, for each $\lambda \in S_{inc}$, we let

$$\tilde{\beta}_{inc}^{\theta(\lambda)} = \beta_{inc}^{\lambda}.$$

This allows us to define

$$\tilde{w} \ : \ \{0,1\}^{2n} \ \to \mathbb{R} \qquad \text{such that} \qquad \tilde{w}(\tilde{X}) = \exp\left(\sum_{\lambda\in\tilde{S}_{inc}} \tilde{\beta}_{inc}^{\lambda} \prod_{k\in\lambda} \tilde{X}_k\right).$$

Note all of this is just to get

$$\tilde{w}\left(\tilde{X}(X,Z)\right) = w(X,Z).$$

Assuming $\tilde{S}_{inc} = S$, as we could ensure by including zero-valued coefficients if necessary, the expected value $\mathbb{E}w(X,Z)$ can be written as

$$\begin{aligned}\mathbb{E}w(X,Z) &= \sum_X w(X,Z)\mathbb{P}(X)\\ &= \sum_X \tilde{w}\left(\tilde{X}(X,Z)\right)\mathbb{P}\left(\tilde{X}(X,Z)\right)\\ &= \sum_X \exp\left(\sum_{\lambda\in S}\left(\tilde{\beta}_{inc}^{\lambda}+\gamma^{\lambda}\right)\prod_{k\in\lambda}\tilde{X}_k(X,Z)\right)\end{aligned} \tag{6.7}$$

Assume we have the representation of $w(X,Z)$ as in (6.4) and the representation of $\mathbb{P}(X)$ as in (6.5). After some modifications, as illustrated in the functions defined for the augmented variable $\tilde{X}$, these can be plugged into "The

AMRF Programs", which will return upper and lower bounds, respectively, on $\mathbb{E}w(X, Z)$ according to the input constant $\nu$ controlling the complexity of the algorithm. The obvious upper and lower bounds, respectively, on $f(Z)$ would then be the corresponding bound on $\mathbb{E}w(X, Z)$ minus the value of $c(Z)$. This way of calculating the value of $f(Z)$ corresponds to the equation

$$f(Z) = \mathbb{E}w(X, Z) - c(Z),$$

as we saw it in Section 5. We will refer to this method as to Method 1.

Section 5 also presented the equivalent formula

$$f(Z) = \mathbb{E}\left(w(X, Z) - c(Z)\right),$$

which introduces a slightly different way of finding bounds on $f(Z)$. That is, here, we want "The AMRF Programs" to work with the function

$$h(X, Z) = w(X, Z) - c(Z)$$

and return bounds on $f(Z)$ directly. However, $h(X, Z)$ is not a positive function, so we have to add a sufficient constant $M$, to get

$$h(X, Z) + M > 0,$$

and thus be able to write

$$h(X, Z) + M = \exp\left(\sum_{\lambda \in \tilde{S}} \beta^{\lambda} \prod_{k \in \lambda} \tilde{X}_k(X, Z)\right)$$

for some collection of index sets $\tilde{S}$ and corresponding collection of coefficients $\{\beta^{\lambda}\}_{\lambda \in \tilde{S}}$. This will be referred to as Method 2. Note that adding the constant $M$ does not change the optimal decision, since

$$\max_{Z \in \Omega^{\Gamma, u}} \{f(Z) + M\} = \max_{Z \in \Omega^{\Gamma, u}} \{f(Z)\} + M,$$

and thus

$$\arg\max_{Z \in \Omega^{\Gamma, u}} \{f(Z) + M\} = \arg\max_{Z \in \Omega^{\Gamma, u}} \{f(Z)\}.$$

# 7 Generating Bayesian Networks

To test our solution method, which is presented in Section 6, we need Bayesian Networks with corresponding cost and income functions. We want to be able to easily generate several test cases, and we want to be able to do this for different sizes of the graph. We also want to study cases where observing a node is relatively cheap compared to the possible incomes, and cases where it is expensive. We also want to be able to control the sparseness of the graph, i.e. the density of edges between the nodes.

The problem discussed in Section 5 assumes that we are given a graphical model $G = (X, E)$ with $n$ nodes represented as entries in the vector

$$X = [X_1, \cdots, X_n],$$

with a corresponding Joint Probability Distribution

$$\mathbb{P}(X_1, \cdots, X_n).$$

For this graphical model, we also define a decision vector

$$Z = [Z_1, \cdots, Z_n],$$

with each entry $Z_i$ corresponding to the Random variable $X_i$. It also assumes a cost function $c(Z)$ and an income function $w(X, Z)$. Thus, we need to generate sets

$$\{\ G = (X, E),\ \mathbb{P}(X),\ c(Z),\ w(X, Z)\ \}.$$

to test our solution.

We will assume that our graphical model $G$ is a Bayesian Network, and that the observable nodes are the leaf nodes in the graph. Thus, $\Gamma$ is the indexes of the leaf nodes. Also, we assume the numbering $i$ of the nodes $X_i$ to be according to some topological ordering where any root node is preceding all non root nodes and any leaf node is succeeding all non leaf nodes. We adapt the terminology from Martinelli et al. (2011), and refer to the root nodes as Kitchens (K), the leaf nodes as Segments (S), and the intermediate nodes as Prospects (P). A Bayesian Network with $\#K$ Kitchens, $\#P$ Prospects and $\#S$ Segments, is therefore a graphical model with

$$n = \#K + \#P + \#S$$

nodes. Our numbering of the nodes plus the above terminology, introduce the following re-naming of the nodes $X_i$,

$$X_i \to \begin{cases} K_i, & 1 \leqslant\ i\ \leqslant \#K \\ P_{i-\#K}, & \#K <\ i\ \leqslant \#K + \#P \\ S_{i-\#K-\#P}, & \#K + \#P <\ i\ \leqslant \#K + \#P + \#S \end{cases},$$

which results in

$$X = [K_1, K_2 \cdots, K_{\#K}, P_1, P_2, \cdots, P_{\#P}, S_1, S_2, \cdots, S_{\#S}].$$

## 7.1 The Directed Acyclic Graph

The first step is to generate the graph $G = (X, E)$. We let the number of Kitchens $\#K$, the number of Prospects $\#P$, the number of Segments $\#S$ and the total number of edges $\#E$ in the graph be deterministic. What is generated, is the start and end point of each edge. By definition, each Kitchen has only out-edges, that is, it can only be the start point of an edge. Also, we assume that any edge from a Kitchen has to enter a Prospect. This implies that

$$\forall K_i \qquad \mathrm{Ch}(K_i) \subseteq \{\ P_1,\ P_2, \cdots, P_{\#P}\ \}\,.$$

We also assume

$$\forall K_i \qquad P_i \in \mathrm{Ch}(K_i), \tag{7.1}$$

which implies that we need to have

$$\#K \leqslant \#P.$$

Correspondingly, by definition, the Segments can only have in-edges, that is, they can only be end points of an edge. More specifically, we assume that each Segment has exactly one parent, which needs to be a Prospect. That is,

$$\forall S_i \qquad \exists P_j \text{ such that } \mathrm{Pa}(S_i) = \{P_j\}. \tag{7.2}$$

We also assume that among the children of a Prospect, there is at least one Segment. That implies that we need to have

$$\#P \leqslant \#S.$$

The fact that each Prospect has at least one Segment among its children, ensures that no Prospects are leaf nodes. The property in (7.1) ensures that no Kitchens are leaf nodes either. Correspondingly, the property in (7.2) ensures that the Segments are not root nodes. To ensure that no Prospects are root nodes, in addition to the edges in (7.1), we need at least one parent for each Prospect $P_i$ with index $i > \#K$. By ensuring that all Prospects and Segments are not root nodes, we get that

$$\#E \geqslant \#P + \#S.$$

Also, there is an upper limit to the possible number of edges. Actually,

$$\#E = \text{The number of edges with a Prospect as end point } + \#S.$$

Since the numbering of the nodes is assumed to follow some topological ordering, we know that

$$\forall P_i \qquad \mathrm{Pa}(P_i) \subseteq \{\ K_1,\ K_2, \cdots, K_{\#K},\ P_1,\ P_2, \cdots, P_{i-1}\ \},$$

which implies that

$$|\mathrm{Pa}(P_i)| \leqslant \#K + i - 1.$$

Thus, the number of edges with a Prospect as end point is bounded,

$$\sum_{i=1}^{\#P} |\mathrm{Pa}(P_i)| \leqslant \sum_{i=1}^{\#P} (\#K + i - 1) = \#P\left(\#K + \frac{\#P - 1}{2}\right).$$

To sum up the discussion, we get for the number of edges $\#E$

$$\#P + \#S \ \leqslant\ \#E\ \leqslant\ \#P\left(\#K + \frac{\#P-1}{2}\right) + \#S,$$

and for the number of nodes of each kind, $\#K,\ \#P,\ \#S$, we have

$$1\ \leqslant\ \#K\ \leqslant\ \#P\ \leqslant\ \#S = n - \#K - \#P.$$

An example graph according to these rules, can be seen in Figure 6.

Generating the edge set $E$ sums up to the following algorithm. Start with $E = \varnothing$, then add edges as follows

- Add edge $(K_i, P_i)$ for each $i \leqslant \#K$.
- Add edge $(X, P_i)$ for an X drawn uniformly from

  $$\{\ K_1,\ K_2, \cdots, K_{\#K},\ P_1,\ P_2, \cdots, P_{i-1}\ \}$$

  for each $\#K < i \leqslant \#K + \#P$.

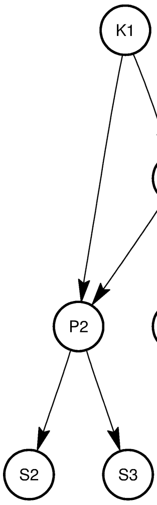

Figure 6: An example plot of a Directed Acyclic Graph with one Kitchen, two Prospects and three Segments.

- Add edge (X,Y), for an X drawn uniformly from all Kitchens and Prospects

$$\{\ K_1,\ K_2, \cdots, K_{\#K},\ P_1,\ P_2, \cdots, P_{\#P}\ \}$$

  and a Y drawn uniformly from the Prospects succeeding X in the ordering. Repeat until the current edge set $E$ has size $\#E - \#S$.

- Start by assuming that each Prospect has one Segment as a child. Then draw a Prospect uniformly, and add one to its number of Segment children. Repeat until the total number of Segment children is $\#S$. Then for each $j \leqslant \#S$, add edge $(P_i, S_j)$ where $i$ starts at $i = 1$ and is increased by 1 whenever $P_i$ has the correct number of Segment children.

## 7.2 The Joint Probability Distribution

The Joint Probability Distribution $\mathbb{P}(X_1, \cdots, X_n)$ for the Bayesian Network over the Random Variables $X_1, \cdots, X_n$, is assumed to be of the same form as in Section 3.3.1. We also assume $p^0_{X_i,X_j} = \epsilon$ for each edge $e_{X_i,X_j}$. That is, what we need to specify to get the full Joint Probability Distribution, is the probability $p_{K_i}$ for each Kitchen $K_i$, and the probability $p^1_{X_i,X_j}$ for each edge $e_{X_i,X_j}$, in addition to the value of the parameter $\epsilon$. The probabilities are all drawn from a beta-distribution. The parameters for the beta-distribution, namely $\alpha, \beta$, are constant within all kitchen probabilities $p_{K_i}$ drawn, and correspondingly, the parameters are constant within all edge probabilities $p^1_{X_i,X_j}$ drawn.

For the hydrocarbon example, as in Martinelli et al. (2011), we let a node $X_i$ correspond to some geographical area. The Kitchen nodes represent places where hydrocarbons can be generated. Then, $p_{K_i}$ is the probability that it has been generated hydrocarbons there. That is, $K_i = 1$ corresponds to the event that hydrocarbons have been generated in the given area, and $K_i = 0$ corresponds to no hydrocarbons generated there. The edges in the graph corresponds to possible directed paths for the hydrocarbons to flow. The probability $p^1_{X_i,X_j}$ corresponding to edge $e_{X_i,X_j}$, is then the probability for hydrocarbons to flow from $X_i$ to $X_j$ assuming that there are hydrocarbons in $X_i$. The Segment nodes are the Potential Drilling Sites, and they will contain hydrocarbons, corresponding to $S_j = 1$, if hydrocarbons have arrived along at least one directed path in the graph, starting at some Kitchen $K_i$ with $K_i = 1$.

The Joint Probability Distribution for the hydrocarbon example coincides with the one discussed in Section 3.3.1. However, the above discussion makes it tempting to set $p^0_{X_i,X_j} = 0$, since no hydrocarbon in a node makes it impossible to have hydrocarbons flowing from it. But actually assuming $p^0_{X_i,X_j} = 0$ for some edge $e_{X_i,X_j}$, introduces assignments to the Random Variables of probability 0. Then, the Joint Probability Distribution cannot be expressed as the exponential of a pseudo-Boolean function, as we want. Thus, we introduce a small $\epsilon > 0$, and deterministically set $p^0_{X_i,X_j} = \epsilon$ for all edges $e_{X_i,X_j}$. Then, the pseudo-Boolean representation of the Joint Probability Distribution is calculated as for the example Network in Section 3.3.1.

## 7.3 The Cost and Income functions

With the graph $G = (V, E)$ and the Joint Probability Distribution $\mathbb{P}(X)$ in hand, the last step is to generate the cost function $c(Z)$ and the income function $w(X, Z)$. We will assume a pseudo-Boolean representation of the cost function and the income function, as in (6.2) and (6.3), respectively. Note that in our setting, we have

$$\Gamma = \{\#K + \#P + 1,\ \#K + \#P + 2,\ \cdots\ ,\ \#K + \#P + \#S\},$$

that is, the observable nodes are exactly the leaf nodes, which equals the set of Segments, placed in the end in the topological ordering by choice. Since we study decisions $Z \in \Omega^{\Gamma,u}$, we let

$$T_{inc}, T_{cost} \subseteq 2^{\Gamma}.$$

Define

$$\vartheta \;:\; \mathbb{N} \;\to\; \mathbb{N} \qquad \text{such that} \qquad \vartheta(m) = \#K + \#P + m,$$

so that for each Segment $S_i$, we have $S_i \equiv X_{\vartheta(i)}$. Then, we assume a model where observing a Segment $S_i$ is associated with an exploring cost $\alpha_{cost}^{\{\vartheta(i)\}}$, and also where observing two sibling Segments $S_i, S_j$ is associated with some savings $\alpha_{cost}^{\{\vartheta(i),\vartheta(j)\}}$. That is, exploring only $S_i$ has a total cost of

$$\alpha_{cost}^{\varnothing} + \alpha_{cost}^{\{\vartheta(i)\}},$$

and exploring both $S_i$ and $S_j$ has a total cost of

$$\alpha_{cost}^{\varnothing} + \alpha_{cost}^{\{\vartheta(i)\}} + \alpha_{cost}^{\{\vartheta(j)\}} - \alpha_{cost}^{\{\vartheta(i),\vartheta(j)\}}.$$

Correspondingly, observing $S_i = 1$ is associated with some gross income $\alpha_{inc}^{\{\vartheta(i)\}}$, and due to savings we associate some extra gross income $\alpha_{cost}^{\{\vartheta(i),\vartheta(j)\}}$ to observing $S_i = S_j = 1$ for two sibling Segments $S_i, S_j$. That is, we assume that there is a discount effect corresponding to the costs of extracting the gain from two sibling Segments, which results in a higher gross income. This, and also the savings for the cost function, could represent savings caused by a partly shared infrastructure. However, we also have to ensure that no combination of any size corresponds to a free exploration. Higher order interactions are assumed not to be present, for simplicity.

We have assumed that for each Segment $S_i$

$$\alpha_{cost}^{\{\vartheta(i)\}},\ \alpha_{inc}^{\{\vartheta(i)\}} \;\geqslant\; 0,$$

and also, that for each pair of sibling Segments $S_i, S_j$

$$\min\{\alpha_{cost}^{\{\vartheta(i)\}}, \alpha_{cost}^{\{\vartheta(j)\}}\} \geqslant -\alpha_{cost}^{\{\vartheta(i),\vartheta(j)\}} \geqslant 0, \qquad \alpha_{inc}^{\{\vartheta(i),\vartheta(j)\}} \;\geqslant\; 0.$$

The model assumes that cost and income value interactions occur only between Segments $S_i, S_j$ that are siblings in the graph representation, i.e. children of the same Prospect $P_k$,

$$\{S_i,\ S_j\} \subseteq \mathrm{Ch}(P_k).$$

That is, we set

$$T_{inc} = T_{cost} = \big\{\varnothing\big\} \cup \big\{\{i\} \mid i \in \Gamma\big\} \cup \big\{\{i,j\} \mid i,j \in \Gamma,\ \mathrm{Pa}(X_i) = \mathrm{Pa}(X_j)\big\}.$$

Without loss of generality, we set $\alpha_{inc}^{\varnothing} = \alpha_{cost}^{\varnothing} = 1$. Then, for each Segment $S_i$, the coefficient $\alpha_{cost}^{\{\vartheta(i)\}}$ is drawn from a Gamma distribution with mean $\mu_{cost}$ and standard deviation $\sigma_{cost}$, and correspondingly, the coefficient $\alpha_{inc}^{\{\vartheta(i)\}}$ is drawn from a Gamma distribution with mean $\mu_{inc}$ and standard deviation $\sigma_{inc}$. That is,

$$\alpha_{cost}^{\{\vartheta(i)\}} \sim \mathrm{Gamma}(\mu_{cost}, \sigma_{cost}),$$

and

$$\alpha_{inc}^{\{\vartheta(i)\}} \sim \mathrm{Gamma}(\mu_{inc}, \sigma_{inc}).$$

Then, for each pair of sibling Segments $S_i, S_j$ with $k$ other Segment siblings, we assume

$$u_{i,j} \sim \frac{1}{k+1}\mathrm{Unif}\left(0,1\right),$$

and set

$$\alpha_{cost}^{\{\vartheta(i),\vartheta(j)\}} = -u_{i,j} \cdot \min\{\alpha_{cost}^{\{\vartheta(i)\}}, \alpha_{cost}^{\{\vartheta(j)\}}\}$$

and we also draw $\alpha_{inc}^{\{\vartheta(i),\vartheta(j)\}}$ from a Gamma distribution, according to

$$\alpha_{inc}^{\{\vartheta(i),\vartheta(j)\}} \sim \mathrm{Gamma}\left(\frac{\mu_{inc}}{10}, \frac{\sigma_{inc}}{10}\right).$$

## 8 First Set of Tests: BN1

The first set of tests were done according to Method 1, as introduced in Section 6, on a test case generated as described in Section 7. We represent the income function $w(X,Z)$ as the exponential of a pseudo-Boolean function, by use of formulas from Section 3.2. Combining this with the representation of the probability distribution, we get a representation for $w(X,Z)\mathbb{P}(X)$. Then, we calculate upper and lower bounds, respectively, on $\mathbb{E}w(X,Z)$ as in (6.7), by use of "The AMRF Programs" introduced in Section 4. This computer program sums out the Random Variables according to the numbering of the nodes. The first Random Variable to be summed out is $K_1$ and the last is $S_{\#S}$. From the resulting bound functions, we try to rule out candidates for the optimal decision, as described in Section 6.

We will present a Bayesian Network with 4 Kitchens, 7 Prospects and 10 Segments. A visualization of the graph can be seen in Figure 7. Corresponding

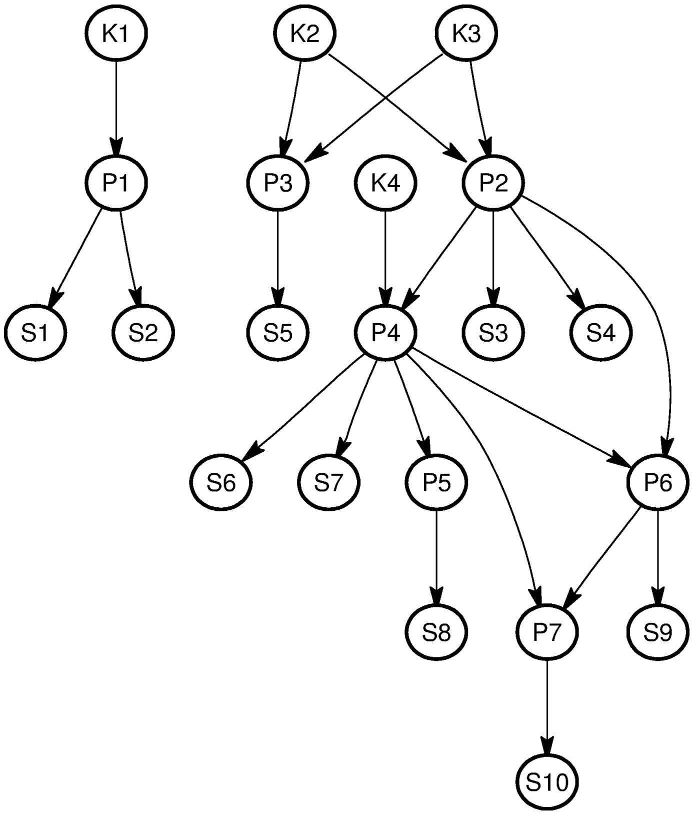

Figure 7: BN1, the Bayesian Network for first set of tests

to this graph, we have a Joint Probability Distribution, which is characterized by the numbers in Table 14 in Section C.1. This graph, with its corresponding Joint Probability Distribution, will be named BN1. Two sets consisting of a cost function $c(Z)$ and an income function $w(X, Z)$, are also generated, and can be found in Tables 15 and 16, respectively, also in Section C.1. For the first set, denoted by $a$, the parameters of the distribution for generating the coefficients were set to

$$\mu_{\text{cost},a} = 2000, \qquad \sigma_{\text{cost},a} = 2000, \qquad \mu_{\text{inc},a} = 4500, \qquad \sigma_{\text{inc},a} = 1200.$$

Correspondingly, for the second set, denoted by $b$, the parameters were set to

$$\mu_{\text{cost},b} = 4500, \qquad \sigma_{\text{cost},b} = 4000, \qquad \mu_{\text{inc},b} = 4500, \qquad \sigma_{\text{inc},b} = 4000.$$

When we refer to BN1a, we think of BN1 with the cost and income functions denoted by $a$, and correspondingly for BN1b. The parameters for drawing the Joint Probability Distribution were set to

$$\alpha_{K_i} = 6.0, \qquad \beta_{K_i} = 2.0, \qquad \alpha_{e_{i,j}} = 5.0, \qquad \beta_{e_{i,j}} = 2.0,$$

and the $\epsilon$ parameter was given the value $\epsilon = 0.01$. Recall that the Joint Probability Distribution is exactly the same for test $a$ and $b$.

Given a cost function and an income function, we calculate bounds on $f(Z)$ corresponding to an input constant $\nu$ for "The AMRF Programs". Actually, we calculate bounds for each

$$\nu \in \{11, 12, 13, 14, 15, 16, 17, 18, 19\}, \tag{8.1}$$

and thus we get nine pairs consisting of upper bound functions $f^+$ and lower bound functions $f^-$, respectively, for each set $a, b$ of cost function and income function. We let $f^+_{\nu,i}$ denote the upper bound function obtained for the given value of $\nu$ for cost and income function $i$, and correspondingly for $f^-_{\nu,i}$. That is, $\nu$ is as in (8.1), and $i \in \{a, b\}$. For each pair $f^+_{\nu,i}, f^-_{\nu,i}$, we evaluate the values $f^+_{\nu,i}(Z), f^-_{\nu,i}(Z)$ for each

$$Z \in \Omega^{\Gamma,u} \qquad \text{for} \qquad u = 5.$$

That is, we evaluate the upper and lower bounds for each decision that involves observing a maximum of five Segments. For each pair $(\nu, i)$, the resulting number of candidates for the optimal decision can be found in Tables 1 and 2 for $a$ and $b$, respectively.

| Decision size | Total number of decisions | Number of accepted decisions for $f^+_{\nu,a}, f^-_{\nu,a}$ for $\nu =$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 10 | 6 | 6 | 6 | 3 | 2 | 1 | 0 | 0 | 0 |
| 2 | 45 | 40 | 42 | 39 | 35 | 30 | 24 | 17 | 9 | 0 |
| 3 | 120 | 119 | 119 | 116 | 111 | 100 | 85 | 64 | 36 | 0 |
| 4 | 210 | 210 | 210 | 209 | 205 | 196 | 175 | 140 | 84 | 0 |
| 5 | 252 | 252 | 252 | 252 | 251 | 246 | 232 | 197 | 127 | 1 |
| Sum | 638 | 627 | 629 | 622 | 608 | 574 | 517 | 388 | 256 | 1 |

Table 1: Number of accepted decisions as a function of $\nu$ for the income function and cost function from Table 15.

| Decision size | Total number of decisions | Number of accepted decisions for $f^+_{\nu,b}, f^-_{\nu,b}$ for $\nu =$ | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 |
| 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 10 | 6 | 7 | 7 | 3 | 2 | 1 | 0 | 0 | 0 |
| 2 | 45 | 41 | 42 | 39 | 36 | 31 | 25 | 18 | 10 | 1 |
| 3 | 120 | 119 | 119 | 116 | 110 | 100 | 85 | 64 | 36 | 0 |
| 4 | 210 | 210 | 210 | 209 | 205 | 195 | 175 | 140 | 84 | 0 |
| 5 | 252 | 252 | 252 | 252 | 251 | 246 | 231 | 196 | 126 | 0 |
| Sum | 638 | 629 | 630 | 623 | 605 | 517 | 418 | 388 | 256 | 1 |

Table 2: Number of accepted decisions as a function of $\nu$ for the income function and cost function from Table 16.

Note that even though this is a quite small graph, we need $\nu = 19$ to find the optimal decision, that is, for the bounds $f^+_{\nu,i}, f^-_{\nu,i}$ to leave out only one candidate. This is also the value of $\nu$ that ensures exact calculation of $f(Z)$. That is, $\nu_{exact} = 19$ corresponds to no approximations for this graph, and we get

$$f^+_{19,i} \equiv f^-_{19,i} \qquad \text{for} \qquad i = a, b.$$

Also note that exact calculation will only leave candidates that actually are optimal decisions. That is, if we are left with more than one decision after exact calculations, these must all correspond to the same value of $f$, that is, the maximal value of $f$ within $\Omega^{\Gamma,u}$. Thus, all decisions left are optimal. Hence, needing $\nu = 19$ to separate out one candidate is actually the worst case for this graph. Also, note that $\nu = 18$ leaves 256 candidates for the optimal decision in both cases $i = 1, 2$, from a total of 638 decisions in $\Omega^{\Gamma,u}$. That is, reducing the value of $\nu$ with 1, and thus reducing the asymptotic complexity in run time from $\mathcal{O}\left(2^{\nu_{exact}}\right)$ to $\mathcal{O}\left(2^{\nu_{exact}-1}\right)$, leaves us with roughly half of our initial decision space still. A further reduction of the value of $\nu$ leads to at least 418 candidates in all cases, which corresponds to a reduction by less than 35% of the candidates. That is, we observe that we need a quite large value of $\nu$ to get useful results for these examples.

Also note that for the first test, denoted $a$, the optimal decision implies observing 5 nodes. In the next, $b$, the costs are higher, which results in an optimal decision with only 3 observations.

The examples presented illustrates the general observations from tests according to Method 1. As soon as approximations were made on our function $w(X, Z)\mathbb{P}(X)$, we end up with a large fraction of the initial decision space $\Omega^{\Gamma,u}$ left. Thus, we had to come up with a revised strategy to be able to find the optimal decision more effectively on larger graphs.

# 9 Second Set of Tests

The tests of Method 1 indicated that we needed to rethink our strategy for finding the optimal decision. First of all, we had to look for any unnecessary approximations. Observe that we can take advantage of (2.7) in Section 2.1. That is, previously, when we calculated the expected value of the function $w(X, Z)$, we took the expected value with respect to the full Joint Probability Distribution. That is, if the biggest maximal clique in the graph had a size greater than the current value of $\nu$ in the calculations, in general, the approximations made would introduce error terms for the whole graph.

However, the form we assume for the cost and income functions, presented in Section 7.3, allows us to divide the cost and income functions into parts corresponding to each Prospect. Let $X_{B_j}$ denote Segment children of Prospect $P_j$, that is

$$X_{B_j} = \{S_i \in \mathrm{Ch}(P_j) \; | \vartheta(i) \in \Gamma\},$$

and also, let $Z_{B_j}$ denote the corresponding entries in the decision vector. This allows us to write the cost function as

$$c(Z) = \sum_{j=1}^{\#P} c_j(Z_{B_j})$$

and the income function as

$$w(X,Z) = \sum_{j=1}^{\#P} w_j(X_{B_j}, Z_{B_j}).$$

In fact, let

$$T_k = \big\{\varnothing\big\} \cup \big\{\{i\} | i \in \Gamma,\; X_i \in \mathrm{Ch}(P_k)\big\} \cup \big\{\{i,j\} | i,j \in \Gamma,\; X_i, X_j \in \mathrm{Ch}(P_k)\big\}.$$

Then,

$$T_{inc} = T_{cost} = \cup_{j=1}^{\#P} T_j,$$

and for positive integers $j, k \leqslant \#P$

$$j \neq k \qquad \Leftrightarrow \qquad T_k \cap T_j = \{\varnothing\}.$$

Thus,

$$w(X,Z) + (\#P - 1)\alpha_{inc}^{\varnothing} = \sum_{j=1}^{\#P} \sum_{\lambda \in T_j} \alpha_{inc}^{\lambda} \prod_{k \in \lambda} X_k Z_k,$$

and

$$c(Z) + (\#P - 1)\alpha_{cost}^{\varnothing} = \sum_{j=1}^{\#P} \sum_{\lambda \in T_j} \alpha_{cost}^{\lambda} \prod_{k \in \lambda} Z_k.$$

If we ignore the multiples of $\alpha_{cost}^{\varnothing}$ and $\alpha_{inc}^{\varnothing}$ added, respectively, we get

$$w_j(X_{B_j}, Z_{B_j}) = \sum_{\lambda \in T_j} \alpha_{inc}^{\lambda} \prod_{k \in \lambda} Z_k$$

and

$$c_j(Z_{B_j}) = \sum_{\lambda \in T_j} \alpha_{cost}^{\lambda} \prod_{k \in \lambda} Z_k.$$

Also note that our choice of

$$\alpha_{cost}^{\varnothing} = \alpha_{inc}^{\varnothing} = 1$$

actually implies

$$w(X, Z) - c(Z) \equiv \sum_{j=1}^{\#P} \left( w_j(X_{B_j}, Z_{B_j}) - c_j(Z_{B_j}) \right).$$

Thus,

$$\begin{aligned} f(Z) &= \mathbb{E}\left(w(X, Z) - c(Z)\right) \\ &= \mathbb{E}\left( \sum_{j=1}^{\#P} \left( w_j(X_{B_j}, Z_{B_j}) - c_j(Z_{B_j}) \right) \right) \\ &= \sum_{j=1}^{\#P} \mathbb{E}\left( w_j(X_{B_j}, Z_{B_j}) - c_j(Z_{B_j}) \right). \end{aligned} \tag{9.1}$$

Assume that $j \leqslant \#P$ is a positive integer. It is when calculating

$$\mathbb{E}\left( w_j(X_{B_j}, Z_{B_j}) - c_j(Z_{B_j}) \right),$$

we get to use the observations from (2.7). Actually, let

$$A_j = \{P_j\} \cup \operatorname{Anc}(P_j) \cup B_j,$$

and observe that $A_j$ meets the requirements set in (2.2). Thus,

$$\mathbb{E}\left( w_j(X_{B_j}, Z_{B_j}) - c_j(Z_{B_j}) \right) = \sum_{X_i \in A_j} \left( w_j(X_{B_j}, Z_{B_j}) - c_j(Z_{B_j}) \right) \mathbb{P}(X_{A_j}).$$

Let

$$M_j = 1 + \max_{Z_{B_j}} c_j(Z_{B_j}),$$

define

$$S_j = \left\{ \lambda \mid \{\tilde{X}_i \mid i \in \lambda\} \in 2^{X_{B_j} \cup Z_{B_j}} \right\},$$

and let $\{\beta_j^\lambda\}_{\lambda \in S_j}$ be constants such that

$$0 < w_j(X_{B_j}, Z_{B_j}) - c_j(Z_{B_j}) + M_j = \exp\left(\sum_{\lambda \in S_j} \beta_j^\lambda \prod_{k \in \lambda} \tilde{X}_k\right).$$

Note that we identify each variable $\tilde{X}_i$ with an entry in $X$ or $Z$, respectively, according to (6.6).

Finally, let

$$M = \sum_{j=1}^{\#P} M_j,$$

and note that (9.1) implies

$$\begin{aligned} f(Z) + M &= \mathbb{E}\left(w(X,Z) - c(Z) + M\right) \\ &= \mathbb{E}\left(\sum_{j=1}^{\#P} \left(w_j(X_{B_j}, Z_{B_j}) - c_j(Z_{B_j}) + M_j\right)\right) \\ &= \sum_{j=1}^{\#P} \mathbb{E}\left(w_j(X_{B_j}, Z_{B_j}) - c_j(Z_{B_j}) + M_j\right) \\ &= \sum_{j=1}^{\#P} \left(\sum_{X_i \in A_j} \mathbb{P}(X_{A_j}) \exp\left(\sum_{\lambda \in S_j} \beta_j^\lambda \prod_{k \in \lambda} \tilde{X}_k\right)\right). \end{aligned}$$

Note that the core idea of Method 2 was introduced by adding the constants $M_j$. Also note that, in practice, the above equation tells us to calculate $f(Z) + M$ by adding the contributions corresponding to each prospect. That is, each of those terms are obtained similarly as for Method 1 in Section 8, but now from the distribution $\mathbb{P}(X_{A_j})$ and the function

$$w_j(X_{B_j}, Z_{B_j}) - c_j(Z_{B_j}) + M_j.$$

Also note that for any positive integer $j \leqslant \#P$,

$$w_j(X_{B_j}, \vec{0}) = \alpha_{inc}^{\varnothing} = 1 = \alpha_{cost}^{\varnothing} = c_j(\vec{0}),$$

and thus

$$f(\vec{0}) = 0.$$

The tests for Method 2 show that we need the graphs to be sparse for our upper and lower bounds to be tight enough to eliminate a significant fraction of the possible decisions $Z \in \Omega^{\Gamma,u}$. This section presents tests where the approximations work well, but it also presents a description of how it fails for an example Bayesian Network. Also in these tests, "The AMRF Programs" sum out the Random Variables according to our numbering of the nodes, and the numbers we present for each test is as before. The strategy for finding the optimal decision is still the same, we just calculate the bounds in a slightly different way.

## 9.1 Sparse graphs: BN2-BN4

First, a set of tests on sparse graphs will be presented. Here, this means that the number of edges in the graph is kept close to the minimum according to our definitions in Section 7. These graphs will encode more conditional independence assumptions than if more edges were added, and the clique sizes in the corresponding undirected graph will be smaller than for a graph with more edges added. That means that "The AMRF Programs" will do less approximations, and it is more likely that our strategy will work as we hope.

Three single Bayesian Networks of different sizes will be presented. For each of these, we will study the number of accepted decisions $Z$ as a function of the approximation constant $\nu$. To help the discussion of these numbers, we also present the value of $\nu = \nu_{exact}(P_i)$ that ensures exact calculation of the term of $f(Z)$ corresponding to Prospect $P_i$.

For each of these three examples, the test is repeated 1000 times. That is, for each example, 1000 Bayesian Networks with cost function and income function are generated from the same distribution as for the corresponding example. From each of these replications, we will study the value of $\nu$ needed to find the optimal decision by our strategy, that is

$$\nu_{one} = \min_{\nu}\{\nu \mid \text{Number of accepted decisions}(\nu) = 1\}$$

We will also look at the value of $\nu$ needed to separate out just a few candidates. In fact, we study

$$\nu_{acceptable} = \min_{\nu}\{\nu \mid \text{Number of accepted decisions}(\nu) \leqslant 7\}.$$

These two numbers will be compared to the value of $\nu$ that ensures exact calculation of the expected gross income $f(Z)$ for the graph, that is

$$\nu_{exact} = \max_{\text{Prospect } P_i} \nu_{exact}(P_i).$$

Then also, for a given value of $\nu$, and corresponding upper bounds $f^+$ and lower bounds $f^-$, respectively, let $\tilde{Z}$ be the decision with the highest lower bound for expected income. That is,

$$\tilde{Z} = \arg \max_{Z \in \Omega^{\Gamma,u}} f^-(Z).$$

Define

$$L_\nu = \frac{f(Z^*) - f(\tilde{Z})}{f(Z^*)},$$

the percentage loss in exact expected income, when following the approximate optimal decision $\tilde{Z}$ instead of the (exact) optimal decision $Z^*$. In each of the three sets of 1000 tests, the value of $\nu$ corresponding to the loss $L_\nu$ is chosen so that we expected (according to our best guesses) that about half of the tests in the set would have $\nu_{exact} > \nu$. That is, so that our program would have to make a guess for the optimal decision in about half the cases.

### 9.1.1 BN2

The following test describes a Bayesian Network with 7 Kitchens, 15 Prospects and 40 Segments. The number of extra edges added after the minimum value of 55, is 10. That is, the result is a graph with 62 nodes and 65 edges, and it is visualized in Figure 8. A corresponding visualization where only the Kitchens and the Prospects are shown, can be found in Figure 9. Observe that the undirected version of this Bayesian Network consists of two connected components, that is, the nodes $\{K2, P2, D2\}$ do not have edges to or from any of the other nodes not in this set. The Bayesian Network with corresponding cost and income functions is generated as described in Section 7. The parameters for drawing cost and income coefficients were set to

$$\mu_{\text{cost}} = 4000, \qquad \sigma^2_{\text{cost}} = 2500, \qquad \mu_{\text{inc}} = 5000, \qquad \sigma^2_{\text{inc}} = 2500.$$

The parameters for drawing the Joint Probability Distribution were set to

$$\alpha_{K_i} = 6.0, \qquad \beta_{K_i} = 2.0, \qquad \alpha_{e_{i,j}} = 5.0, \qquad \beta_{e_{i,j}} = 2.0,$$

and the $\epsilon$ parameter was given the value $\epsilon = 0.1$. For numbers fully describing the probability distribution and the cost and income functions, see Tables 17-32 in Appendix C.2.

Observe from Table 3 that we need $\nu = 11$ to get exact numbers, but Table 4 tells us that already from $\nu = 6$, we get only one candidate for the optimal

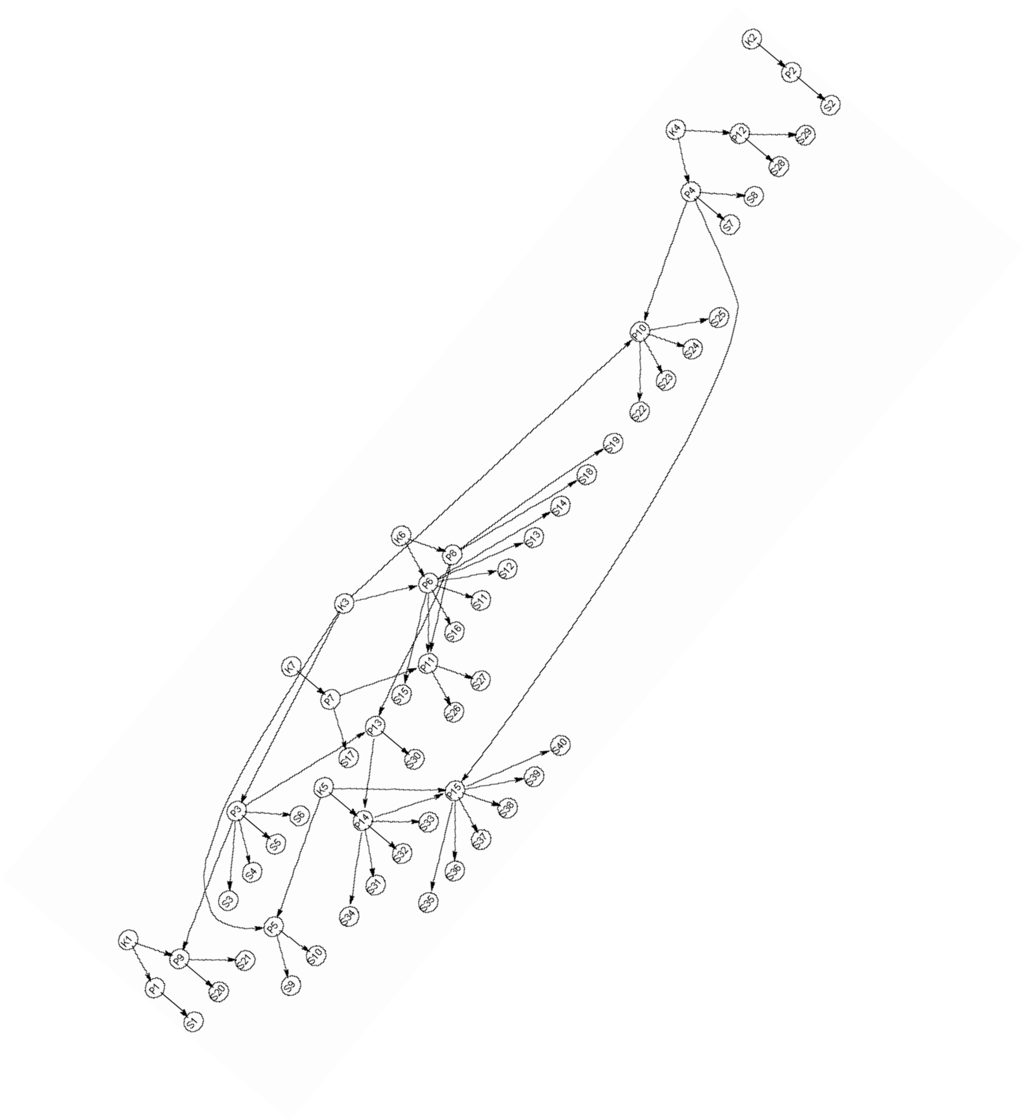

Figure 8: The Bayesian Network 2BN.

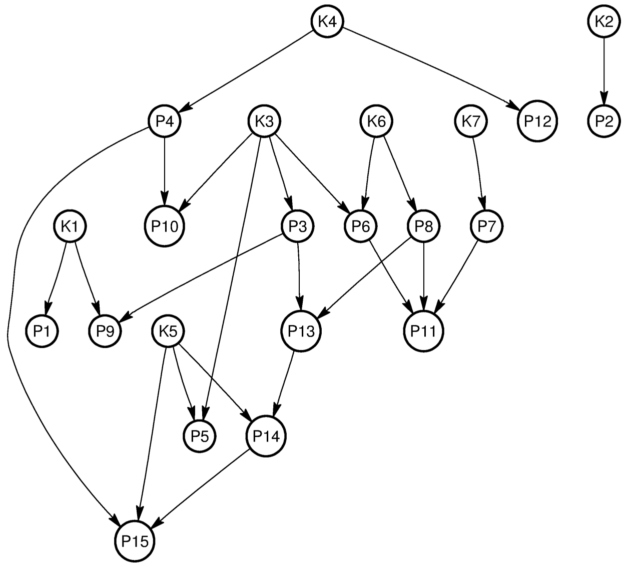

Figure 9: BN2: A visualization of only the Kitchens, the Prospects and the edges between them.

| Prospect $P_i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\nu$ | 1 | 1 | 7 | 3 | 3 | 11 | 1 | 3 | 3 | 7 | 3 | 3 | 2 | 7 | 11 |

Table 3: BN2: Value of $\nu$ that ensures exact calculation of expected gross income function $w_i - c_i + M_i$ corresponding to prospect $P_i$ for the Bayesian Network in Figure 8.

| Decision size | Total number of decisions | Number of accepted decisions for $\nu =$ | | | |
|---|---|---|---|---|---|
| | | $1-3$ | 4 | 5 | $6-11$ |
| 0 | 1 | 1 | 1 | 0 | 0 |
| 1 | 40 | 40 | 40 | 0 | 0 |
| 2 | 780 | 780 | 770 | 1 | 1 |
| 3 | 9880 | 9880 | 9336 | 8 | 0 |
| 4 | 91390 | 91390 | 82081 | 28 | 0 |
| 5 | 658008 | 658008 | 567072 | 58 | 0 |
| Sum | 760099 | 760099 | 65930 | 95 | 1 |

Table 4: BN2: Number of accepted decisions as a function of $\nu$.

decision. Even for $\nu = 5$, we get pretty good results. However, for $\nu \leqslant 3$, we are not able to eliminate any decisions, and $\nu = 4$ only allows us to eliminate 13% of the decisions. Observe from Table 3, that $3 \leqslant \nu \leqslant 6$ all introduces approximations for the functions corresponding to Prospects $P_3, P_6, P_{10}, P_{14}$ and $P_{15}$, so the stepwise improvements of the results from $\nu = 3$ to $\nu = 6$ are all due to better approximations, and not to approximations on a smaller number of functions.

This test was replicated 1000 times. That is, 1000 Bayesian Networks with 7 Kitchens, 15 Prospects and 40 Segments were generated, with Joint Probability Distributions and cost and income functions drawn from the same distributions as those for BN2. Recall from Section 9.1, that $\nu_{exact}$ is the smallest value of the input constant $\nu$ that ensures "The AMRF Programs" to do exact calculations for each replication. Correspondingly, $\nu_{acceptable}$ is the smallest value of $\nu$ that leaves us with a maximum of 7 candidates for the optimal decision, and $\nu_{one}$ is the smallest value of $\nu$ that leaves us with only the optimal decision $Z*$. The results of the 1000 replications are summed up in Figures 10, 11, 13, 12 and 14.

Observe from Figure 10, how the values for $\nu_{exact}$ are dominated by odd numbers. Actually, 931 tests have $\nu_{exact} \in \{7, 9, 11\}$. At the same time, Figure 11 tells us that for 70.5% of the tests, it is sufficient with $\nu = 6$ to separate out only one candidate for the optimal decision. In fact, 625 tests had $\nu_{one} \in \{5, 6\}$, while no tests had $\nu_{exact} \in \{5, 6\}$. Only one test had $\nu_{exact} = 4$, but actually, in 80 of the 1000 tests, $\nu = 4$ was sufficient to find the optimal decision. We observe from Figure 12 that 728 of the 1000 tests had a reduction of 3 or more from the value of $\nu_{exact}$ to the value of $\nu_{one}$. There is even a test where the

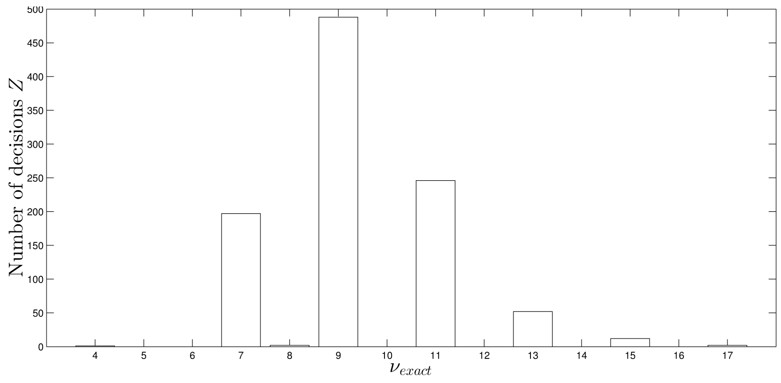

Figure 10: BN2: The value of $\nu$ needed to do exact calculations for the 1000 tests.

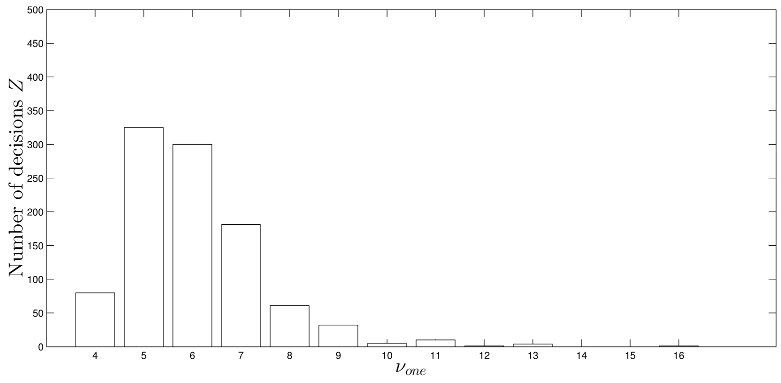

Figure 11: BN2: The value of $\nu$ needed to separate out only one candidate in each of the 1000 tests.

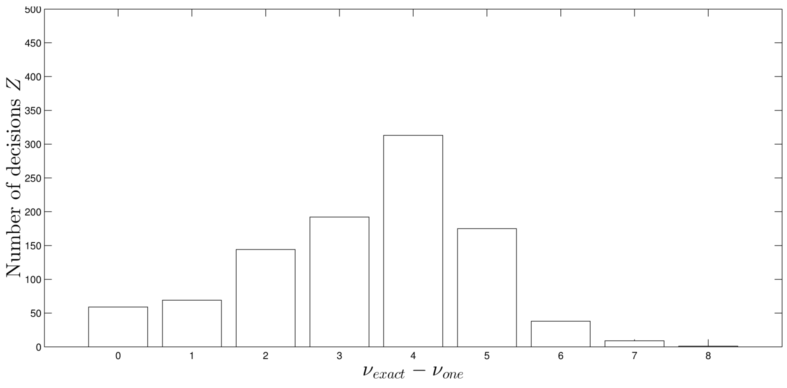

Figure 12: BN2: The difference between $\nu_{exact}$ and $\nu_{one}$ in each of the 1000 tests.

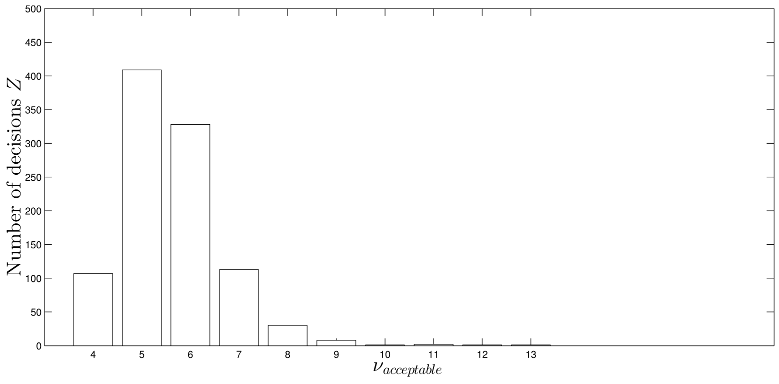

Figure 13: BN2: The value of $\nu$ needed to separate out a maximum of seven candidates in each of the 1000 tests.

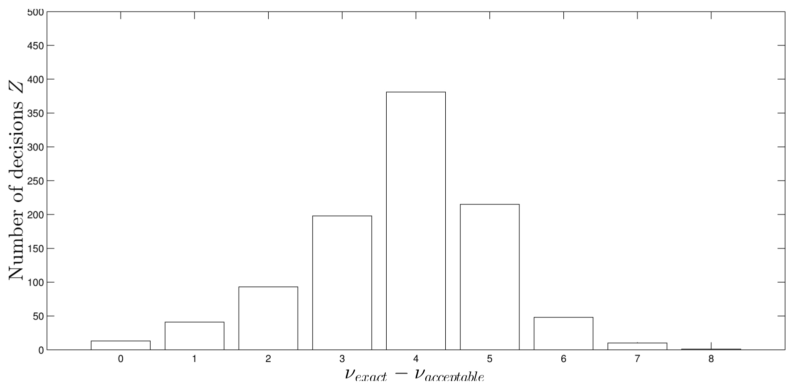

Figure 14: BN2: The difference between $\nu_{exact}$ and $\nu_{acceptable}$ in each of the 1000 tests.

value of this reduction is 8, and as much as 48 had a reduction of at least 6. Similarly, we observe from Figure 13, that for as much as 84.4% of the tests, it is sufficient with $\nu = 6$ to reduce the number of candidates for the optimal decision to seven or less. Actually, 737 tests had $\nu_{acceptable} \in \{5, 6\}$, and as much as 107 tests had $\nu_{acceptable} = 4$. Figure 12 shows how 853 of the 1000 tests had a reduction of 3 or more from $\nu_{exact}$ to the value of $\nu_{acceptable}$. As few as 13% of the tests had $\nu_{acceptable} = \nu_{exact}$, and 59 of the 1000 tests had a reduction of 6 or more.

For these 1000 replications, we chose $\nu = 6$, and studied the loss fraction $L_6$. We got 40 cases where the loss fraction $L_6$ was nonzero. These came from the in total 295 cases where $\nu = 6$ was not enough to leave only one candidate for the optimal decision. Of these 295 cases, where a guess of the optimal decision had to be made, the average of $L_6$ was 0.019, with a standard deviation of 0.081. The smallest nonzero loss fraction was $3.905 \cdot 10^{-5}$, and the greatest loss fraction was as high as 0.8318. Actually, there are five cases where the loss fraction is higher than 0.3, and thus we get a large standard deviation. For a listing of all nonzero loss fractions, the reader is referred to Table 33 in Section C.2. In total, on all 1000 cases, that left us with an average loss $\bar{L_6} = 0.0055$, with a standard deviation of 0.045.

| Prospect $P_i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $\nu$ | 3 | 7 | 3 | 3 | 1 | 3 | 1 | 3 | 1 | 3 |
| Prospect $P_i$ | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| $\nu$ | 3 | 3 | 1 | 3 | 3 | 1 | 3 | 3 | 1 | 3 |

Table 5: BN3: Value of $\nu$ that ensures exact calculation of expected gross income function $w_i - c_i + M_i$ corresponding to prospect $P_i$ for the Bayesian Network in Figure 15.

### 9.1.2 BN3

The following test describes a Bayesian Network with 10 Kitchens, 20 Prospects and 40 Segments. The number of extra edges added after the minimum value of 60, is 10. That is, the result is a graph with 70 nodes and 70 edges, and it is visualized in Figure 15. Observe that the undirected version of this Bayesian Network consists of four connected components. The nodes in the leftmost connected component is visualized in Figure 16, and correspondingly, we get a closer look on the three rightmost connected components in Figure 17. The Bayesian Network with corresponding cost and income functions is generated as described in Section 7. The parameters for drawing cost and income coefficients were set to

$$\mu_{\text{cost}} = 3000, \qquad \sigma^2_{\text{cost}} = 2500, \qquad \mu_{\text{inc}} = 5000, \qquad \sigma^2_{\text{inc}} = 2500.$$

The parameters for drawing the Joint Probability Distribution were set to

$$\alpha_{K_i} = 6.0, \qquad \beta_{K_i} = 2.0, \qquad \alpha_{e_{i,j}} = 5.0, \qquad \beta_{e_{i,j}} = 2.0,$$

and the $\epsilon$ parameter was given the value $\epsilon = 0.01$. For numbers fully describing the probability distribution and the cost and income functions, see Tables 34-54 in Appendix C.3.

Observe from Table 5 that we need $\nu = 7$ to get exact numbers, but Table 6 tells us that already from $\nu = 3$, we are able to eliminate 99.8% of the possible decisions. In addition, we only need $\nu \geqslant 5$ to get just one candidate for the optimal decision. Also for $\nu = 4$, we are able to remove most of the candidates for the optimal decision. Note from Table 5 that for $\nu \in \{1, 2\}$, we calculate exact values for the functions corresponding to $P_5, P_7, P_9, P_{13}, P_{16}$ and $P_{19}$, but that is not enough to eliminate any candidates. However, for $\nu \geqslant 3$, approximations are only made on the function corresponding to $P_2$, and as a result, the number

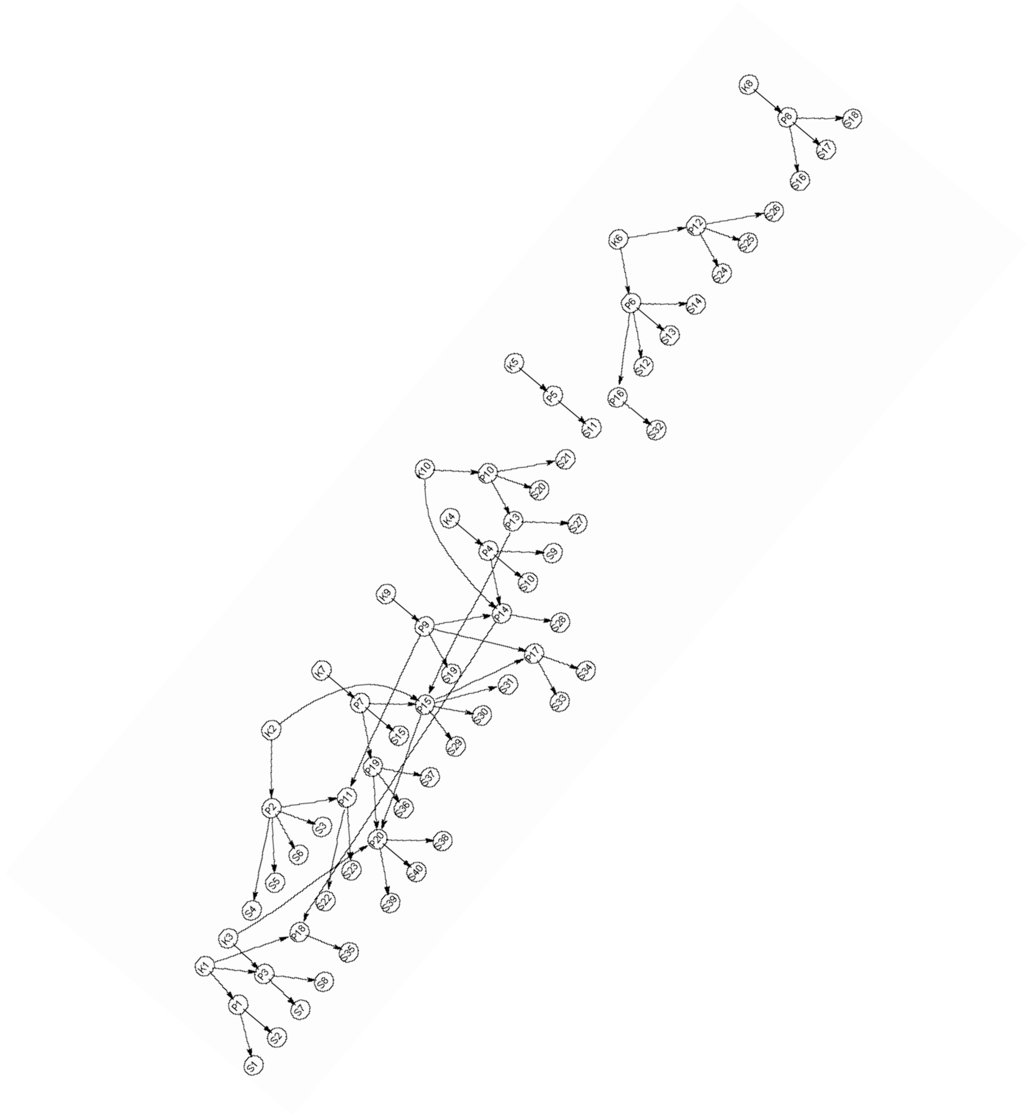
Figure 15: The Bayesian Network 3BN

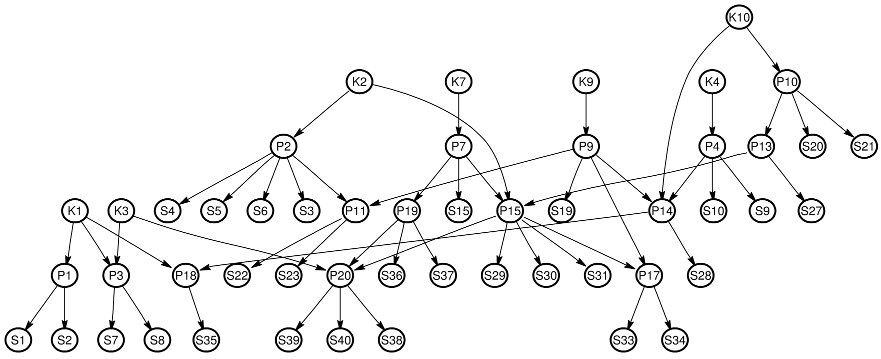

Figure 16: BN3: The left connected component of the Bayesian Network in Figure 15.

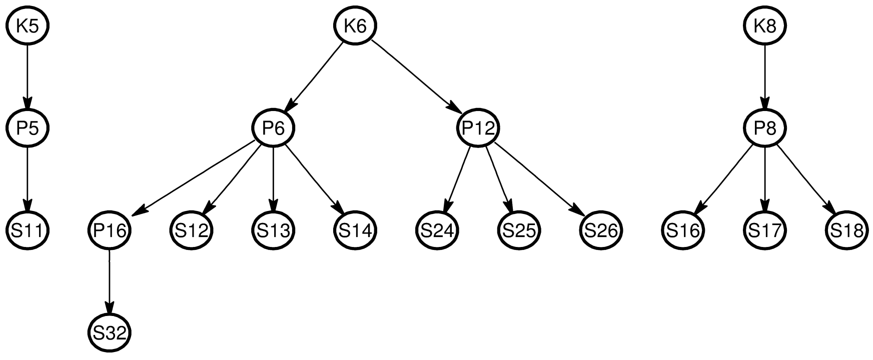

Figure 17: BN3: The right three connected components of the Bayesian Network in Figure 15.

| Decision | Total number | Number of accepted decisions for $\nu$ = | | | |
|---|---|---|---|---|---|
| size | of decisions | $1-2$ | 3 | 4 | $5-7$ |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 40 | 40 | 0 | 0 | 0 |
| 2 | 780 | 780 | 1 | 0 | 0 |
| 3 | 9880 | 9880 | 22 | 0 | 0 |
| 4 | 91390 | 91390 | 212 | 1 | 0 |
| 5 | 658008 | 658008 | 1331 | 4 | 1 |
| Sum | 760099 | 760099 | 1566 | 5 | 1 |

Table 6: BN3: Number of accepted decisions as a function of $\nu$.

of candidates are dropping rapidly from $\nu = 2$ to $\nu = 3$. However, we note that we still find the optimal decision for $\nu$ with value $\nu_{exact}(P_2) - 2$.

This test was replicated 1000 times, and the results are summed up in Figures 18, 19, 21, 20 and 22. Note that the Figures in this Section visualize the results for the same variables as the corresponding Figures in Section 9.1.1. Observe from Figure 18, how the values of $\nu_{exact}$ still are dominated by odd numbers. Actually, 814 tests have $\nu_{exact} \in \{7, 9\}$. At the same time, Figure 19 tells us that for 69.5% of the tests, it is sufficient with $\nu = 5$ to separate out only one candidate for the optimal decision. In fact, 782 tests had $\nu_{one} \in \{4, 5, 6\}$, while only 90 tests had $\nu_{exact} \in \{4, 5, 6\}$. We observe from Figure 20 that 541 of the 1000 tests had a reduction of 3 or more from the value of $\nu_{exact}$ to the value of $\nu_{one}$. 63.4% of the cases had a reduction of either $2, 3$ or 4. Also, 37 of the 1000 tests had a reduction of at least 5, but there are also 243 tests with no reduction. Similarly, we observe from Figure 21, that for as much as 80.5% of the tests, it is sufficient with $\nu = 5$ to reduce the number of candidates for the optimal decision to seven or less. Actually, 907 tests had $\nu_{acceptable} \in \{4, 5, 6\}$. Figure 20 shows how 651 of the 1000 tests had a reduction of 3 or more from the value of $\nu_{exact}$ to the value of $\nu_{acceptable}$. Also, 148 of the tests had $\nu_{acceptable} = \nu_{exact}$, a significant reduction from the 243 tests with $\nu_{one} = \nu_{exact}$. Also note that 45 of the tests had a reduction of 5 or more from the value of $\nu_{exact}$ to the value of $\nu_{acceptable}$.

For these 1000 tests, we chose $\nu = 5$, and got 27 cases where $L_5$ was nonzero. These came from the in total 305 cases where $\nu = 5$ was not enough to leave only one candidate for the optimal decision. Of these 305 cases, where a guess of the optimal decision had to be made, the average of $L_5$ was 0.0059, with a

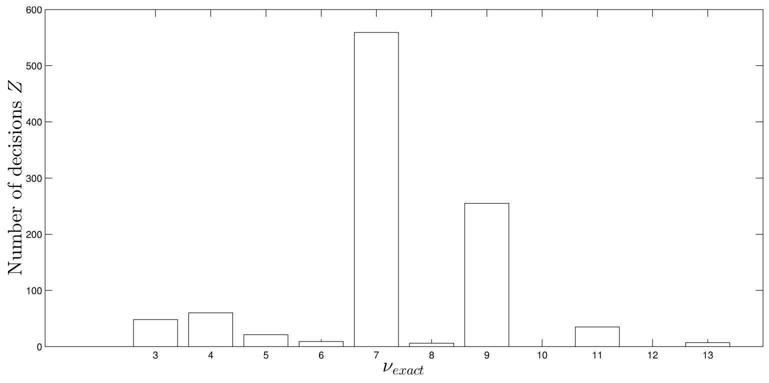

Figure 18: BN3: The value of $\nu$ needed to do exact calculations for the 1000 tests.

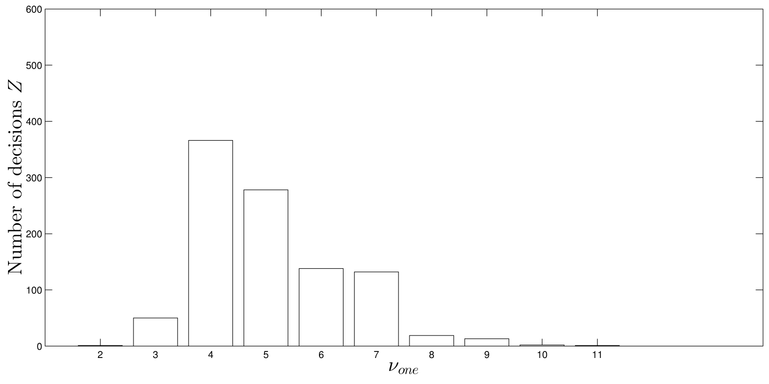

Figure 19: BN3: The value of $\nu$ needed to separate out only one candidate in each of the 1000 tests.

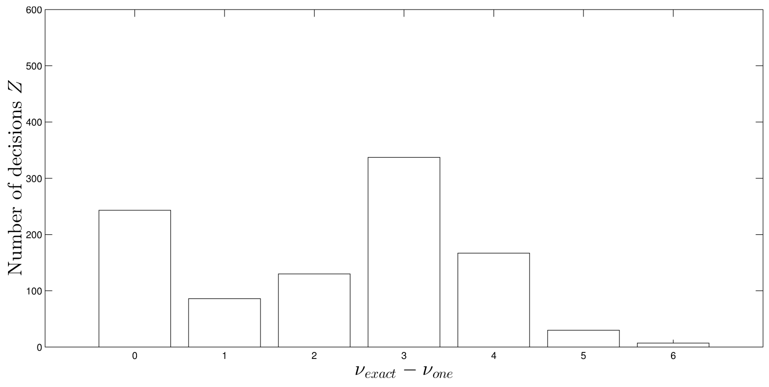

Figure 20: BN3: The difference between $\nu_{exact}$ and $\nu_{one}$ in each of the 1000 tests.

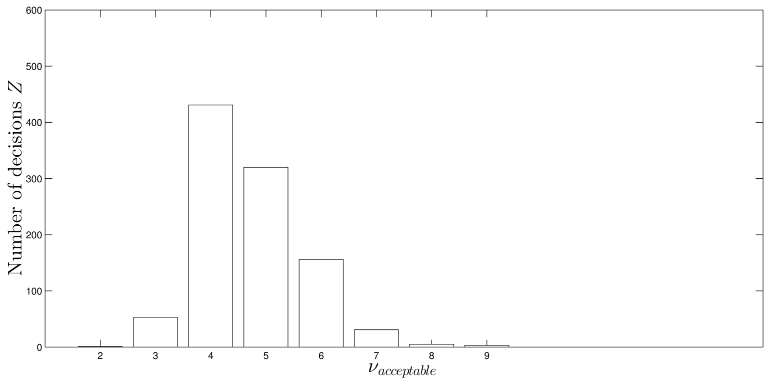

Figure 21: BN3: The value of $\nu$ needed to separate out a maximum of seven candidates in each of the 1000 tests.

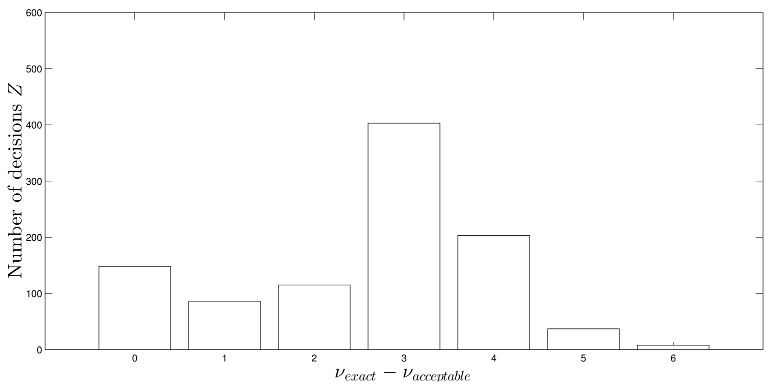

Figure 22: BN3: The difference between $\nu_{exact}$ and $\nu_{acceptable}$ in each of the 1000 tests.

standard deviation of 0.0354. The smallest nonzero loss fraction was $8.46 \cdot 10^{-4}$, and the greatest loss fraction was 0.4682. There was only three cases where the loss fraction was higher than 5%. For a listing of all nonzero loss fractions, the reader is referred to Table 55 in Section C.3. In total, on all 1000 cases, that left us with an average loss $\bar{L_5} = 0.0018$, with a standard deviation of 0.0197.

### 9.1.3 BN4

The following test describes a Bayesian Network with 10 Kitchens, 20 Prospects and 60 Segments. The number of extra edges added after the minimum value of 60, is 20. That is, the result is a graph with 90 nodes and 80 edges, and it is visualized in Figure 23. A corresponding visualization where only the Kitchens and the Prospects are shown, can be found in Figure 24. Observe that the undirected version of this Bayesian Network is fully connected. That is, assuming all edges undirected, it is possible to find a path from any node to any other node. The Bayesian Network with corresponding cost and income functions is generated as described in Section 7. The parameters for drawing cost and income coefficients were set to

$$\mu_{\text{cost}} = 4000, \qquad \sigma^2_{\text{cost}} = 2500, \qquad \mu_{\text{inc}} = 5000, \qquad \sigma^2_{\text{inc}} = 2500.$$

The parameters for drawing the Joint Probability Distribution were set to

$$\alpha_{K_i} = 6.0, \qquad \beta_{K_i} = 2.0, \qquad \alpha_{e_{i,j}} = 5.0, \qquad \beta_{e_{i,j}} = 2.0,$$

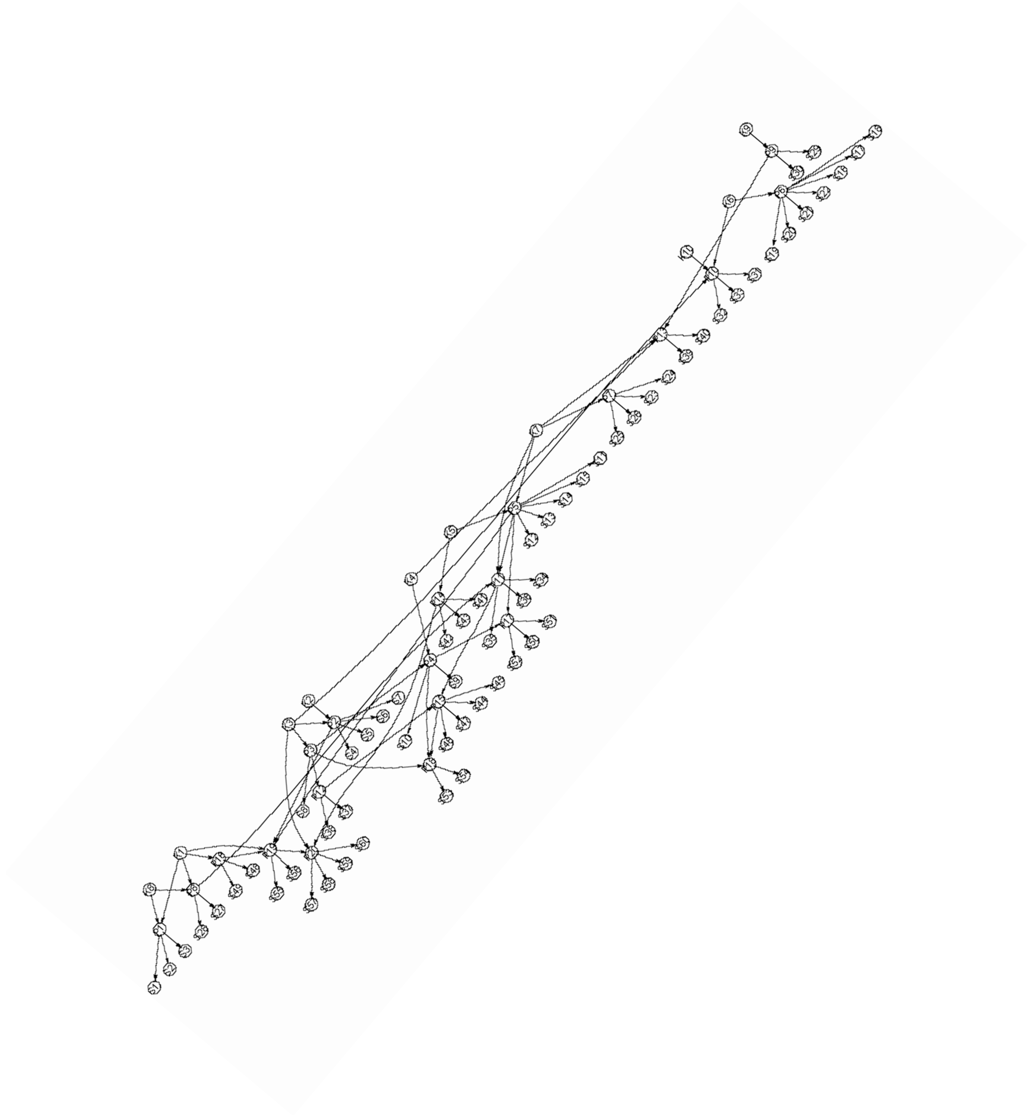
Figure 23: The Bayesian Network 4BN.

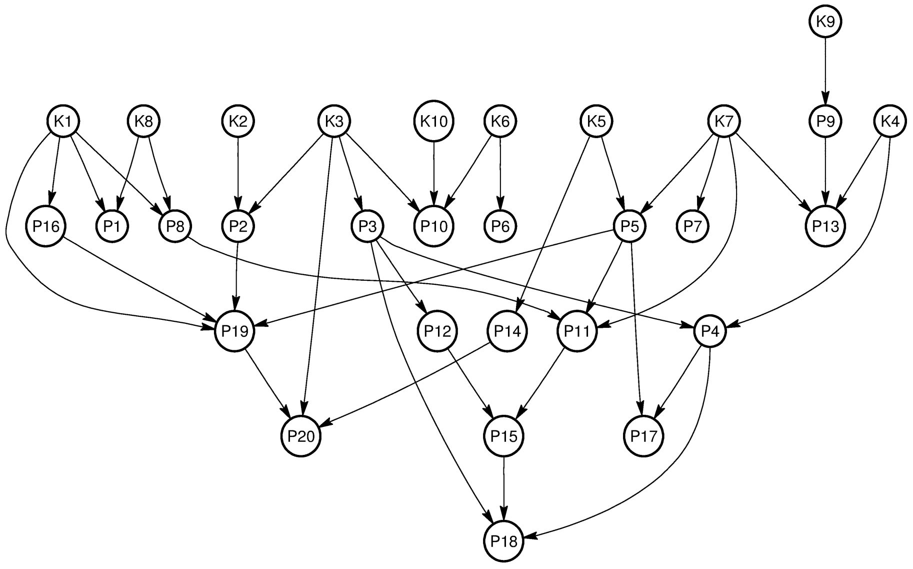

Figure 24: BN4: A visualization of only the Kitchens, the Prospects and the edges between them.

| Prospect $P_i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| $\nu$ | 3 | 7 | 1 | 3 | 9 | 13 | 7 | 3 | 3 | 3 |
| Prospect $P_i$ | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
| $\nu$ | 3 | 3 | 3 | 3 | 7 | 3 | 3 | 4 | 4 | 7 |

Table 7: BN4: Value of $\nu$ that ensures exact calculation of expected gross income function $w_i - c_i + M_i$ corresponding to prospect $P_i$ for the Bayesian Network in Figure 23.

| Decision | Total number | Number of accepted decisions for $\nu =$ | | | |
|---|---|---|---|---|---|
| size | of decisions | $1-6$ | 7 | 8 | $9-13$ |
| 0 | 1 | 1 | 0 | 0 | 0 |
| 1 | 60 | 60 | 0 | 0 | 0 |
| 2 | 1770 | 1770 | 0 | 0 | 0 |
| 3 | 34220 | 34220 | 2 | 2 | 1 |
| 4 | 487635 | 487635 | 2 | 1 | 0 |
| Sum | 523686 | 523686 | 4 | 3 | 1 |

Table 8: BN4: Number of accepted decisions as a function of $\nu$.

and the $\epsilon$ parameter was given the value $\epsilon = 0.01$. For numbers fully describing the probability distribution and the cost and income functions, see Tables 56-76 in Appendix C.4.

Observe from Table 7 that we need $\nu = 13$ to get exact numbers, but Table 8 tells us that already from $\nu = 7$, we are able to eliminate a great fraction of the possible decisions. In addition, we only need $\nu \geqslant 9$ to eliminate all non optimal decisions. Observe from Table 7 that for $\nu \geqslant 7$, we calculate exact values for the functions corresponding to all Prospects except $P_5$ and $P_6$. Also, for $\nu \geqslant 9$, approximations are only made on the function corresponding to $P_6$, and these are the same $\nu$s that allows to eliminate all non optimal decisions.

This test was replicated 1000 times, and the results are summed up in Figures 25, 26, 28, 27 and 29. Note that the Figures in this Section visualizes the same variables as the corresponding Figures in Sections 9.1.1 and 9.1.2. Observe from Figure 25, that also here, the values for $\nu_{exact}$ are dominated by odd numbers. Actually, 899 tests have $\nu_{exact} \in \{9, 11, 13\}$. At the same time, Figure 26 tells us that for 47.9% of the tests, it is sufficient with $\nu = 7$ to separate out only one candidate for the optimal decision. In fact, 658 tests had $\nu_{one} \leqslant 8$, while

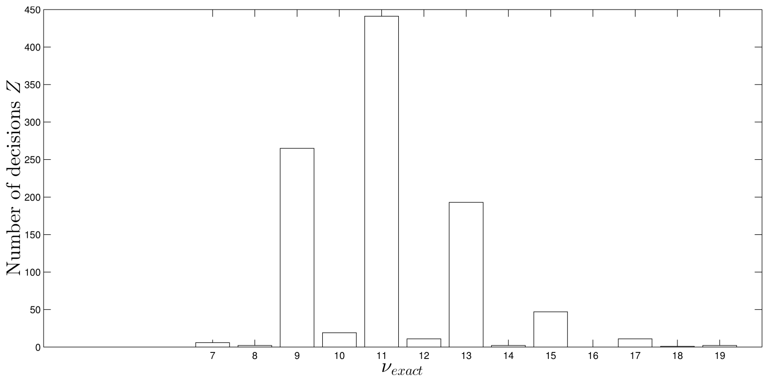

Figure 25: BN4: The value of $\nu$ needed to do exact calculations for the 1000 tests.

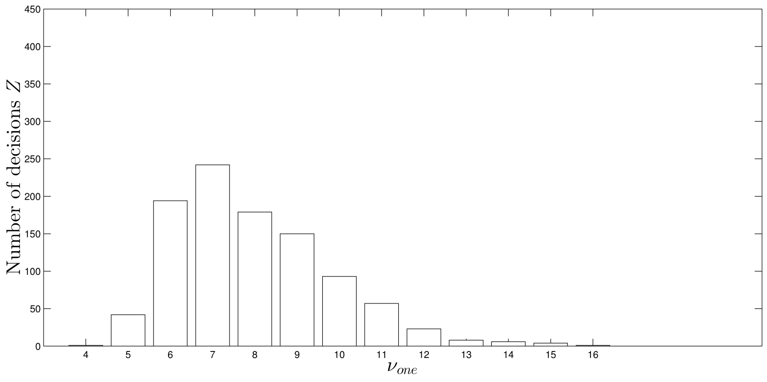

Figure 26: BN4: The value of $\nu$ needed to separate out only one candidate in each of the 1000 tests.

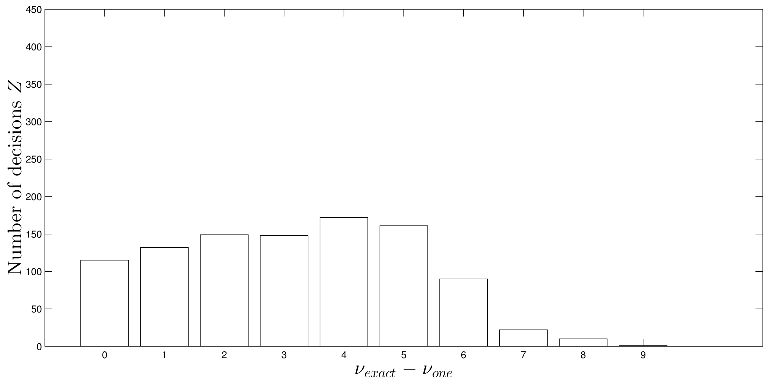

Figure 27: BN4: The difference between $\nu_{exact}$ and $\nu_{one}$ in each of the 1000 tests.

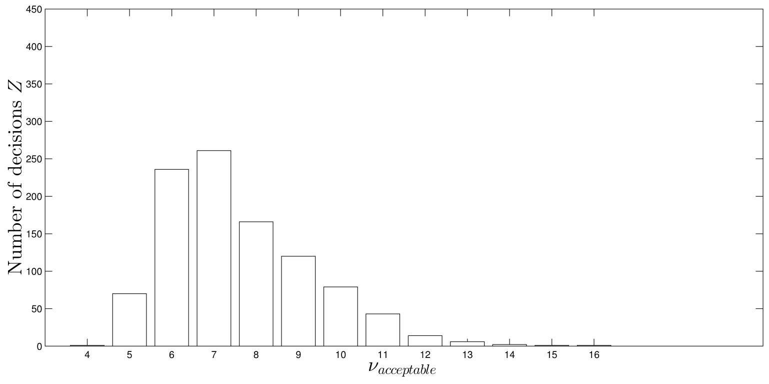

Figure 28: BN4: The value of $\nu$ needed to separate out a maximum of seven candidates in each of the 1000 tests.

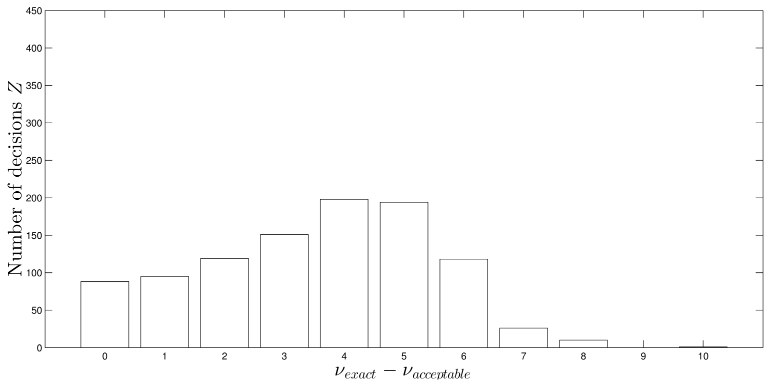

Figure 29: BN4: The difference between $\nu_{exact}$ and $\nu_{acceptable}$ in each of the 1000 tests.

only 8 tests had $\nu_{exact} \leqslant 8$. We observe from Figure 27 that 604 of the 1000 tests had a reduction of 3 or more from the value of $\nu_{exact}$ to the value of $\nu_{one}$. 48.1% of the cases had a reduction of either $3, 4$ or 5. Of the 1000 replications, 33 had a reduction of at least 7, but there are also 115 tests with no reduction from $\nu_{exact}$ to $\nu_{one}$. Similarly, we observe from Figure 28, that for as much as 56.8% of the tests, it is sufficient with $\nu = 7$ to reduce the number of candidates for the optimal decision to seven or less. Actually, 734 tests had $\nu_{acceptable} \leqslant 8$. Figure 27 shows how 698 of the 1000 tests had a reduction of 3 or more from the value of $\nu_{exact}$ to the value of $\nu_{acceptable}$. We observe that 88 of the tests had $\nu_{acceptable} = \nu_{exact}$, a significant reduction from the 115 tests with $\nu_{one} = \nu_{exact}$. Note that 37 of the 1000 tests had a reduction of 7 or more from the value of $\nu_{exact}$ to the value of $\nu_{acceptable}$. Also note that for one of the tests, the difference between $\nu_{exact}$ and $\nu_{acceptable}$ is as high as 10.

Also, for the 1000 tests, we chose $\nu = 7$, and got 67 cases where $L_7$ was nonzero. These came from the in total 521 cases where $\nu = 7$ was not enough to leave only one candidate for the optimal decision. Of these 521 cases, where a guess of the optimal decision had to be made, the average of $L_7$ was 0.032, with a standard deviation of 0.12. The smallest nonzero loss fraction was $7.955 \cdot 10^{-6}$, and the greatest loss fraction was 0.9579. In about half of the nonzero cases, the loss fraction was higher than 15%, and actually in 13 cases, the loss fraction is higher than 50%. For a listing of all nonzero loss fractions, the reader is referred to Table 77 in Section C.4. In total, on all 1000 cases, that left us with

| Prospect $P_i$ | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\nu$ | 2 | 3 | 3 | 7 | 3 | 3 | 7 | 3 | 7 | 7 | 3 | 7 | 6 | 16 | 17 |

Table 9: BN5: Value of $\nu$ that ensures exact calculation of expected gross income function corresponding to this prospect for the Bayesian Network in Figure 30.

an average loss $\bar{L_7} = 0.017$, with a standard deviation of 0.092.

## 9.2 A less sparse graph: BN5

To put our solution to a bigger challenge, we increase the number of edges, in order to observe how and when the solution method fails. Now, we look at a Bayesian Network with 7 Kitchens, 15 Prospects, 40 Segments, and 40 edges more than the minimum of 55 edges for this set of nodes. That means a graph with in total 62 Nodes and 95 edges. A visualization of the Bayesian Network can be seen in Figure 30. Notice the web of edges between the nodes, both in Figure 30 and in Figure 31. The Bayesian Network with corresponding cost and income functions was generated as described in Section 7. The parameters for drawing cost and income coefficients were set to

$$\mu_{\text{cost}} = 4000, \qquad \sigma^2_{\text{cost}} = 2500, \qquad \mu_{\text{inc}} = 5000, \qquad \sigma^2_{\text{inc}} = 2500.$$

The parameters for drawing the Joint Probability Distribution were set to

$$\alpha_{K_i} = 6.0, \qquad \beta_{K_i} = 2.0, \qquad \alpha_{e_{i,j}} = 5.0, \qquad \beta_{e_{i,j}} = 2.0.$$

For numbers fully determining the probability distribution and the cost and income functions, see Tables 78-93 in Section C.5.

From Table 9, we observe that for $7 \leqslant \nu \leqslant 15$, only approximations on $\mathbb{E}w_{14} - c_{14}$ and $\mathbb{E}w_{15} - c_{15}$ are made. Correspondingly, for $\nu = 16$, only approximations on $\mathbb{E}w_{15} - c_{15}$ are made. For the fraction of accepted decisions for a given $\nu$, see Table 10. The table clearly states that our approximations are not good enough. Any $\nu \leqslant 16$ implies getting a lower bound function that easily is strictly bounded by the smallest upper bound value. Hence, we are not able to eliminate any decisions, except when all calculations are done exactly. To study how and why this happens, we investigate the behavior of $\mathbb{P}(X_{A_{15}})$ as we sum out the Random Variables, one at a time. First, we calculate upper bounds on the log normalization constant for $\mathbb{P}(X_{A_{15}})$ for each $\nu \in \{10, \cdots, 17\}$, to obtain the data found in Table 11. Observe from this Table, that for all $\nu$s

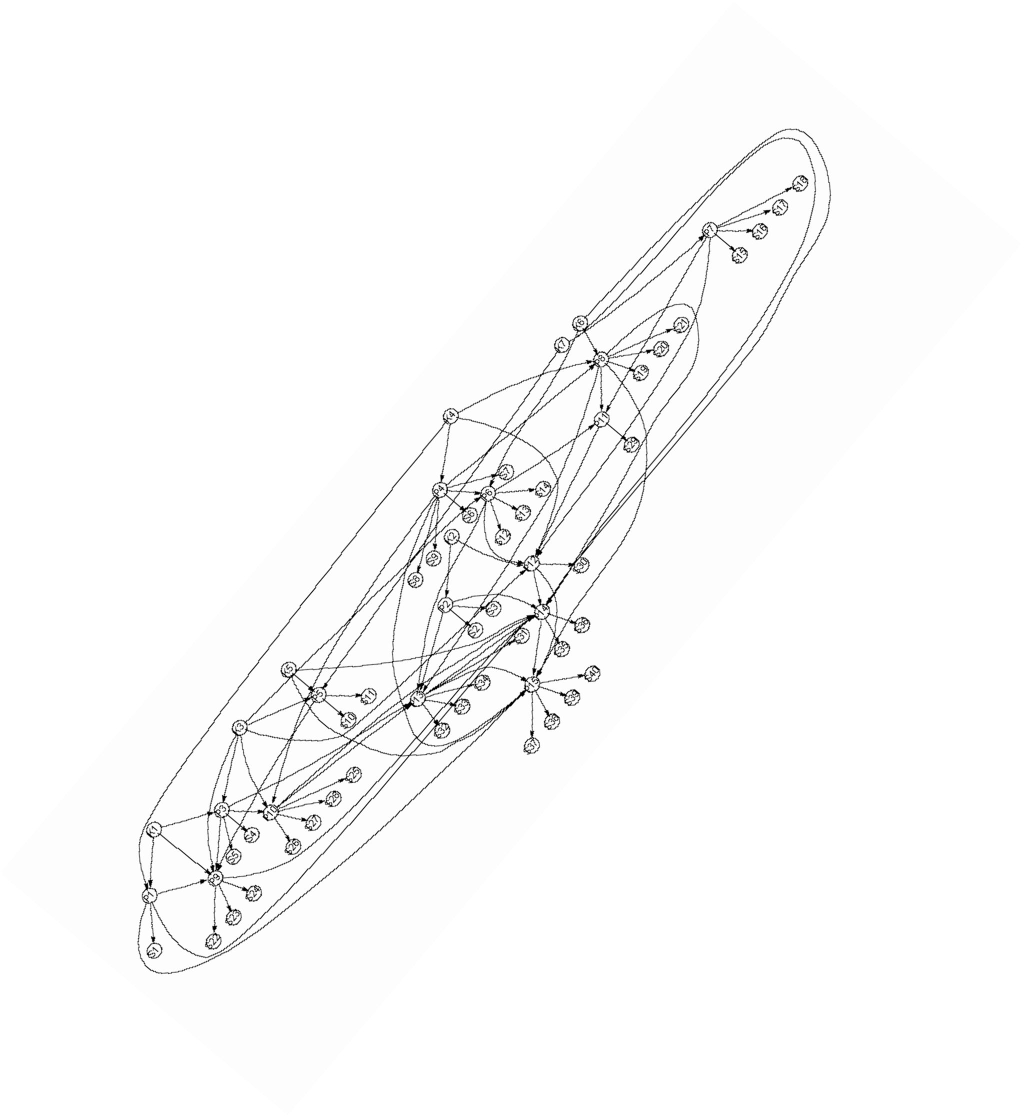

Figure 30: The Bayesian Network BN5.

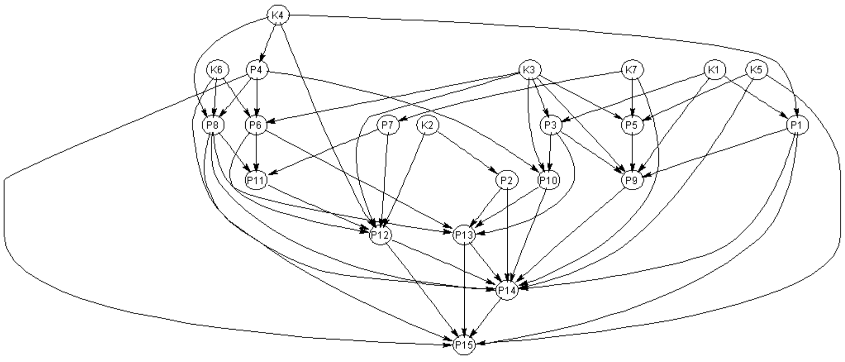

Figure 31: BN5: A visualization of only the Kitchens, the Prospects and the edges between them.

| Decision | Total number | Decisions accepted, $\nu =$ | |
|---|---|---|---|
| size | of decisions | $1-16$ | 17 |
| 0 | 1 | 1 | 0 |
| 1 | 40 | 40 | 0 |
| 2 | 780 | 780 | 1 |
| 3 | 9880 | 9880 | 0 |
| 4 | 91390 | 91390 | 0 |
| 5 | 658008 | 658008 | 0 |
| Sum | 760099 | 760099 | 1 |

Table 10: BN5: Number of accepted decisions as a function of $\nu$. Observe from Table 9 that we need $\nu = 17$ to get exact numbers, and that is the same value that is needed to eliminate any candidates for the optimal decision.

| $\nu$ | Log normalization constant |
|---|---|
| 10 | 193.930367 |
| 11 | 200.448060 |
| 12 | 231.803170 |
| 13 | 217.573379 |
| 14 | 243.259810 |
| 15 | 386.116269 |
| 16 | 92.7247710 |
| 17 | −0.0000000 |

Table 11: BN5: Upper bounds on the log normalization constant for the probability distribution $\pi_{15}$ calculated with different $\nu$s. 17 is the smallest value of $\nu$ that ensures no approximations to be made, and hence $\nu = 17$ gives the exact value $0 = \ln 1$. For all other values of $\nu$, the upper bounds calculated are useless.

except $\nu = 17$, we get upper bounds with enormous values. Note that $\nu = 17$ corresponds to exact calculations and thus, we obtain the exact normalization constant $1 = e^0$. Hence, the upper bounds are useless in this case. Also note that the normalization constant corresponds to some constant function

$$\exp\left(\beta^{\varnothing}\right),$$

and thus, that the bounds that failed to give good results, had resulted in a too large value of this $\beta^{\varnothing}$.

Still focusing on Prospect $P_{15}$, we observe how the value of $\beta^{\varnothing}$ changes as we sum out Random Variables in the expression for

$$\mathbb{E}(w_{15}(X_{B_{15}}, Z_{B_{15}}) - c(Z_{B_{15}}).$$

This test is done for $\nu = 10$, and the result is to be found in Table 12. We observe that the value of $\beta^{\varnothing}$ is slowly growing at first, and that the blow-up starts at the 11th ancestor we sum out. Note that Prospect $P_{15}$ has 13 ancestors, which means that we have to sum out a total of 14 nodes, including $P_{15}$ itself. After summing out all of those Random Variables, the resulting upper bound is a function of the decision $Z$, with $\beta^{\varnothing} = 1968$. Since the logarithm of the exact value for no decision is even close to this number, this is an upper bound function where no single evaluation of the lower bound function can compete with even the minimum of the upper bound. Hence, no candidates for the optimal decision can be eliminated.

| Number of Random Variables summed out | Value of $\beta^{\varnothing}$ in current representation |
|---|---|
| 0 | $-1.023$ |
| 1 | $-1.006$ |
| 2 | $-1.005$ |
| 3 | $-1.005$ |
| 4 | $-0.838$ |
| 5 | $-0.821$ |
| 6 | $-0.819$ |
| 7 | $-0.811$ |
| 8 | 0.715 |
| 9 | 1.450 |
| 10 | 1.702 |
| 11 | 8.014 |
| 12 | 17.48 |
| 13 | 169.7 |
| 14 | 196.8 |

Table 12: BN5: Value of $\beta^{\varnothing}$ for the upper bound function determined in each step, summing out a Random Variable at a time. Here, $\nu = 10$ and $\epsilon = 0.01$.

| $\nu$ | $\epsilon = 0.01$ | $\epsilon = 0.05$ | $\epsilon = 0.1$ | $\epsilon = 0.2$ |
|---|---|---|---|---|
| 15 | 386.1 | 86.34 | 15.90 | 1.409 |
| 16 | 92.72 | 19.51 | 5.190 | 0.733 |
| 17 | $-0.000$ | $-0.000$ | $-0.000$ | $-0.000$ |

Table 13: BN5: Upper bounds on the log normalization constant for the probability distribution $\pi_{15}$ calculated with different $\nu$s and different $\epsilon$s. 17 is the smallest value of $\nu$ that ensures no approximations to be made, and hence $\nu = 17$ gives the exact value $0 = \ln 1$.

Recall that we let $\epsilon > 0$ ensure that any assignment of the Random Variables has a positive probability. For the results in Table 11 and Table 12, $\epsilon = 0.01$ has been used. Taking the same Bayesian Network, with the same edge probabilities, we can vary $\epsilon$ and observe if that has any effect on this seemingly hopeless example. Again, we study only the probability distribution $\mathbb{P}(X_{A_{15}})$. Table 13 shows the resulting log normalization constants. Note that an increase in the value for $\epsilon$ introduces less error, but even high values of $\epsilon$ give useless results. Note that the results for $\epsilon = 0.2$ are not too bad, but 0.2 is definitely a questionable approximation to 0. Recall that $\epsilon > 0$, by definition, should be a small probability, hence much closer to the value 0 than the value 1.

## 10 Some Analysis on Complexity

First, we want to study the complexity of calculating $\mathbb{E}h(X, Z)$ with a naïve, and also exact, method, for some function $h$ depending on the entries in $X, Z$. If the function $h$ and the probability distribution $\mathbb{P}(X)$ is on the exponential pseudo-Boolean form, we can write

$$h(X, Z)\mathbb{P}(X) = \exp\left(\sum_{\lambda \in S} \beta^{\lambda} \prod_{k \in \lambda} \tilde{X}_k(X, Z)\right).$$

We could assume that $S$ has size $2^{n+|\Gamma|}$. That corresponds to a full representation of $\mathbb{P}(X)$, that is, $2^n$ possible combinations of $X_1, \cdots X_n$, and also a full representation for $h(X, Z)$ for the variables $X_1, \cdots X_n$ and $Z_1, \cdots Z_n$ with indexes in $\Gamma$. That is, calculating the value of $h(X, Z)\mathbb{P}(X)$ for an assignment to $X, Z$ has complexity $\mathcal{O}(2^{n+|\Gamma|})$. To get the expected value $\mathbb{E}h(X, Z)$ for a given $Z$, we would have to sum the value of $h(X, Z)\mathbb{P}(X)$ for $2^n$ different $X$s. That implies a complexity of $\mathcal{O}(2^{2n+|\Gamma|})$ for finding the expected value for a given $Z$. Our decision space $\Omega^{\Gamma,u}$ is of size

$$\left|\Omega^{\Gamma,u}\right| = \sum_{i=1}^{u} \binom{|\Gamma|}{i} \in \mathcal{O}\left(|\Gamma|^u\right),$$

since we assume $u << |\Gamma|$. To evaluate the expected value $\mathbb{E}h(X,Z)$ for each $Z$ in our decision space would then have a total complexity of

$$\mathcal{O}(|\Gamma|^u \cdot 2^{2n+|\Gamma|}). \tag{10.1}$$

However, note that this is based on a very naïve approach. In fact, we assume that the expected value is calculated as

$$\sum_{X_n}\left(\sum_{X_{n-1}}\cdots\left(\sum_{X_1}\exp\left(\sum_{\lambda\in S}\beta^\lambda\prod_{k\in\lambda}\tilde{X}_k(X,Z)\right)\right)\right),$$

where the expression inside each parenthesis pair is calculated without checking whether any factors can be moved outside the sum.

Let us again assume that we want to do exact calculations for the expected value $\mathbb{E}h(X,Z)$, now for a function $h$ and a probability distribution $\mathbb{P}(X)$ not in the exponential pseudo-Boolean form. It is natural to assume that we find the value of $\mathbb{P}(X)$ with $\mathcal{O}(n)$ calculations, and that the complexity of calculating $h(X,Z)$ is within this, and usually even simpler. The resulting total complexity would then be

$$\mathcal{O}(n \cdot 2^n \cdot |\Gamma|^u). \tag{10.2}$$

Now, we have two cases to compare to, and we want to study the complexity for our method, using bounds. Again, first, assume that we start with the function $h$ and the probability distribution $\mathbb{P}(X)$ on the exponential pseudo-Boolean form. For the input constant $\nu$, recall that calculating $\mathbb{E}h(X,Z)$ as a function of $Z$ has complexity $\mathcal{O}(n\cdot 2^\nu)$, plus the time spent on calculating the Approximate Markov Random Fields, which we assume is within the given complexity. This function $\mathbb{E}h(X,Z)$ is of the form

$$\mathbb{E}h(X,Z) = \exp\left(\sum_{\lambda\in\tilde{S}}\tilde{\beta}^\lambda\prod_{k\in\lambda}Z_k\right),$$

where for each $\lambda \in \tilde{S}$, we have $\lambda \subseteq \Gamma$. Thus, evaluating the function $\mathbb{E}h(X,Z)$ for a given $Z$ requires a sum over $2^{|\Gamma|}$ terms. Repeating this for each $Z$ in our decision space, yields a total complexity of

$$\mathcal{O}(n \cdot 2^{\nu} + 2^{|\Gamma|} \cdot |\Gamma|^{u}). \tag{10.3}$$

Comparing (10.1) and (10.3), we observe a significant reduction of complexity. However, note that to get this reduction, we both assumed that we start with the exponential pseudo-Boolean forms and that the approximations "The AMRF Program" makes are made in $\mathcal{O}(n \cdot 2^{\nu})$ time.

Looking at the case as in Section 8, we start with a function $h(X, Z)$ of the form

$$h(X, Z) = \sum_{\lambda \in T} \alpha^{\lambda} \prod_{k \in \lambda} X_k Z_k,$$

and then we need to calculate the form

$$h(X, Z) = \exp\left( \sum_{\lambda \in S} \beta_h^{\lambda} \prod_{k \in \lambda} X_k Z_k \right).$$

We can assume $S$ to be the power set of $\Gamma$, and hence that we need to calculate $2^{|\Gamma|}$ coefficients $\beta_h^{\lambda}$. Using formulas from Section 3.2 and the coefficient graph structure and formulas from Section A, we get that calculating each $\beta_h^{\Lambda}$ is of complexity $\mathcal{O}(|\lambda|^2)$. That is, we have assumed the $\beta_h^{\Lambda}$s are calculated in order according to increasing size of $\Lambda$, which ensures that all $\beta_h^{\lambda}$s with $\lambda \subsetneq \Lambda$ are available when we are calculating $\beta_h^{\Lambda}$. Then, calculating the wanted representation for the function $h$ has complexity

$$\mathcal{O}\left( \sum_{\lambda \subseteq \Gamma} |\lambda|^2 \right) = \mathcal{O}\left( \sum_{i=1}^{|\Gamma|} i^2 \binom{|\Gamma|}{i} \right) \subseteq \mathcal{O}\left( |\Gamma|^2 \cdot 2^{|\Gamma|} \right).$$

We also need to calculate the exponential pseudo-Boolean representation of the probability distribution

$$\mathbb{P}(X_1, \cdots X_n) = \prod_{i=1}^{n} \mathbb{P}(X_i | \mathrm{Pa}(X_i).$$

Assuming that

$$r = \max_{1 \leqslant i \leqslant n} |\mathrm{Pa}(X_i)|,$$

we get the pseudo-Boolean representation of $\mathbb{P}(X_i | \mathrm{Pa}(X_i))$ in $\mathcal{O}(2^{r+1})$ calculations. Assuming $r$ is not growing with $n$, which could be a reasonable assumption if we assume a certain "sparseness" to be constant as the size of the Bayesian Network grows, this is a constant running time. Thus, calculating the representation of the full Joint Probability Distribution has a linear complexity in the total number $n$ of Random Variables. This is because we calculate something of constant complexity for each Random Variable $X_i$. That would leave a total complexity of

$$\mathcal{O}(|\Gamma|^2 \cdot 2^{|\Gamma|} + n + n \cdot 2^{\nu} + 2^{|\Gamma|} \cdot |\Gamma|^u) \subseteq \mathcal{O}(n \cdot 2^{\nu} + 2^{|\Gamma|} \cdot |\Gamma|^u), \qquad (10.4)$$

assuming $u \geqslant 2$. Comparing (10.2) and (10.4), we still see a reduction of complexity. Note that the calculation of each of the pseudo-Boolean representations disappeared in the complexity of "The AMRF Programs" and the evaluation for all $Z$s, respectively. Also note that for the tests in Section 9, we would calculate the representation of $w_j - c_j$ for each $P_j$ with a complexity of $\mathcal{O}(t_j^2 2^{t_j})$, where $t_j$ is twice the number of $P_j$s Segment children. If we assume that the biggest such $t_j$ does not grow as we increase the size $n$ of our Bayesian Network, this would again mean that each $w_j - c_j$ representation is calculated in constant time. Thus, in total, all parts of $h$ is calculated with with a complexity linear in the number of Prospects, which could be assumed to grow linearly with the total number $n$ of Random Variables. In other words, this does not reduce the complexity of the full calculations, but it would reduce some of the constant factors.

# 11 Closing Remarks

We have studied five Boolean Bayesian networks; BN1-BN5, with Boolean decision vectors. First of all, from Section 8, we realized that our initial idea for how to do the calculations did not lead to good results. After, we could easily say that these calculations were done in a naïve way. But after some improvements, we observed that our solution method worked well on sparse graphs, as in BN2, BN3 and BN4. We also observed that usually, for a given $\nu$, the loss fraction $L_\nu$ for an approximate optimal decision $\tilde{Z}$ had a value equal to or close to zero. That means that on average, the approximate optimal decision $\tilde{Z}$ worked well. Note that the value of $\nu$ was chosen to be fairly close to the expected size of the maximal clique in the graph. However, in Table 77 in Section C.4, we find several loss fractions $L_\nu$ which clearly implies choosing a decision with a low exact value. This is why we get a high value for the standard deviation, especially for BN4. On average, the performance is still good, because the algorithm finds the optimal decision $Z^*$, or makes the right guess, in most cases, also with approximate values. To get less cases with high loss fractions, the algorithm could be modified to choose the exact best decision from the top few, according to the lower bounds, for example the top 5 or top 10. This would still allow us to do approximate calculations on most decisions.

We also observed how the solution method failed when the sparseness of the graph decreased, that is, when more edges were added. A further study of when this starts to happen, and how we could avoid this, would be interesting. Then, it would be natural to take a closer look on how the "optimal bounds" from Tjelmeland and Austad (2011) are calculated, and look for improvements or one's own version of such bounds. One could also look for improvement on the order of which the Random Variables are summed out, when calculating some expected value. Also here, a broader understanding of how the error bounds are calculated would be helpful.

This solution method could also be broadened by looking at upper and lower bounds on the value for a decision $Z \in \Omega^{\Gamma,u}$ where we only specify some of the entries. That is, for example, if we could find that any decision

$$Z^1 \in \{Z \in \Omega^{\Gamma,u} | Z_i = 1\}$$

would do worse than the best decision $Z^2 \in \Omega^{\Gamma,u}$ with $Z_i^2 = 0$, we would know that the optimal decision $Z^*$ does not have $Z_i = 1$. This could be done by some approximate Viterbi algorithm.

As mentioned in the beginning of this section, we have assumed Boolean Random Variables and Boolean decision vectors. However, there are no reason why this solution method could not be expanded. We could assume Random Variables that take on values in a set $A$ with $|A| > 2$. Then, our Random Vector $X$ would be living in $\Omega = A^n$. Correspondingly, there could be more than one choice for what to do with each Random Variable, that is, more choices for the values of the entries of the decision vector. The challenge is then in how to define functions and probability distributions so that we still can do approximations on these. In this report, we have assumed the approximations from Tjelmeland and Austad (2011), which are defined on pseudo-Boolean functions. Of course, each Random Variable $X_i \in A$ could be represented by a collection of Boolean Variables, and correspondingly for the decision entries $Z_i$. But probably, it would be better to use other upper and lower bounds for that specific case. In either case, the method for eliminating decisions as described in Section 6 is still valid.

Also, we recall from Section 10, that, of course, this solution method with "The AMRF Program" bounds, has the greatest reduction in complexity if we start with functions that already are on exponential pseudo-Boolean form. Thus, according to the complexity of the algorithm, this solution method would be most useful for cases where it is natural to start with such a formulation of the Joint Probability Distribution. However, note that for the dimensions in the tests for this report, most computational time was spent on calculating the expected value of the function $w - c$, and compared to that, the calculation of the exponential pseudo-Boolean forms was done in no time. Anyway, it could be interesting to look for applications, and study the performance on problems where the exponential pseudo-Boolean form is more natural. It is worth noting that in this report, we have only done tests for Bayesian Networks. Other graphical models could be of great interest. Note that for a Markov Random Field with Random Variables in $X$, and an unknown normalization constant $C$ for the probability distribution $\frac{1}{C}P(X)$, an upper bound on the expected value of a function $h(X)$ could be expressed as

$$\frac{\sum^{+}{}_{X} h(X)P(X)}{\sum^{-}{}_{X} P(X)},$$

where we let $\sum^{+}{}_{X}$ denote the upper bounds on the sum over $X$, and correspondingly, $\sum^{-}{}_{X}$ for the lower bound. Note that the Ising model is an example of such a distribution, and so is also any other Markov Random Field where the Joint Probability Distribution

$$\mathbb{P}(X) = \frac{1}{C} \prod_{E(X)} \exp(E(X)),$$

is defined by a collection of energy functions $\{E(X)\}$. Also, it could be interesting to look for applications where it is more natural to have the value function as the exponential of a pseudo-Boolean function. That is, where the value function behaves more like a product.

## References

Bishop, C. M. (2006). Graphical models, *Pattern Recognition and Machine Learning (Information Science and Statistics)*, Springer-Verlag New York, Inc.

Cormen, T. H., Leiserson, C. E., Rivest, R. L. and Stein, C. (2009). *Introduction to Algorithms*, 3rd edn, The MIT Press.

Hammer, P. L. and Rudeanu, S. (1968). *Boolean Methods in Operations Research and Related Areas*, 1st edn, Springer, Berlin.

Jordan, M. I. (n.d.). Conditional independence and factorization, *An Introduction to Probabilistic Graphical Models.* To appear.

Martinelli, G., Eidsvik, J. and Hauge, R. (2011). Dynamic decision making for graphical models applied to oil exploration, *Technical Report Preprint Statistics no. 12/2011*, Norwegian University of Science and Technology.

Russell, S. and Norvig, P. (2003). *Artificial Intelligence: A Modern Approach*, 2nd edn, Prentice Hall.

Tjelmeland, H. and Austad, H. M. (2011). An approximate forward-backward algorithm applied to binary Markov random fields, *Technical Report Preprint Statistics no. 11/2011*, Norwegian University of Science and Technology.

# APPENDIX

## A Implementation of Graph Structure

A graph like the one described in Section 3.1 is implemented by a C++ class. An alternative way of storing the coefficients $\alpha^\lambda$ would be in a vector of length $2^n$. That is, each subset $\lambda \subseteq N$ corresponds to entry number $E(\lambda) = \sum_{k\in\lambda} 2^k$ in the vector. Note that if we only want to look at sets $\lambda$ of size $|\lambda| \leqslant L$ with $L << m$, this alternative would imply a lot of unused entries in the vector. For the graph alternative, we are able to only create the first $L + 1$ layers of the graph, including the root node. However, the main argument for the graph alternative, is the efficient way of calculating sums over subsets, as in

$$\sum_{\lambda\subseteq\Lambda} \alpha^\lambda. \tag{A.1}$$

The number of layers in the graph, $L+1$, together with the number of different values appearing in the sets, give a unique description of the graph structure. Also, each node is an instance of a struct holding a vector of pointers to its parents, a vector of pointers to its children and a vector holding its corresponding values. Associate each node with its index set $\lambda$, that is, in this discussion, we will not distinguish between the index set and its corresponding node. Let the root node be layer 0, and name each layer outwards layer $1, \cdots, L$.

We introduce an ordering on the children and parents of node $a$, where child $b$ precedes child $c$ if the element added to $a$ to obtain $b$ has a lower value than the element added to obtain $c$. Correspondingly for the parents, parent $c$ precedes parent $d$ if the element removed from $a$ to obtain $c$ has a lower value than the element removed to obtain $d$. That is, the listings of children and parents in Section 3.1 follows the ordering described above. In the graph, we let each node $\lambda$ hold pointers to its children and parents, respectively, according to this order.

The length of the value vector may depend on the problem, and what we want to calculate for each node. Since we often want to calculate expressions of the kind in (A.1), this length is a multiple of the layer number $r$. If our purpose is to convert between the linear and exponential pseudo-Boolean representations of a function, as in Section 3.2, we would let the value vector length be twice the layer number, and let the root node hold two values. If we just want to store coefficients, or calculate sums over subsets, it is sufficient with a value vector length equal to the layer number, and the root node would hold one value. That is, call the value vector $v$, and assume for simplicity that each node has only one coefficient associated with it, i.e. we work with one set of coefficients $\alpha^{\lambda}$. Then, the value vector has length $r$ in layer $r$, and the first value element in the node $\Lambda$ holds the value of $\alpha^{\Lambda}$. The $k$th element, $k \geqslant 2$, is the sum of the coefficients $\alpha^{\lambda}$ where $\lambda \subset \Lambda$ and $|\Lambda \backslash \lambda| = k - 1$. That is, the sum over the first value element for each node in the layer $k$ steps down, reachable from $\Lambda$ by $k$ steps of going from a node to one of its parents. Then, if we let $\lambda \to v[k]$ denote the $k$th entry in the value vector for the node $\lambda$, we obtain the following two easy formulas for our graph structure,

$$\sum_{\lambda \subsetneq \Lambda} \alpha^{\lambda} = \varnothing \to v[0] + \sum_{k=2}^{|\lambda|-1} \lambda \to v[k], \tag{A.2}$$

and

$$\sum_{\lambda \subseteq \Lambda} \alpha^{\lambda} = \varnothing \to v[0] + \sum_{k=1}^{|\lambda|-1} \lambda \to v[k]. \tag{A.3}$$

Assume that we are given two nodes $\lambda_1$ and $\lambda_2$, where $\lambda_1$ is an ancestor of $\lambda_2$, $k$ layers up. There are $k!$ different routes from $y_1$ to $y_2$, going from a node to one of its children in each of the $k$ steps. That is, in the first step, you have $k$ different coordinates you can turn on, each of which in the next step gives $k - 1$ choices, and so on.

That results in the following formula, for $1 < k < |\lambda| - 1$,

$$\lambda \to v[k] = \frac{1}{k} \sum_{\tilde{\lambda} \in \mathrm{Pa}(\lambda)} \tilde{\lambda} \to v[k-1]. \tag{A.4}$$

The resulting algorithm first creates the graph structure of the dimensions wanted. Then, the first entry $\lambda \to v[1]$ in the value vector for each node $\lambda$ is assigned the value $\alpha^{\lambda}$. After that, layer by layer, starting with the grand children of the root node and continuing outwards, the rest of the values in each value vector is filled out according to (A.4). In the end, the wanted sums as in (A.1), are easily calculated by the formulas in (A.2) and (A.3).

# B A short User Oriented Introduction to "The AMRF Programs"

The programs referred to as "The AMRF Programs", are already implemented algorithms from Tjelmeland and Austad (2011), as mentioned in Section 4. For more information and the theory behind it, the reader is referred to Tjelmeland and Austad (2011).

## B.1 A pseudo-Boolean function as input

In addition to input parameters specific for each algorithm, a pseudo-Boolean function of the form

$$f(X) = \exp\left(\sum_{\lambda \in S} \beta^{\lambda} \prod_{k \in \lambda} X_k\right)$$

is given in an input file. In the model assumed, the nodes are numbered from $N_0$ to $N_{end}$, and each node $k$, with $N_0 \leqslant k \leqslant N_{end}$, has a corresponding coefficient $\beta^{\{k\}}$ in the coefficient set $\left\{\beta^{\lambda}\right\}_{\lambda \in S}$. The input file needs to be of the following format. The first line holds the start node number $N_0$, and the second line holds the end node number $N_{end}$. Then, on each line, a set coefficient $\beta^{\lambda}$ is presented, with no restriction on the order of the lines. But within each line, the following criteria need to be met.

- The first number on each line, is the size $|\lambda|$ of the coefficient set $\lambda$.
- The next numbers are the indexes $k \in \lambda$, in increasing order.
- Last, is the value of the corresponding coefficient $\beta^{\lambda}$.

As an example, the input file

```
1
3
0 -2.5
1 1 -2.8
1 2 -4.0
2 1 2 1.8
```

represents the function

$$f\left([X_1, X_2]\right) = \exp\left(-2.5 - 2.8X_1 - 4.0X_2 + 1.8X_1X_2\right).$$

## B.2 Calculating an approximate normalizing constant

The approximate normalizing constant, or an approximation of a marginal, for a pseudo-Boolean function as in Section B.1 will be found by use of the function call

```
amrf_graph_v3
(graphfile, simulation, likelihood, sim,limit, sstop, thev),
```

where the input parameters are

- **graphfile** is the name of the file where the pseudo-Boolean function is stored, as described in Section B.1.
- **simulation** is a file name for a file describing a realization for which the likelihood is to be evaluated (we set sim=1 to get one).
- **likelihood** is the name of the file where the output will be stored.
- We set **sim** = 1, since the program requires a realization to evaluate the likelihood of.
- We set **limit** = 0, since we don't want to do approximations corresponding to their $\epsilon$ parameter.
- **sstop** is the number of the last node to be summed out, i.e. we obtain the marginal for the nodes $x_{sstop+1}$ to $x_{N_{end}}$. Default is **sstop** = 0, which corresponds to all nodes being summed out, i.e. finding the normalizing constant of the distribution.
- **thev** is the maximum number of neighbors for each node for not making approximations before its summed out. That is, a small value of **thev** usually means more approximations, while a high enough value means no approximations.

## B.3 Calculating bounds for the normalizing constant

The upper and lower bounds for the normalizing constant, or bounds of a marginal, for a pseudo-Boolean function as in Section B.1, will be found by use of the function call

```
amrf_graph_bound(graphfile,
 simulation, likelihood, sim, limit, sstop, thev, maximum),
```

where the first seven input parameters are as for **amrf_graph_v3**, and the last parameter is set to be 1 for the upper bound, and 0 for the lower bound.

## B.4 An approximate Viterbi algorithm

The upper or lower bounds for the maximum function value, or a function that bounds the maximum over a subset of the variables, for a pseudo-Boolean function as in Section B.1, will be found by use of the function call

```
amrf_graph_viterbi(graphfile,
 simulation, likelihood, sim, limit, sstop, thev, maximum),
```

where the input parameters are as for **amrf_graph_bound**.

# C Probability Distribution, Cost and Income Functions for Section 8 and Section 9

## C.1 Distribution, Cost and Income functions BN1

First, Table 14 provides all edge probabilities $p_{e_{i,j}}$ to fully determine the Joint Probability Distribution for the graph in Figure 7 in Section 8. Recall that we have assumed the distribution to be as in Section 3.3.1. Then, Tables 15 and 16 provides the linear coefficients to fully determine the two sets of cost and income functions, $a$ and $b$, respectively. Recall that $\alpha_{cost}^{\vartheta(i)}$ represents the cost of drilling Segment $S_i$, while $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ represents the savings in the costs when drilling both Segment siblings $S_i, S_j$, and correspondingly for the income coefficients.

| Kitchen nr $K_i$ | $\mathbb{P}(K_i = 1)$ |
|---|---|
| K1 | 0.926 |
| K2 | 0.728 |
| K3 | 0.817 |
| K4 | 0.735 |

| Node pair $(i, j)$ | $p_{e_{i,j}}$ |
|---|---|
| (K1, P1) | 0.217 |
| (P1, D1) | 0.840 |
| (P1, D2) | 0.637 |
| (K2, P2) | 0.773 |
| (K3, P2) | 0.916 |
| (P2, D3) | 0.805 |
| (P2, D4) | 0.515 |
| (K2, P3) | 0.608 |
| (K3, P3) | 0.788 |
| (P3, D5) | 0.909 |
| (K4, P4) | 0.471 |
| (P2, P4) | 0.575 |
| (P4, D6) | 0.818 |
| (P4, D7) | 0.762 |
| (P4, P5) | 0.616 |
| (P5, D8) | 0.613 |
| (P2, P6) | 0.717 |
| (P4, P6) | 0.797 |
| (P6, D9) | 0.583 |
| (P4, P7) | 0.609 |
| (P6, P7) | 0.409 |
| (P7, D10) | 0.524 |

Table 14: BN1: Probabilities for the Bayes Net in Figure 7.

| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
|---|---|---|
| D1 | 4552 | 5660 |
| D2 | 2222 | 4693 |
| D3 | 5382 | 3996 |
| D4 | 1769 | 6829 |
| D5 | 501.9 | 3534 |
| D6 | 44.11 | 7291 |
| D7 | 3170 | 3868 |
| D8 | 365.9 | 2871 |
| D9 | 917.7 | 3342 |
| D10 | 1997 | 5326 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D1, D2) | 871.3 | 236.8 |
| (D3, D4) | 405.8 | 430.5 |
| (D6, D7) | 4.140 | 83.43 |

Table 15: BN1: Cost function $a$ and income function $a$.

| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
|---|---|---|
| D1 | 4729 | 4696 |
| D2 | 4962 | 3830 |
| D3 | 2419 | 2871 |
| D4 | 5094 | 3342 |
| D5 | 5758 | 5211 |
| D6 | 2189 | 6161 |
| D7 | 1928 | 5946 |
| D8 | 3611 | 4478 |
| D9 | 5277 | 7135 |
| D10 | 1673 | 4884 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D1, D2) | 3482 | 83.43 |
| (D3, D4) | 2180 | 139.3 |
| (D6, D7) | 649.8 | 288.9 |

Table 16: BN1: Cost function $b$ and income function $b$.

## C.2 Distribution, Cost and Income functions BN2

In this subsection, in addition to Sections C.3, C.4 and C.5, we will present the edge probabilities $p_e$, and also the cost and income coefficients in groups after which Prospect $P_i$ they belong to. That is, we say that $p_e$ belong to Prospect $P_i$ if Prospect $P_i$ is the end point of edge $e$, or if $e$ is an edge from Prospect $P_i$ to one of its Segment children. The cost and income coefficients under Prospect $P_i$, belongs to Prospect $P_i$s Segment children. Then, the numbers appear in groups as we use them to calculate the representation of $w_j - c_j$ and $\mathbb{P}(X_{A_j})$, respectively. Note that the calculation of the latter also uses calculations corresponding to previous Prospects, however. All numbers listed in this subsection, corresponds to BN2 from Section 9.1.1, visualized in Figure 8. Sections C.2, C.3 and C.4 also provides a table over the nonzero loss fractions $L_\nu$ in the 1000 replications test for each, respectively.

| Kitchen $K_i$ | $\mathbb{P}(K_i = 1)$ |
|---|---|
| K1 | 0.730 |
| K2 | 0.778 |
| K3 | 0.790 |
| K4 | 0.836 |
| K5 | 0.887 |
| K6 | 0.732 |
| K7 | 0.454 |

Table 17: Probabilities corresponding to each of Kitchens $K_i$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K1, P1) | 0.616 | |
| (P1, D1) | 0.358 | |
| PDS $D_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D1 | 3975 | 5068 |

Table 18: BN2: Probabilities, cost coefficients and income coefficients, Prospect $P_1$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K2, P2) | 0.813 | |
| (P2, D2) | 0.838 | |
| PDS $D_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D2 | 4005 | 5037 |

Table 19: BN2: Probabilities, cost coefficients and income coefficients, Prospect $P_2$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K3, P3) | 0.982 | |
| (P3, D3) | 0.700 | |
| (P3, D4) | 0.805 | |
| (P3, D5) | 0.496 | |
| (P3, D6) | 0.781 | |
| PDS $D_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D3 | 4036 | 5037 |
| D4 | 4002 | 5034 |
| D5 | 4012 | 5007 |
| D6 | 3984 | 4951 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D3, D4) | 345.6 | 522.8 |
| (D3, D5) | 153.9 | 490.1 |
| (D3, D6) | 603.1 | 493.7 |
| (D4, D5) | 1104 | 459.2 |
| (D4, D6) | 139.1 | 480.4 |
| (D5, D6) | 264.8 | 496.7 |

Table 20: BN2: Probabilities describing the probability distribution for Prospect $P_3$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K4, P4) | 0.937 | |
| (P4, D7) | 0.928 | |
| (P4, D8) | 0.548 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D7 | 3947 | 5031 |
| D8 | 4014 | 5037 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D7, D8) | 3614 | 501.3 |

Table 21: BN2: Probabilities, cost coefficients and income coefficients, Prospect $P_4$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K3, P5) | 0.797 | |
| (K5, P5) | 0.550 | |
| (P5, D9) | 0.762 | |
| (P5, D10) | 0.956 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D9 | 4046 | 5001 |
| D10 | 4024 | 4945 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D9, D10) | 3897 | 505.0 |

Table 22: BN2: Probabilities, cost coefficients and income coefficients, Prospect $P_5$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K3, P6) | 0.672 | |
| (K6, P6) | 0.613 | |
| (P6, D11) | 0.625 | |
| (P6, D12) | 0.789 | |
| (P6, D13) | 0.753 | |
| (P6, D14) | 0.807 | |
| (P6, D15) | 0.664 | |
| (P6, D16) | 0.841 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D11 | 4095 | 4863 |
| D12 | 3996 | 4985 |
| D13 | 4005 | 4988 |
| D14 | 4052 | 5110 |
| D15 | 3944 | 4949 |
| D16 | 4056 | 5003 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D11, D12) | 223.4 | 499.0 |
| (D11, D13) | 288.7 | 496.5 |
| (D11, D14) | 358.0 | 498.4 |
| (D11, D15) | 656.3 | 502.3 |
| (D11, D16) | 348.0 | 491.0 |
| (D12, D13) | 567.7 | 494.2 |
| (D12, D14) | 218.1 | 501.6 |
| (D12, D15) | 243.5 | 507.2 |
| (D12, D16) | 576.5 | 495.3 |
| (D13, D14) | 159.1 | 514.3 |
| (D13, D15) | 322.9 | 500.7 |
| (D13, D16) | 720.9 | 504.7 |
| (D14, D15) | 413.2 | 514.7 |
| (D14, D16) | 328.2 | 491.4 |
| (D15, D16) | 672.0 | 500.2 |

Table 23: BN2: Probabilities, cost coefficients and income coefficients, Prospect $P_6$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K7, P7) | 0.803 | |
| (P7, D17) | 0.626 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D17 | 3993 | 4982 |

Table 24: BN2: Probabilities, cost coefficients and income coefficients, Prospect $P_7$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K6, P8) | 0.606 | |
| (P8, D18) | 0.934 | |
| (P8, D19) | 0.529 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D18 | 3986 | 5010 |
| D19 | 4049 | 5001 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D18, D19) | 3804 | 510.9 |

Table 25: BN2: Probabilities, cost coefficients and income coefficients, Prospect $P_8$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K1, P9) | 0.265 | |
| (P3, P9) | 0.723 | |
| (P9, D20) | 0.354 | |
| (P9, D21) | 0.587 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D20 | 3996 | 5032 |
| D21 | 4017 | 5084 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D20, D21) | 2830 | 470.1 |

Table 26: BN2: Probabilities, cost coefficients and income coefficients, Prospect $P_9$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K3, P10) | 0.825 | |
| (P4, P10) | 0.809 | |
| (P10, D22) | 0.934 | |
| (P10, D23) | 0.759 | |
| (P10, D24) | 0.676 | |
| (P10, D25) | 0.718 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D22 | 3959 | 5017 |
| D23 | 3989 | 4939 |
| D24 | 4104 | 4976 |
| D25 | 4008 | 4940 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D22, D23) | 307.8 | 490.0 |
| (D22, D24) | 910.8 | 502.8 |
| (D22, D25) | 431.8 | 492.3 |
| (D23, D24) | 1096 | 465.9 |
| (D23, D25) | 1005 | 489.5 |
| (D24, D25) | 300.8 | 496.2 |

Table 27: BN2: Probabilities, cost coefficients and income coefficients, Prospect $P_{10}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (P6, P11) | 0.592 | |
| (P7, P11) | 0.784 | |
| (P8, P11) | 0.844 | |
| (P11, D26) | 0.921 | |
| (P11, D27) | 0.479 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D26 | 4066 | 5022 |
| D27 | 4065 | 5009 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D26, D27) | 3881 | 492.0 |

Table 28: BN2: Probabilities, cost coefficients and income coefficients, Prospect $P_{11}$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K4, P12) | 0.850 | |
| (P12, D28) | 0.837 | |
| (P12, D29) | 0.855 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D28 | 4024 | 5034 |
| D29 | 4034 | 5043 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D28, D29) | 1370 | 494.9 |

Table 29: BN2: Probabilities, cost coefficients and income coefficients, Prospect $P_{12}$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (P3, P13) | 0.267 | |
| (P8, P13) | 0.567 | |
| (P10, D30) | 0.724 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D30 | 4046 | 4988 |

Table 30: BN2: Probabilities, cost coefficients and income coefficients, Prospect $P_{13}$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K5, P14) | 0.892 | |
| (P13, P14) | 0.904 | |
| (P14, D31) | 0.769 | |
| (P14, D32) | 0.793 | |
| (P14, D33) | 0.587 | |
| (P14, D34) | 0.829 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| (D31, D32) | 1054 | 487.1 |
| (D31, D33) | 223.3 | 502.5 |
| (D31, D34) | 184.9 | 478.8 |
| (D32, D33) | 579.3 | 502.7 |
| (D32, D34) | 817.4 | 517.1 |
| (D33, D34) | 420.3 | 524.8 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D9, D10) | 3897 | 505.0 |

Table 31: BN2: Probabilities, cost coefficients and income coefficients, Prospect $P_{14}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ |
|---|---|
| (K5, P15) | 0.921 |
| (P4, P15) | 0.683 |
| (P14, P15) | 0.587 |
| (P15, D35) | 0.849 |
| (P15, D36) | 0.839 |
| (P15, D37) | 0.658 |
| (P15, D38) | 0.805 |
| (P15, D39) | 0.822 |
| (P15, D40) | 0.981 |

| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
|---|---|---|
| D35 | 3952 | 4976 |
| D36 | 3992 | 4958 |
| D37 | 3909 | 5019 |
| D38 | 3962 | 4965 |
| D39 | 4070 | 4964 |
| D40 | 4076 | 4912 |

| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
|---|---|---|
| (D35, D36) | 299.7 | 511.1 |
| (D35, D37) | 744.7 | 502.8 |
| (D35, D38) | 100.5 | 488.2 |
| (D35, D39) | 17.69 | 499.0 |
| (D35, D40) | 548.2 | 504.2 |
| (D36, D37) | 435.0 | 504.6 |
| (D36, D38) | 359.2 | 489.3 |
| (D36, D39) | 358.3 | 514.5 |
| (D36, D40) | 599.7 | 510.3 |
| (D37, D38) | 15.29 | 511.4 |
| (D37, D39) | 129.1 | 490.8 |
| (D37, D40) | 18.45 | 496.0 |
| (D38, D39) | 71.00 | 502.9 |
| (D38, D40) | 534.2 | 504.3 |
| (D39, D40) | 626.4 | 495.2 |

Table 32: BN2: Probabilities, cost coefficients and income coefficients, Prospect $P_{15}$.

| Nonzero loss fractions $L_6$ | | | |
|---|---|---|---|
| 0.0039% | 2.0994% | 6.6597% | 17.086% |
| 0.0171% | 3.0456% | 8.5423% | 22.791% |
| 0.0468% | 3.0662% | 8.7180% | 27.024% |
| 0.0596% | 3.2426% | 8.9417% | 29.596% |
| 0.2432% | 3.4734% | 9.2905% | 31.045% |
| 0.3263% | 4.3125% | 12.990% | 37.278% |
| 0.4715% | 4.4725% | 14.813% | 37.991% |
| 0.7584% | 5.2099% | 15.040% | 41.739% |
| 0.7924% | 5.3182% | 15.099% | 61.640% |
| 1.8310% | 6.0684% | 16.650% | 83.181% |

Table 33: BN2: Listing of all nonzero loss fractions $L_6$ for the 1000 tests in Section 9.1.1.

## C.3 Distribution, Cost and Income functions BN3

All numbers listed in this subsection, corresponds to BN3 from Section 9.1.1, visualized in Figure 15.

| Kitchen nr $K_i$ | $\mathbb{P}(K_i = 1)$ |
|---|---|
| K1 | 0.695 |
| K2 | 0.944 |
| K3 | 0.760 |
| K4 | 0.638 |
| K5 | 0.615 |
| K6 | 0.668 |
| K7 | 0.870 |
| K8 | 0.593 |
| K9 | 0.859 |
| K10 | 0.821 |

Table 34: BN3: Probabilities corresponding to each of Kitchens $K_i$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K1, P1) | 0.510 | |
| (P1, D1) | 0.813 | |
| (P1, D2) | 0.934 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D1 | 3003 | 4949 |
| D2 | 3012 | 5053 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D1, D2) | 1273 | 478.3 |

Table 35: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_1$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K2, P2) | 0.967 | |
| (P2, D3) | 0.878 | |
| (P2, D4) | 0.646 | |
| (P2, D5) | 0.780 | |
| (P2, D6) | 0.586 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D3 | 3004 | 4967 |
| D4 | 2965 | 4987 |
| D5 | 2933 | 5033 |
| D6 | 2969 | 5050 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D3, D4) | 84.85 | 518.9 |
| (D3, D5) | 950.4 | 483.4 |
| (D4, D5) | 478.8 | 499.2 |
| (D3, D6) | 219.1 | 532.8 |
| (D4, D6) | 805.0 | 464.1 |
| (D5, D6) | 566.4 | 531.5 |

Table 36: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_2$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K1, P3) | 0.740 | |
| (K3, P3) | 0.585 | |
| (P3, D7) | 0.830 | |
| (P3, D8) | 0.968 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D7 | 3072 | 5060 |
| D8 | 3000 | 4904 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D7, D8) | 1663 | 488.6 |

Table 37: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_3$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K4, P4) | 0.581 | |
| (P4, D9) | 0.589 | |
| (P4, D10) | 0.378 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D9 | 3042 | 5070 |
| D10 | 2990 | 5021 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D9, D10) | 2256 | 510.0 |

Table 38: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_4$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K5, P5) | 0.782 | |
| (P5, D11) | 0.576 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D11 | 3050 | 4999 |

Table 39: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_5$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K6, P6) | 0.532 | |
| (P6, D12) | 0.325 | |
| (P6, D13) | 0.692 | |
| (P6, D14) | 0.640 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D12 | 3011 | 5044 |
| D13 | 2988 | 4958 |
| D14 | 3086 | 4982 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D12, D13) | 586.1 | 509.0 |
| (D12, D14) | 252.3 | 529.2 |
| (D13, D14) | 431.3 | 490.3 |

Table 40: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_6$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K7, P7) | 0.778 | |
| (P7, D15) | 0.687 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D15 | 2979 | 4939 |

Table 41: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_7$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K8, P8) | 0.563 | |
| (P8, D16) | 0.816 | |
| (P8, D17) | 0.790 | |
| (P8, D18) | 0.494 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D16 | 3078 | 4998 |
| D17 | 3065 | 4921 |
| D18 | 3054 | 4999 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D16, D17) | 1286 | 467.0 |
| (D16, D18) | 901.9 | 518.6 |
| (D17, D18) | 348.5 | 507.8 |

Table 42: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_8$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K9, P9) | 0.549 | |
| (P9, D19) | 0.550 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D19 | 2932 | 5109 |

Table 43: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_9$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K10, P10) | 0.676 | |
| (P10, D20) | 0.678 | |
| (P10, D21) | 0.782 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D20 | 2987 | 5052 |
| D21 | 3010 | 5033 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D20, D21) | 2526 | 479.1 |

Table 44: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_{10}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (P2, P11) | 0.852 | |
| (P9, P11) | 0.860 | |
| (P11, D22) | 0.779 | |
| (P11, D23) | 0.842 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D22 | 2941 | 4916 |
| D23 | 2989 | 5011 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D22, D23) | 2823 | 512.3 |

Table 45: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_{11}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K5, P12) | 0.502 | |
| (P12, D24) | 0.624 | |
| (P12, D25) | 0.823 | |
| (P12, D26) | 0.557 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D24 | 3005 | 5070 |
| D25 | 2924 | 5016 |
| D26 | 2946 | 5008 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D24, D25) | 1247 | 511.6 |
| (D24, D26) | 641.8 | 490.1 |
| (D25, D26) | 956.3 | 497.4 |

Table 46: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_{12}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (P10, P13) | 0.976 | |
| (P13, D27) | 0.901 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D27 | 3072 | 4992 |

Table 47: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_{13}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K10, P14) | 0.978 | |
| (P4, P14) | 0.817 | |
| (P9, P14) | 0.696 | |
| (P14, D28) | 0.659 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D28 | 2969 | 5022 |

Table 48: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_{14}$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K2, P15) | 0.763 | |
| (P7, P15) | 0.890 | |
| (P13, P15) | 0.836 | |
| (P15, D29) | 0.850 | |
| (P15, D30) | 0.861 | |
| (P15, D31) | 0.776 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D29 | 2987 | 4951 |
| D30 | 3004 | 5023 |
| D31 | 3022 | 5020 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D29, D30) | 267.8 | 497.1 |
| (D29, D31) | 894.1 | 482.6 |
| (D30, D31) | 1370 | 490.2 |

Table 49: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_{15}$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (P6, P16) | 0.719 | |
| (P16, D32) | 0.436 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D32 | 2966 | 4989 |

Table 50: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_{16}$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (P9, P17) | 0.898 | |
| (P15, P17) | 0.617 | |
| (P17, D33) | 0.478 | |
| (P17, D34) | 0.561 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D33 | 2913 | 4938 |
| D34 | 2933 | 4980 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D33, D34) | 1932 | 517.8 |

Table 51: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_{17}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K1, P18) | 0.246 | |
| (P14, P18) | 0.863 | |
| (P18, D35) | 0.634 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D35 | 2956 | 4952 |

Table 52: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_{18}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (P7, P19) | 0.910 | |
| (P19, D36) | 0.760 | |
| (P19, D37) | 0.867 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D36 | 3050 | 5064 |
| D37 | 2941 | 5023 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D36, D37) | 1833 | 514.1 |

Table 53: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_{19}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K3, P20) | 0.601 | |
| (P15, P20) | 0.520 | |
| (P19, P20) | 0.498 | |
| (P20, D38) | 0.785 | |
| (P20, D39) | 0.887 | |
| (P20, D40) | 0.702 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D38 | 2920 | 5019 |
| D39 | 2908 | 5048 |
| D40 | 3003 | 4985 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D38, D39) | 1395 | 514.6 |
| (D38, D40) | 179.8 | 523.7 |
| (D39, D40) | 289.0 | 480.6 |

Table 54: BN3: Probabilities, cost coefficients and income coefficients, Prospect $P_{20}$.

| Nonzero loss fractions $L_5$ | | |
|---|---|---|
| 0.0846% | 0.1268% | 0.1278% |
| 0.1483% | 0.1529% | 0.6006% |
| 1.1325% | 1.7083% | 1.7236% |
| 1.8870% | 2.5999% | 2.6884% |
| 3.2948% | 3.3345% | 3.4698% |
| 3.8284% | 4.1870% | 4.5679% |
| 5.1716% | 5.4370% | 6.0701% |
| 6.9327% | 10.1125% | 20.0158% |
| 20.5417% | 25.1488% | 45.8189% |

Table 55: BN3: Listing of all nonzero loss fractions $L_5$ for the 1000 tests in Section 9.1.2.

## C.4 Distribution, Cost and Income functions BN4

All numbers listed in this subsection, corresponds to BN4 from Section 9.1.1, visualized in Figure 23.

| Kitchen nr $K_i$ | $\mathbb{P}(K_i = 1)$ |
|---|---|
| K1 | 0.719 |
| K2 | 0.487 |
| K3 | 0.897 |
| K4 | 0.759 |
| K5 | 0.817 |
| K6 | 0.904 |
| K7 | 0.351 |
| K8 | 0.678 |
| K9 | 0.874 |
| K10 | 0.680 |

Table 56: BN4: Probabilities corresponding to each of Kitchens $K_i$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K1, P1) | 0.700 | |
| (K8, P1) | 0.833 | |
| (P1, D1) | 0.778 | |
| (P1, D2) | 0.820 | |
| (P1, D3) | 0.815 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D1 | 4030 | 5015 |
| D2 | 4046 | 4989 |
| D3 | 4037 | 5016 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D1, D2) | 1889 | 471.4 |
| (D1, D3) | 284.2 | 488.2 |
| (D2, D3) | 457.7 | 495.3 |

Table 57: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_1$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K2, P2) | 0.678 | |
| (K3, P2) | 0.844 | |
| (P2, D4) | 0.707 | |
| (P2, D5) | 0.504 | |
| (P2, D6) | 0.516 | |
| (P2, D7) | 0.760 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D4 | 4042 | 5028 |
| D5 | 3989 | 5013 |
| D6 | 4001 | 4999 |
| D7 | 4000 | 5029 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D4, D5) | 158.8 | 493.2 |
| (D4, D6) | 235.6 | 490.1 |
| (D5, D6) | 1198 | 504.3 |
| (D4, D7) | 178.5 | 502.7 |
| (D5, D7) | 242.3 | 511.3 |
| (D6, D7) | 878.2 | 515.7 |

Table 58: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_2$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (P3, D8) | 0.714 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D8 | 4056 | 5050 |

Table 59: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_3$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K4, P4) | 0.478 | |
| (P3, P4) | 0.815 | |
| (P4, D9) | 0.621 | |
| (P4, D10) | 0.735 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D9 | 4037 | 5044 |
| D10 | 3955 | 4975 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D9, D10) | 727.7 | 491.7 |

Table 60: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_4$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K5, P5) | 0.803 | |
| (K7, P5) | 0.846 | |
| (P5, D11) | 0.676 | |
| (P5, D12) | 0.537 | |
| (P5, D13) | 0.379 | |
| (P5, D14) | 0.523 | |
| (P5, D15) | 0.799 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D11 | 3953 | 5025 |
| D12 | 4054 | 4959 |
| D13 | 4043 | 4994 |
| D14 | 3988 | 5099 |
| D15 | 3954 | 4987 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D11, D12) | 92.44 | 524.1 |
| (D11, D13) | 606.0 | 479.3 |
| (D12, D13) | 64.68 | 530.0 |
| (D11, D14) | 361.3 | 511.1 |
| (D12, D14) | 764.5 | 470.1 |
| (D13, D14) | 948.3 | 505.4 |
| (D11, D15) | 929.5 | 505.9 |
| (D12, D15) | 227.3 | 524.8 |
| (D13, D15) | 575.1 | 502.6 |
| (D14, D15) | 475.3 | 491.8 |

Table 61: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_5$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K6, P6) | 0.937 | |
| (P6, D16) | 0.462 | |
| (P6, D17) | 0.698 | |
| (P6, D18) | 0.877 | |
| (P6, D19) | 0.722 | |
| (P6, D20) | 0.592 | |
| (P6, D21) | 0.395 | |
| (P6, D22) | 0.590 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D16 | 4064 | 5007 |
| D17 | 3944 | 4934 |
| D18 | 3964 | 4964 |
| D19 | 3972 | 5068 |
| D20 | 3959 | 5016 |
| D21 | 4025 | 4982 |
| D22 | 3952 | 5013 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D16, D17) | 414.1 | 517.2 |
| (D16, D18) | 239.6 | 509.5 |
| (D17, D18) | 288.5 | 483.8 |
| (D16, D19) | 295.4 | 502.3 |
| (D17, D19) | 313.5 | 497.2 |
| (D18, D19) | 364.3 | 501.3 |
| (D16, D20) | 631.8 | 471.0 |
| (D17, D20) | 65.94 | 499.8 |
| (D18, D20) | 528.9 | 484.6 |
| (D19, D20) | 302.6 | 512.9 |
| (D16, D21) | 372.8 | 480.1 |
| (D17, D21) | 400.5 | 507.2 |
| (D18, D21) | 360.7 | 482.2 |
| (D19, D21) | 647.6 | 501.6 |
| (D20, D21) | 184.2 | 503.5 |
| (D16, D22) | 519.2 | 495.1 |
| (D17, D22) | 78.64 | 494.0 |
| (D18, D22) | 345.7 | 500.9 |
| (D19, D22) | 549.0 | 480.5 |
| (D20, D22) | 48.52 | 502.9 |
| (D21, D22) | 495.5 | 494.7 |

Table 62: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_6$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K7, P7) | 0.894 | |
| (P7, D23) | 0.569 | |
| (P7, D24) | 0.933 | |
| (P7, D25) | 0.885 | |
| (P7, D26) | 0.396 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D23 | 4008 | 5074 |
| D24 | 3964 | 4969 |
| D25 | 4029 | 5049 |
| D26 | 4072 | 5025 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D23, D24) | 155.5 | 485.7 |
| (D23, D25) | 286.2 | 510.0 |
| (D24, D25) | 1048 | 469.7 |
| (D23, D26) | 853.5 | 489.7 |
| (D24, D26) | 399.8 | 492.3 |
| (D25, D26) | 1079 | 491.1 |

Table 63: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_7$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K1, P8) | 0.801 | |
| (K8, P8) | 0.862 | |
| (P8, D27) | 0.768 | |
| (P8, D28) | 0.943 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D27 | 4028 | 5016 |
| D28 | 4028 | 4933 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D27, D28) | 980.0 | 510.6 |

Table 64: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_8$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K9, P9) | 0.838 | |
| (P9, D29) | 0.757 | |
| (P9, D30) | 0.759 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D29 | 4042 | 4938 |
| D30 | 4078 | 4945 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D29, D30) | 1355 | 496.9 |

Table 65: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_9$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K3, P10) | 0.838 | |
| (K6, P10) | 0.674 | |
| (K10, P10) | 0.753 | |
| (P10, D31) | 0.677 | |
| (P10, D32) | 0.840 | |
| (P10, D33) | 0.776 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D31 | 4042 | 5005 |
| D32 | 3934 | 4966 |
| D33 | 3949 | 5013 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D31, D32) | 332.0 | 499.2 |
| (D31, D33) | 931.6 | 480.5 |
| (D32, D33) | 1910 | 484.4 |

Table 66: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_{10}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K7, P11) | 0.813 | |
| (P5, P11) | 0.554 | |
| (P8, P11) | 0.766 | |
| (P11, D34) | 0.623 | |
| (P11, D35) | 0.475 | |
| (P11, D36) | 0.697 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D34 | 3930 | 5065 |
| D35 | 3969 | 4944 |
| D36 | 4058 | 5007 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D34, D35) | 581.2 | 474.5 |
| (D34, D36) | 1263 | 504.4 |
| (D35, D36) | 531.3 | 484.6 |

Table 67: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_{11}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (P3, P12) | 0.900 | |
| (P12, D37) | 0.858 | |
| (P12, D38) | 0.613 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D37 | 4004 | 5034 |
| D38 | 3986 | 4980 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D37, D38) | 3758 | 491.5 |

Table 68: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_{12}$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K4, P13) | 0.706 | |
| (K7, P13) | 0.653 | |
| (P9, P13) | 0.816 | |
| (P13, D39) | 0.563 | |
| (P13, D40) | 0.895 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D39 | 3965 | 5049 |
| D40 | 4043 | 4993 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D39, D40) | 1186 | 476.6 |

Table 69: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_{13}$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K5, P14) | 0.866 | |
| (P14, D41) | 0.822 | |
| (P14, D42) | 0.593 | |
| (P14, D43) | 0.508 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D41 | 4073 | 5059 |
| D42 | 4015 | 4962 |
| D43 | 3977 | 4963 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D41, D42) | 292.3 | 511.7 |
| (D41, D43) | 1978 | 478.8 |
| (D42, D43) | 1792 | 486.1 |

Table 70: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_{14}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (P11, P15) | 0.605 | |
| (P12, P15) | 0.864 | |
| (P15, D44) | 0.853 | |
| (P15, D45) | 0.572 | |
| (P15, D46) | 0.532 | |
| (P15, D47) | 0.942 | |
| **Segment $S_i$** | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D44 | 4059 | 5024 |
| D45 | 4045 | 4921 |
| D46 | 3992 | 4975 |
| D47 | 3992 | 5049 |
| **Segment siblings $S_i, S_j$** | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D44, D45) | 72.11 | 485.5 |
| (D44, D46) | 696.3 | 475.1 |
| (D45, D46) | 449.1 | 522.0 |
| (D44, D47) | 950.7 | 489.1 |
| (D45, D47) | 777.7 | 458.4 |
| (D46, D47) | 1012 | 516.7 |

Table 71: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_{15}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K1, P16) | 0.625 | |
| (P16, D48) | 0.590 | |
| (P16, D49) | 0.488 | |
| **Segment $S_i$** | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D48 | 4002 | 5015 |
| D49 | 3967 | 4955 |
| **Segment siblings $S_i, S_j$** | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D48, D49) | 2572 | 484.7 |

Table 72: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_{16}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (P4, P17) | 0.633 | |
| (P5, P17) | 0.826 | |
| (P17, D50) | 0.822 | |
| (P17, D51) | 0.963 | |
| (P17, D52) | 0.555 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D50 | 4055 | 4999 |
| D51 | 3964 | 4993 |
| D52 | 4081 | 4947 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D50, D51) | 739.5 | 501.3 |
| (D50, D52) | 701.6 | 502.1 |
| (D51, D52) | 1895 | 510.7 |

Table 73: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_{17}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (P3, P18) | 0.841 | |
| (P4, P18) | 0.750 | |
| (P15, P18) | 0.771 | |
| (P18, D53) | 0.711 | |
| (P18, D54) | 0.971 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D53 | 4049 | 5046 |
| D54 | 4062 | 5044 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D53, D54) | 146.4 | 502.4 |

Table 74: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_{18}$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K1, P19) | 0.695 | |
| (P2, P19) | 0.833 | |
| (P5, P19) | 0.618 | |
| (P16, P19) | 0.351 | |
| (P19, D55) | 0.386 | |
| (P19, D56) | 0.741 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D55 | 3944 | 5013 |
| D56 | 3923 | 5003 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D55, D56) | 2692 | 508.8 |

Table 75: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_{19}$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K3, P20) | 0.669 | |
| (P14, P20) | 0.984 | |
| (P19, P20) | 0.813 | |
| (P20, D57) | 0.509 | |
| (P20, D58) | 0.633 | |
| (P20, D59) | 0.774 | |
| (P20, D60) | 0.789 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D57 | 4003 | 4981 |
| D58 | 4042 | 4964 |
| D59 | 4051 | 4945 |
| D60 | 3945 | 4938 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D57, D58) | 802.0 | 509.7 |
| (D57, D59) | 304.2 | 472.6 |
| (D58, D59) | 439.7 | 499.7 |
| (D57, D60) | 279.4 | 530.4 |
| (D58, D60) | 951.4 | 491.4 |
| (D59, D60) | 423.4 | 512.8 |

Table 76: BN4: Probabilities, cost coefficients and income coefficients, Prospect $P_{20}$.

| Nonzero loss fractions $L_7$ | | | |
|---|---|---|---|
| 0.0008% | 0.0143% | 0.1081% | 0.1408% |
| 0.4298% | 0.4729% | 1.2513% | 1.3379% |
| 1.4173% | 1.5933% | 1.6141% | 1.6901% |
| 2.3847% | 2.4245% | 3.1822% | 3.7893% |
| 3.8178% | 4.3210% | 4.3220% | 4.9154% |
| 5.3173% | 5.8367% | 6.0373% | 6.1426% |
| 6.6618% | 7.5549% | 8.2433% | 9.0723% |
| 9.3761% | 9.5243% | 10.5092% | 12.3766% |
| 14.8005% | 15.5166% | 15.6522% | 17.3257% |
| 17.6869% | 17.7826% | 19.2693% | 20.2625% |
| 20.9264% | 21.2441% | 23.4648% | 24.5481% |
| 26.7271% | 27.8775% | 29.7215% | 36.6901% |
| 36.9098% | 39.4034% | 40.1745% | 44.1414% |
| 47.6808% | 49.1176% | 50.9428% | 53.4218% |
| 55.2268% | 63.2664% | 66.5337% | 66.7269% |
| 69.9822% | 72.1796% | 73.6152% | 76.5762% |
| 78.0995% | 90.0164% | 95.7869% | |

Table 77: BN4: Listing of all nonzero loss fractions $L_7$ for the 1000 tests in Section 9.1.3.

## C.5 Distribution, Cost and Income functions BN5

All numbers listed in this subsection, corresponds to BN5 from Section 9.2, visualized in Figure 30.

| Kitchen $K_i$ | $\mathbb{P}(K_i = 1)$ |
|---|---|
| K1 | 0.730 |
| K2 | 0.778 |
| K3 | 0.790 |
| K4 | 0.836 |
| K5 | 0.887 |
| K6 | 0.732 |
| K7 | 0.454 |

Table 78: BN5: Probabilities corresponding to each of Kitchens $K_i$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K1, P1) | 0.616 | |
| (K4, P1) | 0.813 | |
| (P1, D1) | 0.769 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D1 | 3863 | 4996 |

Table 79: BN5: Probabilities, cost coefficients and income coefficients, Prospect $P_1$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K2, P2) | 0.982 | |
| (P2, D2) | 0.793 | |
| (P2, D3) | 0.587 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D2 | 3985 | 5005 |
| D3 | 3988 | 5001 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D2, D3) | 313.8 | 494.2 |

Table 80: BN5: Probabilities, cost coefficients and income coefficients, Prospect $P_2$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K1, P3) | 0.937 | |
| (K3, P3) | 0.797 | |
| (P3, D4) | 0.829 | |
| (P3, D5) | 0.849 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D4 | 4052 | 5110 |
| D5 | 3996 | 5058 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D4, D5) | 3511 | 501.6 |

Table 81: BN5: Probabilities, cost coefficients and income coefficients, Prospect $P_3$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K4, P4) | 0.550 | |
| (P4, D6) | 0.839 | |
| (P4, D7) | 0.658 | |
| (P4, D8) | 0.805 | |
| (P4, D9) | 0.822 | |
| **Segment $S_i$** | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D6 | 3981 | 5052 |
| D7 | 4046 | 4944 |
| D8 | 4091 | 5048 |
| D9 | 4002 | 5036 |
| **Segment siblings $S_i, S_j$** | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D6, D7) | 473.2 | 502.3 |
| (D6, D8) | 940.6 | 484.3 |
| (D7, D8) | 1312 | 517.3 |
| (D6, D9) | 1160 | 495.3 |
| (D7, D9) | 1201 | 504.7 |
| (D8, D9) | 540.3 | 491.4 |

Table 82: BN5: Probabilities, cost coefficients and income coefficients, Prospect $P_4$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K3, P5) | 0.672 | |
| (K5, P5) | 0.613 | |
| (K7, P5) | 0.803 | |
| (P5, D10) | 0.981 | |
| (P5, D11) | 0.459 | |
| **Segment $S_i$** | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D10 | 3990 | 5002 |
| D11 | 3993 | 4982 |
| **Segment siblings $S_i, S_j$** | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D10, D11) | 516.4 | 493.2 |

Table 83: BN5: Probabilities, cost coefficients and income coefficients, Prospect $P_5$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K3, P6) | 0.606 | |
| (K6, P6) | 0.265 | |
| (P4, P6) | 0.723 | |
| (P6, D12) | 0.677 | |
| (P6, D13) | 0.520 | |
| (P6, D14) | 0.863 | |
| **Segment $S_i$** | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D12 | 4054 | 4928 |
| D13 | 3956 | 5036 |
| D14 | 4009 | 5034 |
| **Segment siblings $S_i, S_j$** | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D12, D13) | 1456 | 494.1 |
| (D12, D14) | 85.20 | 508.3 |
| (D13, D14) | 1244 | 486.7 |

Table 84: BN5: Probabilities, cost coefficients and income coefficients, Prospect $P_6$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K7, P7) | 0.825 | |
| (P7, D15) | 0.589 | |
| (P7, D16) | 0.996 | |
| (P7, D17) | 0.495 | |
| (P7, D18) | 0.478 | |
| **Segment $S_i$** | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D15 | 4017 | 4989 |
| D16 | 3939 | 5077 |
| D17 | 3961 | 5021 |
| D18 | 3978 | 4948 |
| **Segment siblings $S_i, S_j$** | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D15, D16) | 225.9 | 502.8 |
| (D15, D17) | 678.7 | 465.9 |
| (D16, D17) | 117.9 | 484.0 |
| (D15, D18) | 1085 | 481.0 |
| (D16, D18) | 429.6 | 492.3 |
| (D17, D18) | 998.7 | 489.5 |

Table 85: BN5: Probabilities, cost coefficients and income coefficients, Prospect $P_7$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K4, P8) | 0.809 | |
| (K6, P8) | 0.592 | |
| (P4, P8) | 0.784 | |
| (P8, D19) | 0.624 | |
| (P8, D20) | 0.603 | |
| (P8, D21) | 0.625 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D19 | 3936 | 4976 |
| D20 | 4024 | 5034 |
| D21 | 4030 | 4968 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D19, D20) | 480.0 | 504.2 |
| (D19, D21) | 1540 | 514.3 |
| (D20, D21) | 1748 | 491.0 |

Table 86: BN5: Probabilities, cost coefficients and income coefficients, Prospect $P_8$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K1, P9) | 0.844 | |
| (K3, P9) | 0.850 | |
| (P1, P9) | 0.267 | |
| (P3, P9) | 0.567 | |
| (P5, P9) | 0.892 | |
| (P9, D22) | 0.877 | |
| (P9, D23) | 0.635 | |
| (P9, D24) | 0.443 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D22 | 3951 | 5004 |
| D23 | 3965 | 5009 |
| D24 | 4119 | 4985 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D22, D23) | 1700 | 506.7 |
| (D22, D24) | 963.1 | 502.5 |
| (D23, D24) | 872.3 | 502.7 |

Table 87: BN5: Probabilities, cost coefficients and income coefficients, Prospect $P_9$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K3, P10) | 0.904 | |
| (P3, P10) | 0.921 | |
| (P4, P10) | 0.683 | |
| (P10, D25) | 0.501 | |
| (P10, D26) | 0.798 | |
| (P10, D27) | 0.707 | |
| (P10, D28) | 0.807 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D25 | 4007 | 4939 |
| D26 | 4046 | 5046 |
| D27 | 4028 | 5104 |
| D28 | 3956 | 4921 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D25, D26) | 599.3 | 487.0 |
| (D25, D27) | 1076 | 492.0 |
| (D26, D27) | 738.7 | 486.4 |
| (D25, D28) | 626.0 | 470.9 |
| (D26, D28) | 452.6 | 510.1 |
| (D27, D28) | 532.4 | 504.6 |

Table 88: BN5: Probabilities, cost coefficients and income coefficients, Prospect $P_{10}$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (P6, P11) | 0.587 | |
| (P7, P11) | 0.358 | |
| (P8, P11) | 0.838 | |
| (P11, D29) | 0.860 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D29 | 3962 | 4965 |

Table 89: BN5: Probabilities, cost coefficients and income coefficients, Prospect $P_{11}$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K2, P12) | 0.700 | |
| (K3, P12) | 0.805 | |
| (K4, P12) | 0.496 | |
| (P6, P12) | 0.781 | |
| (P7, P12) | 0.928 | |
| (P11, P12) | 0.548 | |
| (P12, D30) | 0.591 | |
| **Segment $S_i$** | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D30 | 3953 | 4964 |

Table 90: BN5: Probabilities, cost coefficients and income coefficients, Prospect $P_{12}$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (P2, P13) | 0.762 | |
| (P3, P13) | 0.956 | |
| (P6, P13) | 0.625 | |
| (P8, P13) | 0.789 | |
| (P10, P13) | 0.753 | |
| (P13, D31) | 0.846 | |
| (P13, D32) | 0.795 | |
| (P13, D33) | 0.716 | |
| (P13, D34) | 0.808 | |
| **Segment $S_i$** | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D31 | 4037 | 5070 |
| D32 | 3964 | 4989 |
| D33 | 4010 | 5076 |
| D34 | 3885 | 4979 |
| **Segment siblings $S_i, S_j$** | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D31, D32) | 262.9 | 499.0 |
| (D31, D33) | 747.7 | 497.8 |
| (D32, D33) | 644.5 | 504.2 |
| (D31, D34) | 825.0 | 510.3 |
| (D32, D34) | 30.57 | 496.0 |
| (D33, D34) | 873.2 | 504.3 |

Table 91: BN5: Probabilities, cost coefficients and income coefficients, Prospect $P_{13}$.

| Node pair $(i,j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K5, P14) | 0.807 | |
| (K6, P14) | 0.664 | |
| (K7, P14) | 0.841 | |
| (P1, P14) | 0.626 | |
| (P2, P14) | 0.934 | |
| (P8, P14) | 0.529 | |
| (P9, P14) | 0.354 | |
| (P10, P14) | 0.587 | |
| (P12, P14) | 0.934 | |
| (P13, P14) | 0.759 | |
| (P14, D35) | 0.921 | |
| (P14, D36) | 0.750 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D35 | 3986 | 4977 |
| D36 | 4018 | 4985 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D35, D36) | 2715 | 493.5 |

Table 92: BN5: Probabilities, cost coefficients and income coefficients, Prospect $P_{14}$.

| Node pair $(i, j)$ | $p_{e_{i,j}}$ | |
|---|---|---|
| (K5, P15) | 0.676 | |
| (P1, P15) | 0.718 | |
| (P4, P15) | 0.921 | |
| (P8, P15) | 0.479 | |
| (P12, P15) | 0.837 | |
| (P13, P15) | 0.855 | |
| (P14, P15) | 0.724 | |
| (P15, D37) | 0.913 | |
| (P15, D38) | 0.650 | |
| (P15, D39) | 0.873 | |
| (P15, D40) | 0.611 | |
| Segment $S_i$ | $\alpha_{cost}^{\vartheta(i)}$ | $\alpha_{inc}^{\vartheta(i)}$ |
| D37 | 3983 | 5030 |
| D38 | 3971 | 5035 |
| D39 | 4000 | 5008 |
| D40 | 3982 | 5028 |
| Segment siblings $S_i, S_j$ | $-\alpha_{cost}^{\vartheta(i),\vartheta(j)}$ | $\alpha_{inc}^{\vartheta(i),\vartheta(j)}$ |
| (D37, D38) | 524.5 | 505.4 |
| (D37, D39) | 972.3 | 493.8 |
| (D38, D39) | 372.8 | 509.1 |
| (D37, D40) | 1251 | 486.5 |
| (D38, D40) | 1257 | 508.7 |
| (D39, D40) | 289.9 | 495.8 |

Table 93: BN5: Probabilities, cost coefficients and income coefficients, Prospect $P_{15}$.