Aalto University
School of Science

Pekka Laitila

Improving the Use of Ranked Nodes in the Elicitation of Conditional Probabilities for Bayesian Networks

Master's Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology in the Degree Programme in Engineering Physics and Mathematics.

Espoo, September 16, 2013

Supervisor:	Professor Raimo P. Hämäläinen
Instructor:	D.Sc. (Tech) Kai Virtanen

AALTO UNIVERSITY
SCHOOL OF SCIENCE

ABSTRACT OF THE
MASTER'S THESIS

Author:	Pekka Laitila		
Title:	Improving the Use of Ranked Nodes in Elicitation of Conditional Probabilities for Bayesian Networks		
Date:	September 16, 2013	**Pages:**	123
Professorship:	Systems and Operations Research	**Code:**	Mat-2
Degree Programme:	Degree Programme in Engineering Physics and Mathematics		
Major subject:	Systems and Operations Research		
Minor subject:	Computational Science and Engineering		
Supervisor:	Prof. Raimo P. Hämäläinen		
Instructor:	Kai Virtanen, D.Sc. (Tech.)		

This thesis studies the ranked nodes method (RNM) developed to construct conditional probability tables (CPTs) to Bayesian networks (BNs) based on expert elicitation. RNM is used with BNs consisting of discrete random variables called ranked nodes. The idea of RNM is to generate a CPT based on parameters that are assessed by an expert and whose number is smaller than the number of elements in the CPT. In this thesis, RNM is explained more explicitly than in the existing literature and its properties are studied from both modeling and computational aspects.

The study on the modeling aspect of RNM interprets the properties of the method. The use of RNM is shown to approximate the use of a hierarchical Bayesian model of continuous random variables. While this finding helps to understand RNM, it is also utilized to explain results of an experimental study concerning the modeling accuracy of the method. Furthermore, exact interpretations are derived for the weight parameters used in RNM. In addition, the use of these interpretations in the transparent and consistent elicitation of the weights is introduced. The study also discusses the application of RNM when the random variables have interval or ratio scales — a theme which has not been addressed earlier.

The examination of the computational aspect of RNM consists of two experimental studies. In the first study, the calculation times of CPTs are measured. The results imply that CPTs of ordinary sizes are calculated within one second when using a standard desktop computer. In the second study, the modeling accuracy of RNM is explored by approximating CPTs found in real-life benchmark BNs. Though providing accurate approximations in some cases, the modeling accuracy is found to be generally poorer than that of another method examined in the literature. The result is considered to reflect the relative rarity of the probabilistic relationships compatible with the assumptions of RNM in the applications of BNs. On the other hand, the poorer modeling accuracy of RNM is suspected to be caused by the smaller amount of parameters. Overall, the results of the experimental studies imply that RNM provides means to readily construct CPTs that represent the probabilistic relationships of random variables in a coarse manner. These rough CPTs can then be used as initial probability estimates in an iterative elicitation process based on, e.g., the sensitivity analysis of a BN.

Keywords: Bayesian networks, Influence diagrams, Probability elicitation, Ranked nodes

AALTO-YLIOPISTO
PERUSTIETEIDEN KORKEAKOULU | DIPLOMITYÖN
TIIVISTELMÄ

Tekijä:	Pekka Laitila		
Työn nimi:	Järjestysperusteisten solmujen käytön parantaminen Bayes-verkkojen ehdollisten todennäköisyyksien arvioimisessa		
Päivämäärä:	16. syyskuuta 2013	**Sivumäärä:**	123
Professuuri:	Systeemi- ja operaatiotutkimus	**Koodi:**	Mat-2
Tutkinto-ohjelma:	Teknillisen fysiikan ja matematiikan tutkinto-ohjelma		
Pääaine:	Systeemi- ja operaatiotutkimus		
Sivuaine:	Laskennallinen tiede ja tekniikka		
Valvoja:	Prof. Raimo P. Hämäläinen		
Ohjaaja:	TkT Kai Virtanen		

Tässä työssä tarkastellaan järjestysperusteisten solmujen menetelmää (JSM), jolla muodostetaan ehdollisten todennäköisyyksien taulukoita (ETT) Bayes-verkkoihin (BV) perustuen asiantuntija-arvioihin. JSM soveltuu järjestysperusteisiksi solmuiksi kutsutuista diskreeteistä satunnaismuuttujista koostuvien Bayes-verkkojen (BV) todennäköisyyksien arviointiin. JSM:ssä ETT:t perustuvat asiantuntijan arvioimiin parametreihin, joiden lukumäärä on ETT:n alkioiden lukumäärää huomattavasti pienempi. Tässä työssä JSM:n toimintaperiaate esitellään aiempaa kirjallisuutta täsmällisemmin ja sen ominaisuuksia tarkastellaan yhtäältä mallinnuksellisesta ja toisaalta laskennallisesta näkökulmasta.

Mallinnusnäkökulmaan liittyvässä tarkastelussa selvennetään JSM:n toimintaperiaatetta. Menetelmän käytön osoitetaan approksimoivan jatkuvista satunnaismuuttujista koostuvan hierarkkisen Bayes-mallin käyttöä. Tämä havainto auttaa ymmärtämään menetelmän toimintaperiaatetta ja sitä käytetään myös selittämään menetelmän mallinnustarkkuutta käsittelevän kokeen tuloksia. Lisäksi työssä johdetaan JSM:ssä käytettäville painoparametreille tulkinnat ja esitellään niiden hyödyntäminen painojen läpinäkyvässä ja johdonmukaisessa arvioimisessa. Havainnollistavan esimerkin avulla tarkastellaan JSM:n soveltamista välimatka- tai suhdeasteikollisiin satunnaismuuttujiin. Tätä teemaa ei ole ole käsitelty kirjallisuudessa aiemmin.

JSM:n tarkastelu laskennallisesta näkökulmasta koostuu kahdesta kokeesta. Ensimmäisessä mitataan erikokoisten ETT:iden laskenta-aikoja. Kokeen tulokset osoittavat, että JSM:llä kyetään laskemaan tyypillisen kokoisia ETT:itä sekunnissa tavanomaisella pöytätietokoneella. Toisessa kokeessa JSM:n mallinnustarkkuutta tutkitaan approksimoimalla BV-sovelluksista löytyviä ETT:ita. Vaikka JSM tuottaa tarkkoja approksimaatioita joissain tapauksissa, sen mallinnustarkkuuden todetaan olevan yleisesti heikompi kuin toisella kirjallisuudessa esitetyllä menetelmällä. Tulokset viittaavat siihen, että JSM:n oletusten kanssa yhteensopivat satunnaismuuttujien riippuvuustyypit ovat harvinaisia BV:iden sovelluksissa. Toisaalta JSM:n heikompi mallinnustarkkuus johtuu siitä, että siinä käytetään vähemmän parametreja. Työn laskennallisen osan tulokset osoittavat, että JSM:llä kyetään muodostamaan nopeasti ETT:ita, jotka kuvaavat satunnaismuuttujien välisiä riippuvuussuhteita karkeasti. Näitä ETT:ita voidaan tarkentaa perustuen esimerkiksi BV:n herkkyysanalyysiin.

Avainsanat: Bayes-verkot, Vaikutuskaaviot, Todennäköisyyksien arviointi, Järjestysperusteiset solmut

Acknowledgements

I wish to thank my instructor, D.Sc. (Tech.) Kai Virtanen for the comprehensive instruction and guidance during the writing of this thesis. My supervisor, Professor Raimo P. Hämäläinen, I wish to thank for the patience he has showed regarding the completion of this thesis as well as for the valuable advice he provided at the end of the long writing process.

Sharing an office with Heikki Puustinen and Eero Rantala has meant that I don't need to fetch help for various technical challenges from far. Moreover, it has been a highly valued privilege for me to share an easy and relaxed spirit in the office with these guys, even on the most hectic days. I also wish to thank Jouni Pousi for his versatile help and support at various phases of a work project connected to this thesis. I am grateful for all my collaborators at the Systems Analysis Laboratory for creating a positive work environment.

During the writing of this thesis, my friends have a been a huge source of strength and happiness for me. So, I sincerely thank all my mates on the path of budo and on other routes of life for the good times and shared experiences.

To my parents and my brother I also owe a big praise. The diligence that was needed in the writing of this thesis I feel to have acquired from home. On the other hand, I thank my family for giving me the basis to lead a good life where the success of a person is not determined by his or her work achievements.

Espoo, September 16, 2013

Pekka Laitila

Contents

1 Introduction **1**

1.1 Bayesian Network Representation 1

1.2 Existing Approaches to Probability Elicitation for Bayesian Networks 3

1.3 Contribution 5

1.4 Structure 6

2 Bayesian Networks **7**

2.1 Definition 7

2.2 Probabilistic Inference 10

3 Ranked Nodes Method (RNM) **13**

3.1 Origin and Basic Idea 13

3.2 Ranked Nodes 14

3.3 Functioning of RNM 15

3.3.1 Mapping of States to State Intervals 16

3.3.2 Weight Expressions 17

3.3.3 Weights of Parent Nodes 20

3.3.4 Variance Parameter 21

3.3.5 Computation of Conditional Probability Tables 21

3.4 Analogy to Linear Regression 27

3.5 Extensions to RNM 28

3.5.1 Non-Monotonic Interaction 28

3.5.2 Partitioned Expressions 29

3.6 Benefits of RNM 30

3.7 Issues on RNM 32

4 On Modeling Aspect of RNM **34**

4.1 Interpretation of Scale [0, 1] . . . 34
4.2 Interpretation of Weights . . . 37
4.2.1 WMEAN . . . 38
4.2.2 WMIN . . . 38
4.2.3 WMAX . . . 40
4.2.4 MIXMINMAX . . . 42
4.2.5 Use of Interpretations in Elicitation of Weights . . . 42
4.2.6 Conceptual Challenges . . . 44
4.3 Variables with Interval or Ratio Scales as Ranked Nodes: Illustrative Example . . . 45
4.3.1 Discretization of Interval and Ratio Scales . . . 46
4.3.2 Revised Interpretation of Scale [0, 1] . . . 49
4.3.3 Elicitation of Weights . . . 50
4.3.4 Inconsistency in Elicitation . . . 53
4.4 Comparison to Other Canonical Models . . . 57

5 On Computational Aspect of RNM 61
5.1 Experimental Study on Computational Complexity of RNM . . . 61
5.1.1 Experimental Design . . . 62
5.1.2 Results . . . 64
5.1.3 Summary . . . 70
5.2 Experimental Study on Modeling Accuracy of RNM . . . 71
5.2.1 Experimental Design . . . 71
5.2.2 Results . . . 78
5.2.3 Summary . . . 98

6 Conclusion 101

Bibliography 104

Appendices 111

A Distributions of Nodes in Figure 2.1 111

B Proof of Equation 4.5 113

C Results of Experiment in Section 5.1 **121**

Chapter 1

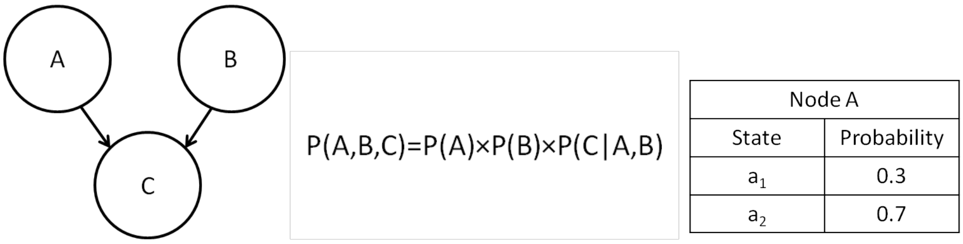
Introduction

This thesis studies the ranked nodes method (RNM, [25]) developed to construct conditional probability tables (CPTs) to Bayesian networks (BNs, e.g., [41, 45]) based on expert elicitation. RNM is designed for BNs consisting of discrete random variables called ranked nodes [25]. The thesis contains a more exact and clearer presentation of RNM than is available in the earlier literature discussing the method [25, 26, 29]. On the basis of this presentation, RNM is examined from both modeling and computational aspects. The results enhance the understanding of the properties of RNM as well as the comprehension of the possibilities and limitations related to it. Moreover, the new results developed in the thesis provide practical improvements to RNM in the construction of CPTs for BNs and for the decision analytic extension of BNs — influence diagrams — as well.

1.1 Bayesian Network Representation

BNs, see, e.g., [41] and [45], are used in many areas to represent uncertain knowledge and reasoning with it. These include, e.g., medical diagnosis [33, 42], hardware troubleshooting and diagnosis [32], as well as military applications [21, 47]. As a subclass of influence diagrams [41], BNs provide both graphical and numerical representation for the probabilistic relationship between random variables. The difference between influence diagrams and BNs is that, whereas in the former the decisions and the values of the related outcomes are explicitly presented as decision and value nodes, the latter consists of only chance nodes associated with random variables. If the states of some chance nodes in the BN are known, the probability distributions of the others can be updated accordingly. This kind of probabilistic inference helps one to make conclusions and decisions about the system under consideration. In this sense, BNs act as a tool supporting decision making, problem solving, and risk analysis, similarly as the influence diagrams.

A BN is a directed acyclic graph of random variables that are represented by chance nodes. As chance nodes are the only type of nodes in BNs, they are referred to as only nodes from now on. The BN implies the interrelationships of the nodes through three levels of specification displayed in Figure 1. The graphical level presents the nodes and directed arcs between directly interdependent nodes. The functional level of specification defines the conditional and joint probability distributions of the nodes in an algebraic fashion. The numerical level specifies the actual numbers associated to the functional level.



$P(A,B,C)=P(A)\times P(B)\times P(C|A,B)$

Node A	
State	Probability
a_1	0.3
a_2	0.7

(a) Relational level (b) Graphical level (c) Numerical level

Figure 1: Three levels of a BN.

Figure 1a displays a group of nodes consisting of a single ”child node” and its ”parent nodes”. Usually, the nodes have discrete states and a CPT is used to define the probabilistic relationship between the child node and the parent nodes. A CPT contains the conditional distributions of the child node for every possible combination of the states of the parent nodes. The structure of the CPTs contains information needed to form mathematical clauses on the functional level of the BN. On the other hand, the actual numbers contained in the CPTs form the basis of the numerical level of the BN. In many practical settings, there is not enough statistical data or other necessary probabilistic information to define the distributions in the CPTs. In this case, domain experts are the only source for obtaining the probabilities. The process of acquiring subjective views of the probabilities from experts is often referred to as expert or probability elicitation [25, 49]. The method studied in this thesis, RNM, is developed to aid the elicitation of probabilities from experts.

1.2 Existing Approaches to Probability Elicitation for Bayesian Networks

In practice, probability elicitation for a BN is a challenging task. Acquiring probabilities from people is acknowledged to be problematic in general [49, 12]. People, even experts of the subject matter at hand, often find it difficult to assess probabilities and therefore are prone to use heuristics, which leads often to biased and poorly calibrated assessments [49]. People tend to, e.g., assess the probability of an event by the ease with which instances or occurrences of it can be brought to mind. To overcome, or at least suppress, the problems of bias and poor calibration of the probability assessments, a number of probability elicitation techniques, such as probability wheel and reference lottery, are used in the fields of decision and risk analysis, see, e.g., [40] and [49] for a review. However, these conventional techniques tend to be time consuming and obtaining a single probability assessment can take up to 30 minutes [19]. The size of a CPT grows exponentially with the number of parent nodes and the amount of probabilities needed to assess for a BN may be up to hundreds or thousands. Thus, the use of the conventional elicitation techniques may cause insuperable time problems and they are generally acknowledged to be inadequate for the probability elicitation of BNs [19, 49, 54]. Another issue related to elicitation of the CPTs is that some of the combinations of states of the parent nodes may be hard to imagine by the expert. In such cases, the expert may feel reluctant to give exact numbers having a high level of accuracy due to feeling incapable of doing it [11].

To deal with the challenges related to the probability elicitation of BNs, the concept of iterative probability elicitation is introduced in [11]. In [11], it is argued that the elicitation of probabilities from experts can be supported to a large extent with iterative sensitivity analysis starting with rough initial probability assessments. The analysis gives insight into which probabilities require a high level of accuracy and therefore should have the elicitation be focused more carefully. With this kind of approach, one can ease up the burden of elicitation and also save time as the expert does not need to ponder every single probability assessment so carefully. An elicitation technique aimed at getting the rough initial assessments is presented in [54]. The technique combines qualitative and quantitative information and is reported to require little time from the experts in their assessments. However, the technique does not reduce the amount of probabilities to be elicited which is marked to be a drawback in some contexts [13]. For example, the military experts operate under time constraints and may not be keen to assess a large set of probability distributions, no matter how little time and effort each assessment requires.

In [19], it is suggested that the cognitive workload of an expert should primarily be reduced by introducing methods that require less probability assessments and by developing tools that support the quantification of the assessments. In the context of BNs, the amount of probabilities needed to elicit from an expert is reduced in two ways. One way is to alter and redesign the structure of the network. This can be conducted, e.g., using the principle of divorcing parents by introducing intermediate variables [42], and removing arcs representing weak dependencies [55]. Another way to reduce the number of probabilities assessed is to use so-called canonical models [45]. In canonical models, underlying assumptions are made concerning the probabilistic relationships. Using these assumptions, the CPT is generated algorithmically from parameters that are assessed by an expert and whose amount grows only linearly with the number of parent nodes. The parameters are usually conditional probabilities related to some particular combinations of states of the parent nodes. Distributions obtained with the canonical models are sometimes also referred to as parametric probability distributions [19]. Well-known examples of canonical models are noisy-OR [45] for binary nodes and its generalization called noisy-MAX [17, 52]. A review of the canonical models is found in [18].

The use of canonical models is limited by the underlying assumptions. In most of the canonical models, parent nodes are considered as cause variables and the child node as an effect variable. Most of such models are also based on the assumption of independence of causal influence of the parent nodes [31, 30, 18]. This means that the parent nodes are independent of each other in their ability to affect the child node and that there is no synergy between them. In addition, it is often assumed that each variable has a distinguished state, usually attributed to *absence* or *false*, and that at least some causes must be capable of producing the presence of the effect although all the other causes are absent. Because of such underlying assumptions, the use of canonical models is not always possible. Moreover, in [57], the parameters of some canonical models are judged to be too hard to perceive to make their elicitation reasonable.

Despite their limitations, canonical models are found to be practical and suitable for many applications. The main reason for their popularity is the relief given to the cognitive workload of the expert in the form of reduced amount of parameters. The parameters of the canonical models are also often considered to be more intuitive and easier to assess than direct entries of CPTs [18]. Moreover, there is little insight into whether or not a BN with separately specified probability assessments is more trustworthy than a BN in which parametric probability distributions are used [19, 58].

In [59], there is a report of a study in which a test group is allowed to experiment a stochastic model obeying noisy-OR distributions. Each member in the test group forms opinion about the underlying probabilities of the model and these subjective probabilities are then elicited from them directly in the form of CPTs as well as noisy-OR parameters. It turns out that when measuring their Euclidean distance from the true distributions, the probability distributions generated with the noisy-OR parameters are closer than those elicited directly. In any case, with the canonical models less time is required for elicitation and the saved time can be exploited for verifying and refining both the structure and the probability distributions of the BN. Therefore, one can use the canonical models to produce the rough initial probability estimates for the iterative elicitation process introduced in [11].

1.3 Contribution

This thesis deals with a canonical model designed for so called *ranked nodes* introduced in [25]. Ranked nodes are random variables with discrete ordinal scales that can be considered as subjective abstractions of some underlying continuous quantities. The canonical model presented in [25] is designed to generate a CPT in the case where the parent nodes and the child node are all ranked nodes. In [25], the canonical model is not referred to with any particular name, but in this text, the term *Ranked Nodes Method* (RNM) is used. In RNM, the expert assigns weights describing the relative strengths of influence of the parent nodes on the child node. The discrete states of the nodes are identified with subintervals on the normalized scale $[0, 1]$, and the conditional probability distribution of the child node is obtained by appropriately weighing the states of the parent nodes.

RNM has been implemented in AgenaRisk software [38] and the method has been used in real-life applications [3, 7, 15, 23, 24, 46]. Nevertheless, the exact functioning of RNM has not been documented in the earlier literature. In [25] [26], and [29], RNM is described with varying precision. However, the method has not been presented in any of them thoroughly enough for implementing it in practice. Related to this, there is a lack of discussion concerning the properties of RNM and hence, insufficient understanding of its possibilities and limitations.

In this thesis, RNM is presented for the first time with a precision that provides means for implementing it in practice. Based on this presentation, both modeling and computational aspects of RNM are studied. The study dealing with the modeling aspect includes interpreting the use of the normalized scale $[0, 1]$ and showing that the use of RNM approximates the use of a hierarchical Bayesian model of continuous random variables. Moreover, exact interpretations are derived for the weights used in RNM and the use of these interpretations in the elicitation of the weights is discussed. In addition, the thesis discusses through an illustrative example how to apply RNM to ranked nodes which have ratio or interval scales representing the underlying continuous quantities. The study of RNM on the modeling aspect also includes discussing its relation to other canonical models. The examination of RNM computationally consists of two experimental studies. In the first one, the computational complexity of the method is investigated by measuring the calculation times of CPTs with a self-made implementation of RNM. The aim of this study is to clarify the limits in which the use of RNM is sensible from the point of view of calculations times of the CPTs. The second experimental study examines the modeling accuracy of RNM and the effect of various computational configurations on it. In this study, CPTs in existing real-life benchmark BNs are approximated using RNM similarly to an experiment in [58] concerning noisy-MAX. Investigating RNM gives insights to how common the probabilistic relationships described by CPTs obtained with RNM actually are and how accurate CPTs the method can be expected to generate.

1.4 Structure

The thesis is organized as follows. Chapter 2 gives a short introduction to BNs and their properties. RNM is thoroughly presented in Chapter 3 along with discussion on its properties. In Chapter 4, RNM is studied and discussed from the viewpoint of the modeling aspect. In turn, Chapter 5 contains two empirical studies related to the computational features of RNM. Finally, concluding remarks and themes of further research are given in Chapter 6.

Chapter 2

Bayesian Networks

A Bayesian network (BN) is a mathematical model that represents the probabilistic relationships of random variables both graphically and numerically. This chapter contains basic theory related to BNs and lays the foundation for understanding the canonical model presented in Chapter 3 and discussed for the rest of the thesis. Section 2.1 gives a formal definition of BNs. Section 2.2 presents computational mechanisms for probabilistic inference enabling the use of BNs in supporting decision making, problem solving, and risk analysis. Overviews of BNs are found, e.g., from [41] and [45].

2.1 Definition

There are different but equivalent definitions for the BN. In [45], the definition of the BN is based on the concepts of *d-separation* and *I-map*. In [41], the concept of *Markov condition* is used in the definition of the BN. Because these theoretical concepts are not in focus in the thesis, it is now preferred to give a more descriptive definition of the BN. The definition below follows that in [51].

Definition 1. *A Bayesian network* is a directed graph in which each node is annotated with quantitative probability information. The full specification is as follows:

1. A set of random variables makes up the nodes of the network. Variables may have a finite number of discrete states. Alternatively, the variables may have an infinite amount of states corresponding to the points of a continuous scale.

2. A set of directed links or arcs connects pairs of nodes. If there is an arc from node X to node Y, X is said to be a parent node of Y and Y is said to be a child node of X.

3. If the node X_i does not have any parent nodes, a marginal probability distribution $P(X_i)$ is associated with it. These types of nodes are called root nodes. If X_i has parent nodes $\mathbf{Pa}_i$, then a set of conditional distributions $P(X_i|\mathbf{Pa}_i)$ is associated with it. When the nodes have discrete states, $P(X_i|\mathbf{Pa}_i)$ is represented by a conditional probability table (CPT). The CPT specifies the conditional probability distributions of the dependent node, known as the child node, for each combination of states of the parent nodes.

4. The graph has no directed cycles.

The study in this thesis is concentrated to BNs consisting of discrete nodes. Hence, the future presentation of BNs is written from the point of view of discrete nodes using the corresponding mathematical notation.

For a BN consisting of nodes $\mathbf{X} = \{X_i\}_{i=1}^n$, the joint probability $P(\mathbf{X} = \mathbf{x})$ is obtained with the formula [45, 51, 41]

$$P(\mathbf{X} = \mathbf{x}) = \prod_{i=1}^{n} P(X_i = x_i|\mathbf{Pa}_i = \mathbf{pa}_i). \tag{2.1}$$

Equation 2.1 implies that the joint probability distribution of the whole BN can be represented by using the marginal distributions of the root nodes and the CPTs of the other nodes. Thus, the BNs provide a dexterous way to store and represent the joint probability distribution of random variables [51].

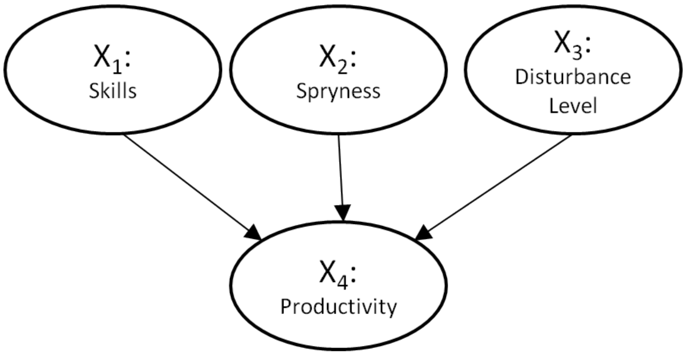


Figure 2.1: Example Bayesian network modeling the productivity of an employee at a workplace.

Figure 2.1 presents an example BN that models the work productivity of a single employee. There are three parent nodes: *Skills*, *Spryness*, and *Disturbance Level*. *Skills* presents the professional skills of the employee. *Spryness* describes how brisk or tired the employee is. *Disturbance Level* illustrates the level of disruption affecting the employee. *Productivity* is the child node of the previous three and describes the work productivity of the employee. All the nodes have states $\{High, Medium, Low\}$. The probability distributions associated with the nodes are presented in Tables A-1–A-4 of Appendix A.

According to [16], the BN implies the interrelationships of the nodes through three levels of specification. The graphical level presents all the nodes and the arcs between directly interdependent nodes. It is the graphical level of the example BN that is displayed in Figure 2.1. The functional level of specification defines the states of the nodes and thus implies the probability distributions associated with the nodes algebraically. On the functional level, one can form mathematical clauses about the probabilistic relationships the BN implies. For example, by applying Equation 2.1, the clause for a particular joint probability $P(X_1 = Medium, X_2 = Low, X_3 = Low, X_4 = High)$ in the example BN is given by

$$\begin{aligned}
&P(X_1 = Medium, X_2 = Low, X_3 = Low, X_4 = High) = \\
&P(X_1 = Medium, X_2 = Low, X_3 = Low)* \\
&P(X_4 = High|X_1 = Medium, X_2 = Low, X_3 = Low) = \\
&P(X_1 = Medium)P(X_2 = Low)P(X_3 = Low)* \\
&P(X_4 = High|X_1 = Medium, X_2 = Low, X_3 = Low), \qquad (2.2)
\end{aligned}$$

where the joint distribution $P(X_1 = Medium, X_2 = Low, X_3 = Low)$ is equal to the product $P(X_1 = Medium)P(X_2 = Low)P(X_3 = Low)$ because the parent nodes X_1, X_2, and X_3 are independent of each other.

The third level, i.e., the numerical level, specifies the numbers in the probability distributions associated with the nodes. Through the numerical level, the clauses on the functional level get real values associated with them. For example, using the numbers in Tables A-1–A-4 of Appendix A, Equation 2.2 gives

$$\begin{aligned}&P(X_1 = Medium, X_2 = Low, X_3 = Low, X_4 = High) =\\&P(X_1 = Medium)P(X_2 = Low)P(X_3 = Low)*\\&P(X_4 = High|X_1 = Medium, X_2 = Low, X_3 = Low) =\\&0.60 * 0.25 * 0.167 * 0.142 = 0.0036.\end{aligned}$$

The direction of the arc between two directly interdependent nodes can be freely decided. In practice, the direction of the arc can depend on different factors. If the parent node and the child node describe a causal dependence, it may feel natural that the arc goes from the cause to the effect. Sometimes the probabilistic information available may define the direction of the arc. For example, there might be data on the probability of the presence of a certain disease given certain symptoms but no data on the probabilities of the occurrence of different symptoms when a certain disease is present. In the absence of data, the direction of the arc may be decided based on which way it is easier to assess the conditional probabilities.

2.2 Probabilistic Inference

As a BN represents the probabilistic relationships between its variables, it is used to answer probabilistic queries about them. The queries may concern the joint distribution of the whole network as well as marginal and conditional distributions of specific nodes.

The joint probability distribution of the whole BN is obtained by applying Equation 2.1. The marginal distribution of a single node is obtained by repeatedly summing conditional probabilities over the state combinations of the parent nodes. The terms to be summed are weighted with the probability of the state combination of the parent nodes. Calculating the marginal distribution is called marginalization. For example, in the example network, the probability for *Productivity* (X_4) to be in state *High* is

$$
\begin{aligned}
&P(X_4 = High) = \sum_{\mathbf{pa}_4} P(X_4 = High|\mathbf{Pa}_4 = \mathbf{pa}_4)P(\mathbf{Pa}_4 = \mathbf{pa}_4) = \\
&\sum_{x_1}\sum_{x_2}\sum_{x_3} P(X_4 = High|X_1 = x_1, X_2 = x_2, X_3 = x_3)P(X_1 = x_1, X_2 = x_2, X_3 = x_3) = \\
&\sum_{x_1}\sum_{x_2}\sum_{x_3} [P(X_4 = High|X_1 = x_1, X_2 = x_2, X_3 = x_3)] * \\
&P(X_1 = x_1)P(X_2 = x_2)P(X_3 = x_3) = 0.20, \qquad (2.3)
\end{aligned}
$$

where $\sum_{x_i}$ refers that the sum is taken over all the states of the node X_i.

If some nodes in the BN are fixed to be in certain states, the marginal distributions of other nodes can be updated based on this set evidence. Calculation of the updated marginal distributions is called belief updating [45] or probabilistic inference [41]. For example, if *Skills* (X_1) is known to be in state Low, the posterior probability for *Productivity* (X_4) to be in state $High$ is obtained by marginalizing over the states of *Spryness* (X_2) and *Disturbance Level* (X_3) as follows:

$$
\begin{aligned}
&P(X_4 = High|X_1 = Low) = \\
&\sum_{x_2}\sum_{x_3} P(X_4 = High|X_1 = Low, X_2 = x_2, X_3 = x_3)P(X_2 = x_2, X_3 = x_3|X_1 = Low) = \\
&\sum_{x_2}\sum_{x_3}[P(X_4 = High|X_1 = Low, X_2 = x_2, X_3 = x_3)]* \\
&P(X_2 = x_2)P(X_3 = x_3) = 0.01. \qquad (2.4)
\end{aligned}
$$

Here, the probability values in Appendix A are applied again. In addition, the independence of the parent nodes X_1, X_2, and X_3 is also utilized.

The directions of the arcs in BNs define the forms of the CPTs in the sense that the conditional distributions of the child node are displayed. However, probabilistic queries can be conducted against the directions of the arcs. For example, if it is known that *Productivity* (X_4) is in state $High$, the posterior probability for *Skills* (X_1) to be in state Low is calculated by applying Bayes' rule [51]

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \qquad (2.5)$$

which yields

$$\begin{aligned} P(X_1 = Low|X_4 = High) &= \frac{P(X_4 = High|X_1 = Low)P(X_1 = Low)}{P(X_4 = High)} \\ &= \frac{0.01 * 0.15}{0.20} = 0.008, \end{aligned} \tag{2.6}$$

where $P(X_4 = Low)$ is calculated according to Equation 2.3 and $P(X_4 = High|X_1 = Low)$ according to Equation 2.4. $P(X_1 = Low)$ is from Table A-1 in Appendix A.

In real-life applications, probabilistic inference is computerized using various inference algorithms [41, 51]. Some of the inference algorithms are exact in the sense that no stochastic sampling is involved with them. Inference algorithms based on stochastic sampling are referred to as approximate. A challenge is the computational complexity. In the worst case, inference algorithms are NP-hard referring to the complexity of the task in terms of nondeterministic polynomial time [8]. However, there are several exact inference algorithms that allow probabilistic inference in BNs consisting of tens or hundreds of nodes to be tractable, see, e.g., [36] and [44]. In most of the applications, the BNs are not bigger than this and thus, the computerized Bayesian updating is rapid, taking between a fraction of a second and a few seconds. There are several software that enable the construction and manipulation of BNs through a graphical user interface. Examples are commercial Hugin [2] and Netica [9], as well as GeNIe [14] which is free to download from the internet.

BNs and the related software provide a useful way to model complex systems of random variables and investigate their dependencies in an illustrative way. The possibility to build the network by oneself is often helpful in perceiving the structural nature of the system under consideration. This feature is discussed, e.g., in [5], [11], and [35]. The analysis of the probabilistic dependencies between the variables is relieved by the ability of the software to carry out probabilistic inference and present the updated marginal distributions. For this reason, BNs have been exploited in various practical problems, see, e.g., [21], [23], [47], and [54].

Chapter 3

Ranked Nodes Method (RNM)

The ranked nodes method (RNM, [25]) is a canonical model designed to aid in the construction of CPTs for a type of random variables called ranked nodes whose parent nodes are also ranked nodes. This chapter presents RNM in detail and motivates the study of its properties in Chapters 4 and 5. The presentation of RNM provided in this chapter is the first time when the method is explained meticulously enough for its implementation in practice. The chapter begins with Section 3.1 presenting the origin and the basic idea of RNM. The concept of ranked nodes is explained more properly in Section 3.2. Section 3.3 presents a detailed description of RNM and Section 3.4 discusses its analogy to linear regression. Section 3.5 presents means that diversify the use of RNM. The benefits of RNM are discussed in Section 3.6 whereas Section 3.7 presents issues on RNM that motivate the more elaborative studies of the method in Chapters 4 and 5.

3.1 Origin and Basic Idea

RNM evolved over a number of years from the process of engaging with domain experts in various practical applications of BNs [25]. In many of the BNs, there have been nodes with states measurable only on a subjective scale like *Very Low, Low, Medium, High, Very High* and only limited statistical data, if any, has been available to inform the probabilistic relationship between the nodes. Yet, there has been significant expert subjective judgement to utilize. While assessing the CPTs, rather than considering numerical probabilities, the experts have usually found it easier to understand and express the interaction of the nodes with assertions about the ″central tendency″ of the child node [25]. Here, the term central tendency refers to a qualitative description of how the unimodal probability distribution is clustered over the states of the child node. An example of the form of statements given is ”When parent node X_1 is in the state *Very Low* and parent node X_2 is in the state *Very High*, child node Y is centred below the state *Medium*”. Often, the experts see the central tendency of the child node on its ordinal scale forming as some kind of a weighted combination of the relative states of the parent nodes on their ordinal scales [25]. Based on this mental weighting scheme, they can give a series of assertions describing the central tendency of the child node for a given combination of states of the parent nodes.

RNM was developed to model quantitatively the mental weighting scheme of the experts. The basis of RNM is to map the states of the nodes into a numerical scale. By using the numerical scale and weights assigned to the parent nodes, the central tendency of the child node is calculated for a given combination of states of the parent nodes. The less precise the central tendency of the child node is considered, the vaguer distribution it will receive.

3.2 Ranked Nodes

In [25] that introduces the concept of ranked nodes, they are said to represent qualitative variables that are abstractions of some underlying continuous quantities. More specifically, ranked nodes are defined in [25] as discrete random variables whose states are expressed on an ordinal scale that can be mapped onto a bounded continuous scale. The continuous scale used in [25] is the unit interval $[0, 1]$ and the states of a ranked node correspond to subintervals of equal width on $[0, 1]$. In [26], ranked nodes are explained to represent real-world variables that are typically measured on a discrete subjective scale. Usually, either 3, 5 or 7 point scales are used for the ranked nodes. In the 3 point scale, the states could be $\{High, Medium, Low\}$. On the 5 and 7 point scales, the ranges could be from $Very\ High$ to $Very\ Low$ and from $Highest$ to $Lowest$, respectively.

The nodes in the example BN in Figure 2.1 are examples of ranked nodes. For each of these nodes, the states of the 3 point scale $\{High, Medium, Low\}$ are subjective in nature and can be considered to represent the discretization of a continuous quantity. For example, one might say that there are different kinds of degrees of spryness that all can be summoned under the state Low of the node *Spryness*. Considering the unit interval $[0, 1]$, the states of a given node in the example BN could be mapped onto subintervals $[0, 1/3)$, $[1/3, 2/3)$, and $[2/3, 1]$ or $[2/3, 1]$, $[1/3, 2/3)$, $[0, 1/3)$, respectively. Whether the first subinterval is starting from 0 or ending to 1 depends on the context. Now, a logical mapping is the one depicted in Figure 3.1. That is, (Low, $Medium$, $High$) are mapped onto ($[0, 1/3)$, $[1/3, 2/3)$, $[2/3, 1]$) for *Skills*, *Spryness*, and *Productivity*, whereas for *Disturbance Level* the same states would be mapped onto ($[2/3, 1]$, $[1/3, 2/3)$, $[0, 1/3)$), respectively. With this mapping, the subinterval ending to 1 corresponds to the best state for all the nodes. This kind of consistency in the mapping is required when using RNM. In the rest of the thesis, the unit interval $[0, 1]$ is referred to as the *normalized scale*. In addition, a subinterval of $[0, 1]$ corresponding to a certain state of a ranked node is referred to as a *state interval*.

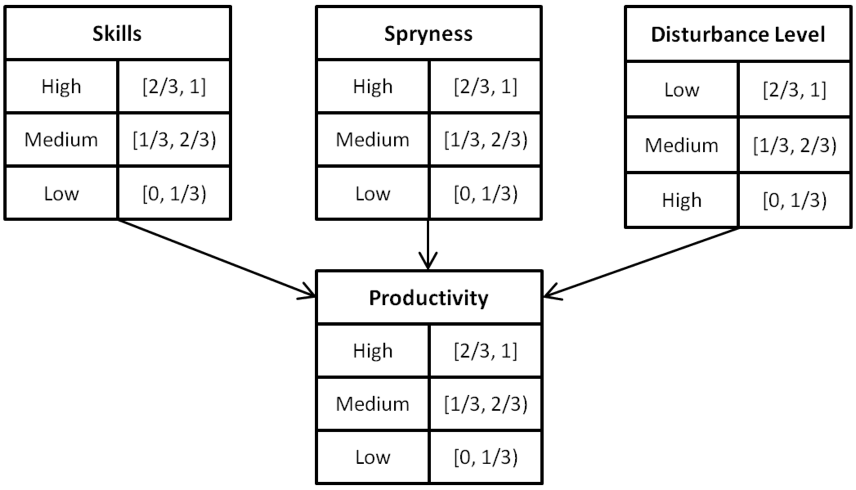


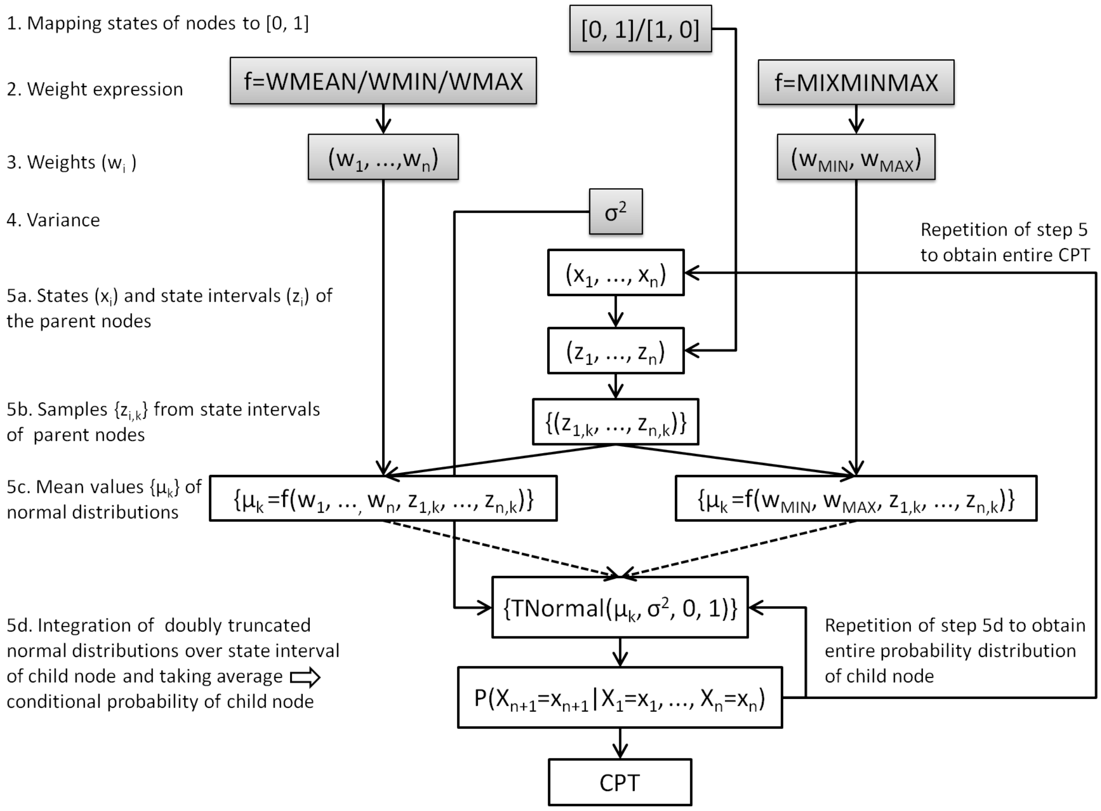
Figure 3.1: Example network of Figure 2.1 with the state intervals corresponding to the states of each variable.

3.3 Functioning of RNM

In its basic form, the generation of a CPT for a child node using RNM consists of five major steps:

1. Mapping states of the nodes to state intervals
2. Selecting a weight expression
3. Assigning weights to the parent nodes
4. Assigning value to a variance parameter
5. Computation of the CPT using the above settings

The first four steps require involvement of the expert whereas the fifth step is carried out automatically. Figure 3.2 presents RNM in a conceptual diagram when a child node has n parent nodes. In Figure 3.2, steps 1–4 are displayed as grey boxes whereas the parts related to step 5 are white boxes. The five main steps of RNM are next thoroughly explained in separate sections. The example BN in Figure 2.1 is used to demonstrate the execution of the steps in practice.



1. Mapping states of nodes to [0, 1]
2. Weight expression
3. Weights (w_i)
4. Variance
5a. States (x_i) and state intervals (z_i) of the parent nodes
5b. Samples {z_i,k} from state intervals of parent nodes
5c. Mean values {μ_k} of normal distributions
5d. Integration of doubly truncated normal distributions over state interval of child node and taking average ⇨ conditional probability of child node
[0, 1]/[1, 0]
f=WMEAN/WMIN/WMAX
f=MIXMINMAX
(w_1, ...,w_n)
(w_MIN, w_MAX)
σ^2
Repetition of step 5 to obtain entire CPT
(x_1, ..., x_n)
(z_1, ..., z_n)
{(z_1,k, ..., z_n,k)}
{μ_k =f(w_1, ..., w_n, z_1,k, ..., z_n,k)}
{μ_k =f(w_MIN, w_MAX, z_1,k, ..., z_n,k)}
{TNormal(μ_k, σ^2, 0, 1)}
Repetition of step 5d to obtain entire probability distribution of child node
P(X_n+1=x_n+1|X_1=x_1, ..., X_n=x_n)
CPT

Figure 3.2: Conceptual diagram of RNM when a child node has n parent nodes. The grey boxes require involvement of the expert. The dashed lines indicate that either the weights $(w_1, ..., w_n)$ or (w_{MIN}, w_{MAX}) are used.

3.3.1 Mapping of States to State Intervals

The first step in RNM is to decide how the states of the ranked nodes are mapped into the state intervals. As explained in Section 3.2, the ordered states of the ranked node are always identified with consecutive subintervals of same width on [0, 1]. However, the direction of this mapping has to be defined. That is, one must decide whether, e.g., the states Low, $Medium$, $High$ are identified to the state intervals $[0, 1/3)$, $[1/3, 2/3)$, and $[2/3, 1]$ or $[1, 2/3]$, $(2/3, 1/3]$, and $(1/3, 0]$, respectively. The directions of the mappings for different nodes should be sensible with respect to each other from the point of view of the probabilistic relationship between the parent nodes and the child node. The way the states of the nodes of the example BN are mapped into state intervals in Figure 3.1 demonstrates a sensible mapping scheme.

3.3.2 Weight Expressions

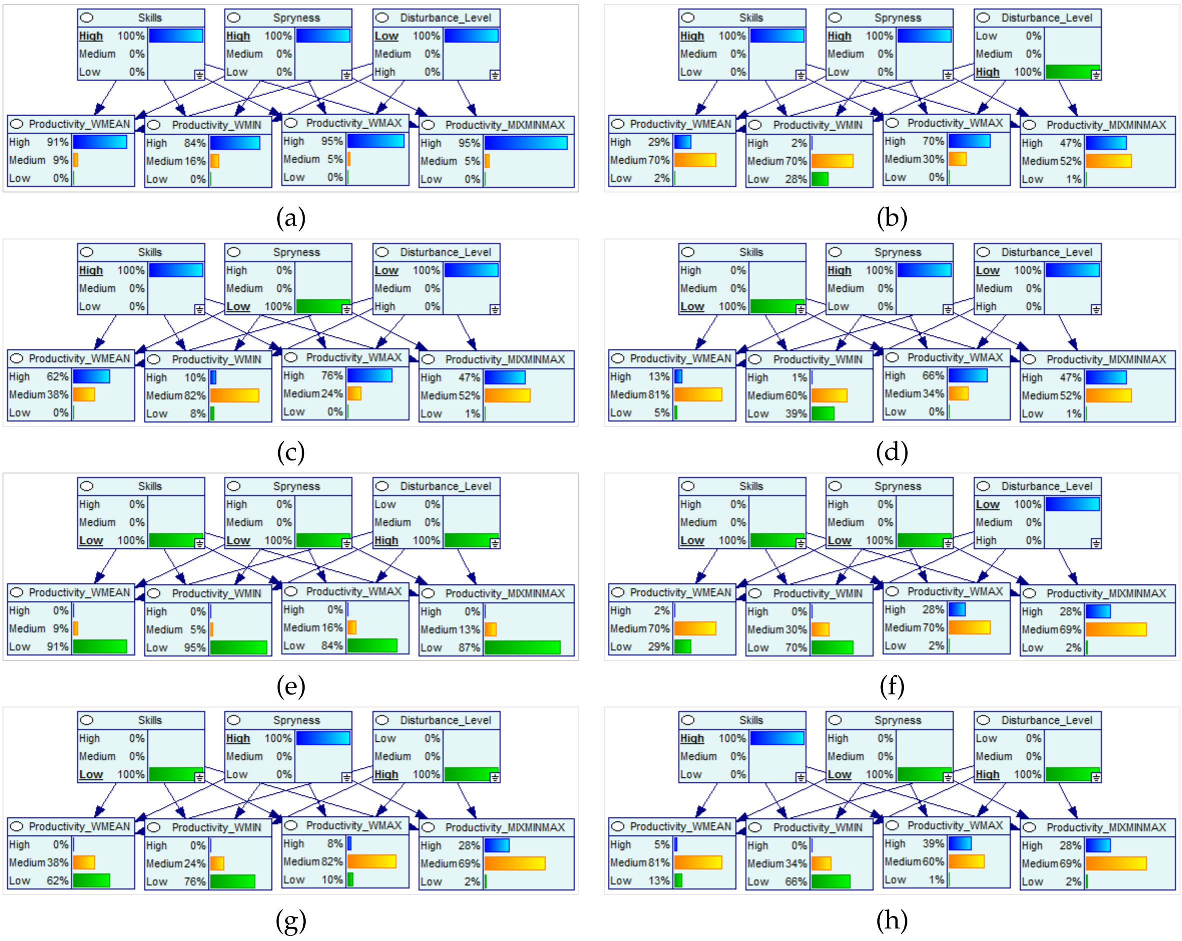
After defining how the states of the ranked nodes are mapped to the normalized scale, one needs to select a weight expression that best describes the probabilistic relationship between the parent nodes and the child node. In practice, the weight expressions are functions that determine the central tendency of the child node for a given combination of states of the parent nodes. The weight expressions aggregate weighted points from the normalized scales of the parent nodes to a point on the normalized scale of the child node. The mathematical forms of the weight expressions are presented in Section 3.3.5 where the calculations carried out in RNM are explained in detail. In this section, the weight expressions are described so that one can understand the differences between them and select the most suitable one without the need to see or understand their mathematical nature. The term *weight expression* is not actually used in [25]. However, the term is found in AgenaRisk software [38] containing the implementation of RNM and is therefore adapted in this thesis.

There are four different weight expressions presented in [25]:

1. Weighted mean (WMEAN)
2. Weighted minimum (WMIN)
3. Weighted maximum (WMAX)
4. Mixture of minimum and maximum (MIXMINMAX)

The characteristics of the weight expressions are described using the example BN and by referring to Figure 3.3. In Figures 3.3a–3.3d, the probability distribution of *Productivity* is shown for the different weight expressions when it is conditioned to an extreme combination of the states of the parent nodes *Skills*, *Spryness*, and *Disturbance Level*. Recall now, that for *Skills*, *Spryness*, and *Productivity*, the order of the states with decreasing preference is (*High*, *Medium*, *Low*), but for *Disturbance Level* it is the opposite.

In WMEAN, the expected value of the child node on [0, 1] corresponds to the weighted mean of the states of the parent nodes on [0, 1]. Comparing the WMEAN scenarios in Figures 3.3b–3.3d to the WMEAN scenario in Figure 3.3a reveals that dropping one parent node from the best state to the worst one shifts the central


Skills High 100% Medium 0% Low 0%
Spryness High 100% Medium 0% Low 0%
Disturbance_Level Low 100% Medium 0% High 0%
Productivity_WMEAN High 91% Medium 9% Low 0%
Productivity_WMIN High 84% Medium 16% Low 0%
Productivity_WMAX High 95% Medium 5% Low 0%
Productivity_MIXMINMAX High 95% Medium 5% Low 0%
Skills High 100% Medium 0% Low 0%
Spryness High 100% Medium 0% Low 0%
Disturbance_Level Low 0% Medium 0% High 100%
Productivity_WMEAN High 29% Medium 70% Low 2%
Productivity_WMIN High 2% Medium 70% Low 28%
Productivity_WMAX High 70% Medium 30% Low 0%
Productivity_MIXMINMAX High 47% Medium 52% Low 1%
Skills High 100% Medium 0% Low 0%
Spryness High 0% Medium 0% Low 100%
Disturbance_Level Low 100% Medium 0% High 0%
Productivity_WMEAN High 62% Medium 38% Low 0%
Productivity_WMIN High 10% Medium 82% Low 8%
Productivity_WMAX High 76% Medium 24% Low 0%
Productivity_MIXMINMAX High 47% Medium 52% Low 1%
Skills High 0% Medium 0% Low 100%
Spryness High 100% Medium 0% Low 0%
Disturbance_Level Low 100% Medium 0% High 0%
Productivity_WMEAN High 13% Medium 81% Low 5%
Productivity_WMIN High 1% Medium 60% Low 39%
Productivity_WMAX High 66% Medium 34% Low 0%
Productivity_MIXMINMAX High 47% Medium 52% Low 1%
Skills High 0% Medium 0% Low 100%
Spryness High 0% Medium 0% Low 100%
Disturbance_Level Low 0% Medium 0% High 100%
Productivity_WMEAN High 0% Medium 9% Low 91%
Productivity_WMIN High 0% Medium 5% Low 95%
Productivity_WMAX High 0% Medium 16% Low 84%
Productivity_MIXMINMAX High 0% Medium 13% Low 87%
Skills High 0% Medium 0% Low 100%
Spryness High 0% Medium 0% Low 100%
Disturbance_Level Low 100% Medium 0% High 0%
Productivity_WMEAN High 2% Medium 70% Low 29%
Productivity_WMIN High 0% Medium 30% Low 70%
Productivity_WMAX High 28% Medium 70% Low 2%
Productivity_MIXMINMAX High 28% Medium 69% Low 2%
Skills High 0% Medium 0% Low 100%
Spryness High 100% Medium 0% Low 0%
Disturbance_Level Low 0% Medium 0% High 100%
Productivity_WMEAN High 0% Medium 38% Low 62%
Productivity_WMIN High 0% Medium 24% Low 76%
Productivity_WMAX High 8% Medium 82% Low 10%
Productivity_MIXMINMAX High 28% Medium 69% Low 2%
Skills High 100% Medium 0% Low 0%
Spryness High 0% Medium 0% Low 100%
Disturbance_Level Low 0% Medium 0% High 100%
Productivity_WMEAN High 5% Medium 81% Low 13%
Productivity_WMIN High 0% Medium 34% Low 66%
Productivity_WMAX High 39% Medium 60% Low 1%
Productivity_MIXMINMAX High 28% Medium 69% Low 2%

(a) (b) (c) (d) (e) (f) (g) (h)

Figure 3.3: Distributions of *Spryness* obtained with the alternative weight expressions.

tendency of *Productivity* from *High* towards the worse states. The shift is largest when *Skills* is in the weakest state and smallest when *Spryness* is in the weakest state. Thus, *Skills* has the strongest effect on *Productivity* and *Spryness* the weakest, respectively. When comparing the WMEAN scenarios in Figures 3.3f–3.3h to that in Figure 3.3e, one observes that as one of the parent nodes changes from the worst state to the best one, also the distribution of *Productivity* shifts towards the best state. The magnitude of the shift caused by a given parent node is the same as in Figures 3.3b–3.3d. This type of behaviour of *Productivity* indicates that its central tendency can be depicted as a weighted average of the states of the parent nodes.

In WMIN, the child node tends to follow that parent node whose state is the lowest on the normalized scale [0, 1]. How strong this tendency is, depends on the relative strengths of influence of the parent nodes on the child node. The comparison of the WMIN scenarios in Figures 3.3b–3.3d and Figure 3.3a exemplifies the nature of WMIN: a change in any parent node from the best state to the worst one strongly shifts the distribution of *Productivity* away from state *High*. As with WMEAN, the shift is largest when the state of *Skills* is changed, and smallest when the state of *Spryness* is changed. Therefore, the parent nodes now have the same order with respect to the strengths of influence as with WMEAN. However, WMIN is more pessimistic than WMEAN — if any of the parent nodes is in the worst state, then that alone makes it highly improbable for *Productivity* to be in the best state. The stronger the influence of the parent node is, the more it can "pull" the central tendency of the child node towards the worst state. On the other side, the parent nodes with a strong influence can better "lift" the central tendency of the child node away from the worst state. This is seen when comparing the WMIN scenarios in Figures 3.3e–3.3h. The degree of the shift of *Productivity* towards the best state varies with the parent node rising from the worst to the best state. The largest shift is achieved when *Skills* rises, and the smallest when *Spryness* rises.

The nature of WMAX is the same as of WMIN but to opposite direction. That is, in WMAX the child node tends to follow the parent node whose state is the highest on the normalized scale. Therefore, in the WMAX scenarios in Figures 3.3a–3.3d, the change of any parent node from the best to the worst state does not shift the distribution of *Productivity* substantially. On the other side, the WMAX scenarios in Figures 3.3e–3.3h point out that the change of any parent node from the worst state to the best one can strongly shift the distribution of the child node towards the best state. Again, the parent nodes have varying strengths of influence on *Productivity*. The order of the strengths in the WMAX scenarios is the same as with WMEAN and WMIN.

In MIXMINMAX, the central tendency of the child node is determined by a weighted mean of the highest and lowest states on the normalized scale found from the combination of states of the parent nodes. Thus, the MIXMINMAX scenarios are all identical in Figures 3.3b–3.3d. The difference to the distribution of *Productivity* is the same independent of which parent node changes its state from the best to the worst. Similarly, the change of any parent node from the worst state to the best one, as displayed in Figures 3.3e–3.3h, always results to the same change in the distribution of the child node. In all of the Figures 3.3b–3.3d and 3.3f–3.3h, the best and worst states found from the parent nodes are the same, i.e., they correspond to the lowest and highest third on the normalized scale [0, 1]. One might wonder, why there is then any difference between the MIXMINMAX scenarios in Figures 3.3b–3.3d and 3.3f–3.3h. This is because in Figures 3.3b–3.3d there is always only one parent node in the worst state whereas in Figures 3.3f–3.3h there are always two. This difference causes the computational routine in MIXMINMAX to produce different results. The routine is thoroughly presented in Section 3.3.5.

In the case study reported in [25], the selection of the weight expression was based on taking combinations of extreme states of the parent nodes, such as *Very High*, *Very Low*, etc., and asking the experts to estimate the most probable response of the child node conditioned on these states. As an aid, a ″truth table″ similar to the one presented in Table 3.1 would be used. Examination of the filled truth table then reveals which weight expression one should select. However, in [25], no exact scheme is presented for how to deduce the most appropriate weight expression from the filled truth table. Presumably in most of the cases it is evident due to the distinct nature of each of the weight expressions. This idea is supported by the note in [25] that once the experts became familiar with the approach, they often identified the appropriate weight expression without the use of the truth table.

Table 3.1: Example of a ″truth″ table for determining the most suitable weight expression in RNM.

Parent Node 1	Parent Node 2	Child Node
Very High	Very High	Very High
Very High	Very Low	Very Low
Very Low	Very High	Low
Very Low	Very Low	High

Suppose that in the example BN, the probability distribution of *Productivity* tends to incline towards *Low* if any of the parent nodes is in the worst state. In addition, suppose that this effect is considered to be stronger when *Skills* is in the worst state compared to either *Spryness* or *Disturbance Level* being in the worst state. In this case, the most suitable weight expression is WMIN.

3.3.3 Weights of Parent Nodes

The third step in RNM is to assign the weights related to the selected weight expression. The weights depict the relative strengths by which the parent nodes affect the central tendency of the child node. If the selected weight expression is WMEAN, WMIN, or WMAX, a weight w_i needs to be assigned to each parent node. If the weight expression MIXMINMAX is used, the central tendency of the child node is determined only by the best and worst state in the state combination of the parent nodes. Hence, only two weights, denoted by w_{MIN} and w_{MAX}, need to be assigned in this case. For all the weight expressions it applies, that the larger the weight is, the stronger is the influence of the corresponding parent node. RNM is implemented in AgenaRisk software [38] developed by the innovators of RNM. In AgenaRisk, the default range of the weights used with all weight expressions is $[1, 5]$.

For the example BN, let the weights of the parent nodes to be used with WMIN be $(w_1, w_2, w_3) = (5, 3, 3)$. This choice of weights reflects the idea that *Skills* is considered to be the most important factor determining *Productivity*, whereas *Spryness* and *Disturbance Level* are thought to be less important factors in an equal manner.

3.3.4 Variance Parameter

Assessing the variance parameter σ^2 is the fourth step in RNM. This parameter represents how much fluctuation one considers to be in the central tendency of the child node for a given combination of states of the parent nodes. The smaller σ^2 is, the less vague the conditional probability distribution of the child node will be. In practice, σ^2 is the variance of a normal distribution that is integrated over the state intervals of the child node to generate the conditional probability distribution. This procedure is explained in detail in the next section. In AgenaRisk software [38], the default scale for σ^2 is $[5 * 10^{-4}, 0.5]$.

Suppose that in the example BN, the conditional probability distribution of the child node is known to concentrate strongly on some specific state for a given combination of states of the parent nodes. To reflect this lack of vagueness in the probability distributions, the value of the variance parameter is set to $\sigma^2 = 0.01$ which is considered small in [25].

3.3.5 Computation of Conditional Probability Tables

Once the first four steps displayed in Figure 3.2 are performed, the CPT of the child node can be generated automatically. This corresponds to the fifth step in Figure 3.2 and is next presented and illustrated using the example BN.

In the fifth step of RNM, the probability distributions in the CPT are generated by repeating the same routine of calculations for each combination of states of the parent nodes. This loop of routines proceeds as follows. Let $(x_1, ..., x_n)$ be a given combination of states of the parent nodes and let $(z_1, ..., z_n)$ denote the state intervals identified with them, respectively. Determining the state intervals $(z_1, ..., z_n)$ corresponding to the given $(x_1, ..., x_n)$ is step 5a in Figure 3.2. For example, the combination of states (x_1, x_2, x_3)=(Low, $High$, Low) of the parent nodes *Skills* (X_1), *Spryness* (X_2), and *Disturbance Level* (X_3) in the example BN is identified with the combination $(z_1, z_2, z_3) = ([0, 1/3), [2/3, 1], [2/3, 1])$ of the state intervals, see Figure 3.1.

After determining the state intervals $(z_1, ..., z_n)$, s equidistant sample points $\{z_{i,j}\}_{j=1}^{s}$ are taken from each of them. This is performed so that the first sample point is the lower bound of the given state interval and the last one is the upper bound. Thus, the sample points are determined deterministically. From the $n * s$ sample points taken, s^n combinations of sample points $\{(z_{1,k}, ..., z_{n,k})\}_{k=1}^{s^n}$ are formed so that in each combination there is one sample point from each parent node. Taking the sample points and forming combinations of them is step 5b in Figure 3.2. For example, using $s = 5$, the sample points taken from z_1 determined above are $\{z_{1,j}\}_{j=1}^{5} = \{0, 0.0833, 0.1667, 0.25, 0.333\}$. For z_2 and z_3, the sample points are $\{z_{2,j}\}_{j=1}^{5} = \{z_{3,j}\}_{j=1}^{5} = \{0.6667, 0.75, 0.8333, 0.9167, 1\}$. From these sample points, $5^3 = 125$ combinations $\{(z_{1,p}, z_{2,r}, z_{3,q})\}_{p,r,q=1}^{5}$ are then formed. These include combinations such as $(z_{1,1}, z_{2,1}, z_{3,1}) = (0, 0.6667, 0.6667)$ and $(z_{1,3}, z_{2,1}, z_{3,5}) = (0.1667, 0.6667, 1)$.

When the combinations of sample points $\{(z_{1,k}, ..., z_{n,k})\}_{k=1}^{s^n}$ are formed, the selected weight expression is used to calculate the corresponding mean values $\{\mu_k\}_{k=1}^{s^n}$ for all of them. This is step 5c in Figure 3.2. The functional forms of the weight expressions are given below with examples presenting their properties.

With WMEAN, μ_k is calculated by

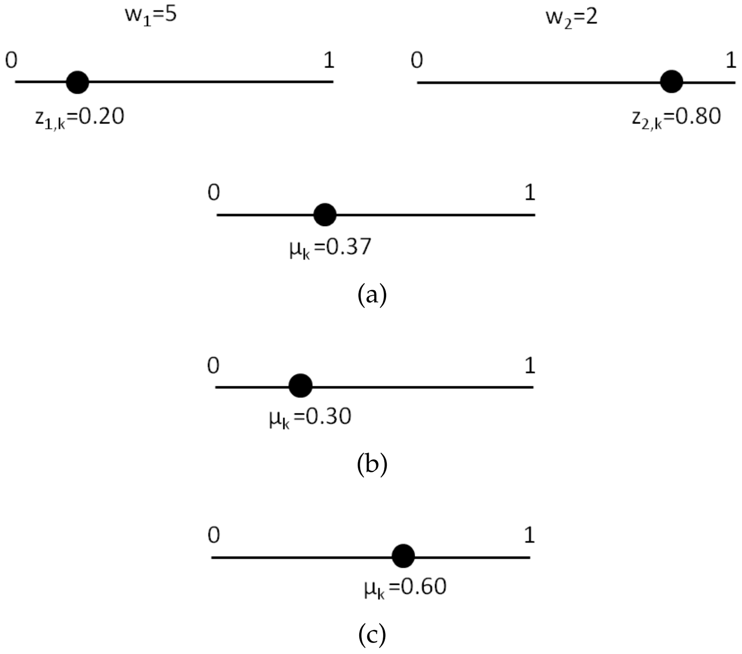
$$\mu_k = WMEAN(z_{1,k}, ..., z_{n,k}, w_1, ..., w_n) = \frac{\sum_{i=1}^{n} w_i * z_{i,k}}{\sum_{i=1}^{n} w_i}, \tag{3.1}$$

where $z_{i,k}$ is the sample point of the ith parent node in the kth combination of the sample points and w_i is the weight of the ith parent node. WMEAN is the weighted average of the sample points. Figure 3.4a illustrates μ_k given by WMEAN, when there are two parent nodes with weights $w_1 = 5$, $w_2 = 2$ and sample points $z_{1,k} = 0.20$, $z_{2,k} = 0.80$. The weighted mean of the sample points, 0.37, is now clearly closer to the sample point $z_{1,k} = 0.20$ coming from the parent node with the larger weight.

When WMIN is used, μ_k is

$$\mu_k = WMIN(z_{1,k}, ..., z_{n,k}, w_1, ..., w_n) = \min_{i=1,...,n} \left\{ \frac{w_i * z_{i,k} + \sum_{j \neq i}^{n} z_{j,k}}{w_i + n - 1} \right\}. \tag{3.2}$$

If all the weights w_i are large, WMIN is close to taking the minimum of the set $\{z_{i,k}\}_{i=1}^{n}$. On the other hand, when all the weights $\{w_i\}$ are equal to 1, then WMIN gives the arithmetic mean of the set $\{z_{i,k}\}_{i=1}^{n}$. With weights of different magnitudes, WMIN results in a value between the minimum and the arithmetic mean. Figure 3.4b displays μ_k obtained when applying WMIN to the sample points $z_{1,k} = 0.20$, $z_{2,k} = 0.80$. The resulting value, 0.30, is smaller than the value calculated using WMEAN, 0.37, implying the tendency of WMIN to let μ_k follow more strongly the small values in the combination of the sample points than WMEAN would.



(a)

(b)

(c)

Figure 3.4: Resulting μ_k when using the sample points of two parent nodes and the weight expression (a) WMEAN, (b) WMIN, and (c) WMAX.

Using WMAX, μ_k is calculated by

$$\mu_k = WMAX(z_{1,k}, ..., z_{n,k}, w_1, ..., w_n) = \max_{i=1,...,n} \left\{ \frac{w_i * z_{i,k} + \sum_{j \neq i}^{n} z_{j,k}}{w_i + n - 1} \right\}. \tag{3.3}$$

Now, μ_k is the maximum of the set from which WMIN selects the minimum. Thus, the value given by WMAX is between the maximum and the arithmetic mean of the set $\{z_{i,k}\}_{i=1}^{n}$. Figure 3.4c presents μ_k resulting from the use of WMAX. The value obtained, 0.60, is larger than the value 0.37 from WMEAN. This depicts how WMAX inclines μ_k towards the large values in the combination of the sample points more strongly than WMEAN.

Figure 3.5 presents μ_k for (a) WMEAN (b) WMIN, and (c) WMAX, when the sample points of the parent nodes in Figure 3.4 are switched. As the parent node with weight 5 has a large point value, 0.8, and the parent node with weight 2 has a small point value, 0.2, the weighted average of them in Figure 3.5a, 0.63, is quite large as well. The value of the outcome of WMIN in Figure 3.5b, 0.40, is rather small again, but not as small as in Figure 3.4b. In WMIN, a parent node with a small weight cannot make the mean of the child node go as low as a parent node with a greater weight. From another perspective, a parent node with a large weight can prevent the mean of the child node from dropping too low.

The behavior of WMAX reveals the same features as WMIN but in the opposite manner. The value in Figure 3.5c, 0.70, resulting from the use of WMAX is larger than the value 0.60 produced by WMAX in Figure 3.4c. Thus, a parent node with a large weight can lift the mean of the child node higher than a parent node with a smaller weight. On the other hand, the small value of a parent node with a large weight can prevent the mean of the child node from rising too high.

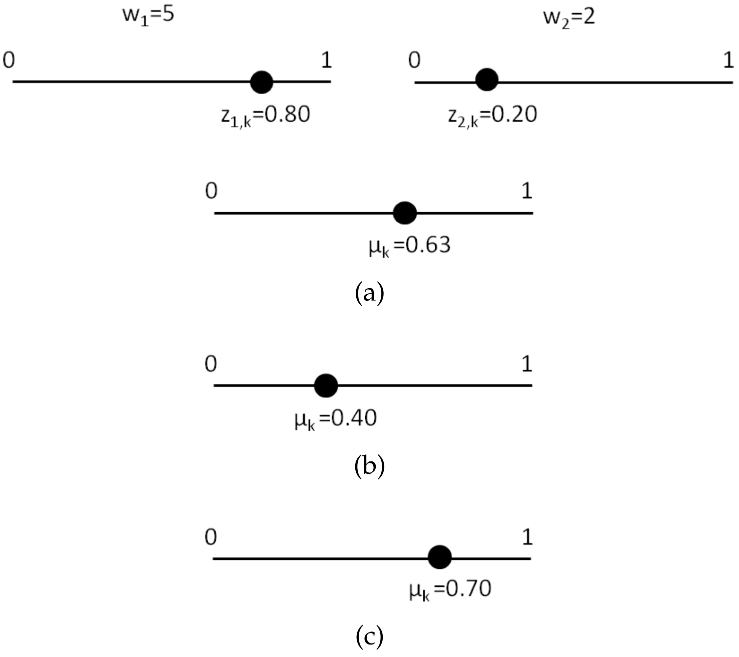


Figure 3.5: Resulting μ_k when the sample points of the parent nodes presented in Figure 3.4 are switched and (a) WMEAN, (b) WMIN, and (c) WMAX is used.

When the selected weight expression is MIXMINMAX, μ_k is determined by

$$\mu_k = MIXMINMAX(z_{1,k}, ..., z_{n,k}, w_{MIN}, w_{MAX}) = \frac{w_{MIN} * \min_{i=1,...,n}\{z_{i,k}\} + w_{MAX} * \max_{i=1,...,n}\{z_{i,k}\}}{w_{MIN} + w_{MAX}}. \tag{3.4}$$

As MIXMINMAX is the weighted average of the minimum and maximum values of the set $\{z_{i,k}\}_{i=1}^{n}$, no weights are assigned to the parent nodes. Thus, all the parent nodes have the same contribution to the child node when using this weight expression. Figure 3.6 illustrates the generation of μ_k based on the combination of sample points of three parent nodes with $w_{MIN} = 5$ and $w_{MAX} = 2$. One obtains the same $\mu_k = 0.37$ as with WMEAN in Figure 3.4. This stems from that the sample point $z_{3,k}$ does not have any effect on μ_k in Equation 3.4.

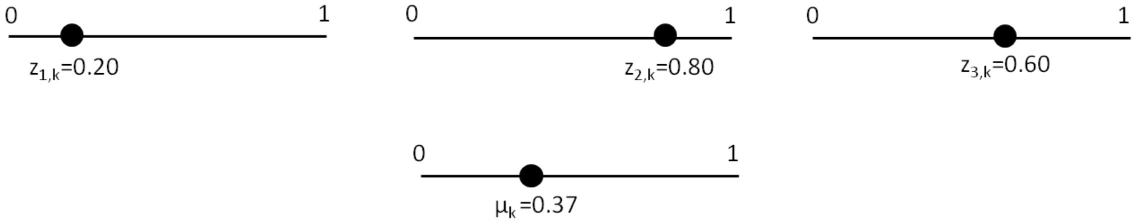


Figure 3.6: Resulting μ_k when MIXMINMAX is used as the weight expression with $w_{MIN} = 5$ and $w_{MAX} = 2$.

The calculation of μ_k is now illustrated with the example BN. The weight expression is WMIN and the weights are $(w_1, w_2, w_3) = (5, 3, 3)$. The state combination under consideration is (x_1, x_2, x_3)=(Low, $High$, Low) which corresponds to the combination $(z_1, z_2, z_3) = ([0, 1/3), [2/3, 1], [2/3, 1])$ of the state intervals. Now, with $s = 5$, the sample points are $\{z_{1,j}\}_{j=1}^{5} = \{0, 0.0833, 0.1667, 0.25, 0.333\}$ and $\{z_{2,j}\}_{j=1}^{5} = \{z_{3,j}\}_{j=1}^{5} = \{0.6667, 0.75, 0.8333, 0.9167, 1\}$. For the combination $(z_{1,1}, z_{2,1}, z_{3,1}) = (0, 0.6667, 0.6667)$ of the sample points, μ_1 is calculated as

$$\mu_1 = WMIN(0, 0.6667, 0.6667, 5, 3, 3) = \\ \min\left\{\frac{5 * 0 + 0.6667 + 0.6667}{5+2}, \frac{3 * 0.6667 + 0 + 0.6667}{3+2}, \frac{3 * 0.6667 + 0 + 0.6667}{3+2}\right\} = \\ \min\{0.1905, 0.5334, 0.5334\} = 0.1905. \tag{3.5}$$

After calculating the mean values $\{\mu_k\}_{k=1}^{s^n}$ associated with all the combinations of the sample points, each μ_k is used as the mean of a normal distribution $N(\mu_k, \sigma^2)$ truncated to the normalized scale $[0, 1]$. That is, the probability density function of this doubly truncated normal distribution $TNormal(\mu_k, \sigma^2, 0, 1)$ is given by

$$TNormpdf(x, \mu_k, \sigma^2, 0, 1) = \frac{Normpdf(x, \mu_k, \sigma^2)}{Normcdf(1, \mu_k, \sigma^2) - Normcdf(0, \mu_k, \sigma^2)}, \tag{3.6}$$

where $Normpdf(x, \mu_k, \sigma^2)$ is the probability density function of $N(\mu_k, \sigma^2)$, i.e.,

$$Normpdf(x, \mu_k, \sigma^2) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu_k)^2}{2\sigma^2}}, \tag{3.7}$$

and $Normcdf(x, \mu_k, \sigma^2)$ is the corresponding cumulative distribution function. For each μ_k, a value $P(X_{n+1} = x_{n+1} | X_1 = x_1, ..., X_n = x_n; \mu_k)$ is determined by the equation

$$P(X_{n+1} = x_{n+1} | X_1 = x_1, ..., X_n = x_n; \mu_k) = \int_{a_{n+1}}^{b_{n+1}} TNormpdf(x, \mu_k, \sigma^2, 0, 1) dx, \tag{3.8}$$

where $[a_{n+1}, b_{n+1}] = z_{n+1}$ is the state interval corresponding to the state x_{n+1} of the child node X_{n+1}. This is step 5d in Figure 3.2. To illustrate this step with the example BN, the value of $P(X_4 = Low | X_1 = Low, X_2 = High, X_3 = Low; \mu_1)$ is calculated. Now, state *Low* of *Productivity* (X_4) corresponds to the state interval $[0, 1/3)$ and thus $TNormal(\mu_1, \sigma^2, 0, 1)$ is integrated over that interval. Technically, the integration is carried out by integrating the normal distribution $N(\mu_1, \sigma^2)$ with mean $\mu_1 = 0.1905$ and variance $\sigma^2 = 0.01$ over $[0, 1/3)$, and dividing the result with the integral of $N(\mu_1, \sigma^2)$ over $[0, 1]$. This yields

$$\begin{aligned} & P(X_4 = Low | X_1 = Low, X_2 = High, X_3 = Low; \mu_1) = \\ & \frac{\int_0^{1/3} Normpdf(x, \mu_1, \sigma^2) dx}{\int_0^1 Normpdf(x, \mu_1, \sigma^2) dx} = \\ & \frac{\int_0^{1/3} Normpdf(x, 0.1905, 0.01) dx}{\int_0^1 Normpdf(x, 0.1905, 0.01) dx} = 0.9212. \end{aligned} \tag{3.9}$$

When $P(X_{n+1} = x_{n+1} | X_1 = x_1, ..., X_n = x_n; \mu_k)$ is calculated for each μ_k, the probability $P(X_{n+1} = x_{n+1} | X_1 = x_1, ..., X_n = x_n)$ in the CPT of X_{n+1} is constructed as their arithmetic mean, i.e.,

$$\begin{aligned} & P(X_{n+1} = x_{n+1} | X_1 = x_1, ..., X_n = x_n) = \\ & \frac{1}{s^n} \sum_{k=1}^{s^n} P(X_{n+1} = x_{n+1} | X_1 = x_1, ..., X_n = x_n; \mu_k). \end{aligned} \tag{3.10}$$

To obtain $P(X_4 = Low | X_1 = Low, X_2 = High, X_3 = Low)$ in the example, the calculation presented in Equation 3.9 would be carried out with each member of $\{\mu_k\}_{k=1}^{125}$. Then, $P(X_4 = Low|X_1 = Low, X_2 = High, X_3 = Low)$ would be obtained as the arithmetic mean of the set $\{P(X_4 = Low|X_1 = Low, X_2 = High, X_3 = Low; \mu_k)\}_{k=1}^{125}$.

The discussion above referring to Figure 3.2 deals with the generation of a single conditional probability $P(X_{n+1} = x_{n+1}|X_1 = x_1, ..., X_n = x_n)$. In order to determine the complete conditional probability distribution of the child node $P(X_{n+1}|X_1 = x_1, ..., X_n = x_n)$, step 5d is repeated for all the state intervals of the child node. This is indicated in Figure 3.2 with the smaller loop. The whole CPT of the child node is obtained by repeating step 5 for all the state combinations of the parent nodes. This is depicted in Figure 3.2 with the larger loop.

3.4 Analogy to Linear Regression

As discussed in Section 3.1, the basic idea in RNM is based on the desire to describe the central tendency of the child node as some kind of a weighted average of the states of the parent nodes. More specifically, the use of scale $[0, 1]$ and doubly truncated normal distributions in RNM is explained in [25] by referring to an analogy with linear regression. Given a data set $\{y_i, x_{i,1}, ..., x_{i,n}\}_{i=1}^{r}$, the linear regression assumes the relationship

$$y_i = \boldsymbol{\beta}^T * (x_{1,i}, ..., x_{n,i}) + \varepsilon_i, \tag{3.11}$$

where $\boldsymbol{\beta}$ is a vector of regression coefficients and ε_i is an error term modeling all factors other than $(x_{1,i}, ..., x_{n,i})$ that influence the dependent observation y_i. Often, ε_i is assumed to follow a normal distribution with zero mean and a constant variance θ^2, i.e, , $\varepsilon_i \sim N(0, \theta^2)$. The larger θ^2 is, the less exactly can the independent variables $(x_{1,i}, ..., x_{n,i})$ explain the value of the dependent variable y_i. In the regression model, the expected value $E(y_i) = \beta^T * (x_{1,i}, ..., x_{n,i})$ is a weighted sum of the independent observations $(x_{1,i}, ..., x_{n,i})$.

In RNM, μ_k, the mean value of the child node in $[0, 1]$ is obtained by substituting the sample points $(z_{1,k}, ..., z_{n,k})$ to the formula of the selected weight expression. Thus, the role of sample points in RNM corresponds to that of the independent variables $(x_{1,i}, ..., x_{n,i})$ in the linear regression model. Similarly, the variance parameter σ^2 is used in RNM as measure of uncertainty, analogically to θ^2 in the linear regression model.

The weight expression WMEAN used in RNM is directly analogical to linear regression. That is, $(w_1, ..., w_n)$, $(z_{i,k}, ..., z_{n,k})$, and σ^2 are used in RNM exactly in the same way as $\boldsymbol{\beta}$, $(x_{1,i}, ..., x_{n,i})$, and θ^2 in linear regression model, respectively. The use of other weight expressions is motivated in [25] by noting that WMEAN is not the only natural function to measure the central tendency of the child node but interactions described by WMIN, WMAX, or MIXMINMAX exist as well. It is stated in [25] that for most applications, the four weight expressions presented are found to be sufficient.

3.5 Extensions to RNM

The use of RNM can be diversified through different means to represent types of probabilistic relationships that do not correspond purely to any of the weight expressions presented in Section 3.3.3. Two ways are described in [25] and they are discussed below.

3.5.1 Non-Monotonic Interaction

Substituting $n = 1$ to any of Equations 3.1–3.4 gives out $\mu_k = z_{i,k}$. Thus, a given sample point $z_{i,k}$ of a parent node always promotes that same value to be the mean of the child node on [0, 1]. Therefore, RNM can describe only monotonic interaction between the parent node and the child node. That is, lifting the state of the parent node on the ordinal scale promotes the central tendency of the child node to move to one direction only. For taking non-monotonic interactions into account, the use of auxiliary variables is suggested in [25]. Let their use be illustrated with an example. Figure 3.7 presents a BN modeling the concentration of a worker and factors affecting it. Each of the nodes has three states which are displayed in Figure 3.7 along with state intervals that seem appropriate to be used in RNM.

Now, suppose that the states *Overload* and *Underload* of *Workload* (W) actually have a similar effect on *Concentration* (C), i.e.,

$$P(C|W = \textit{Underload}) = P(C|W = \textit{Overload}). \tag{3.12}$$

This condition cannot be treated with the state intervals presented in Figure 3.7. To overcome the issue, an auxiliary variable Z is introduced, see Figure 3.8. Z is conditioned to *Workload* according to Table 3.2. That is, the states *Underload* and *Overload* of *Workload* are associated with the state *Abnormal* of Z. Now, the

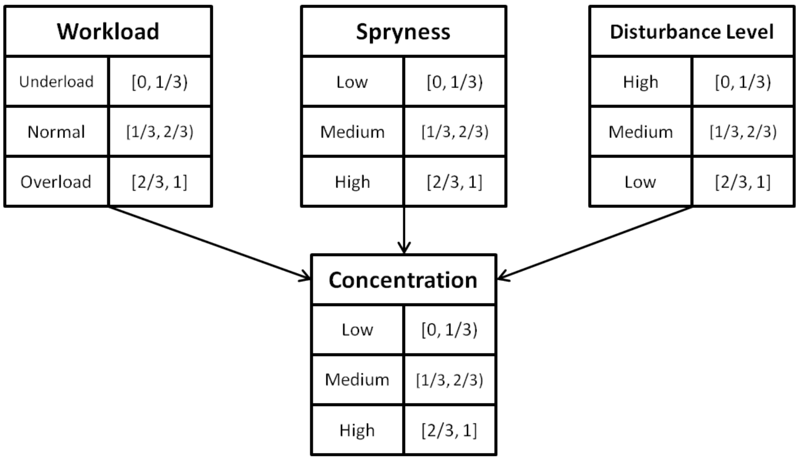


Figure 3.7: Example BN related to the discussion about the non-monotonic influence of a parent node on the child node.

states of Z are identified to state intervals according to Figure 3.8 to properly represent the condition posed by Equation 3.12. In this way, the auxiliary variable Z provides means to express the non-monotonic interaction between *Workload* and *Concentration* in RNM.

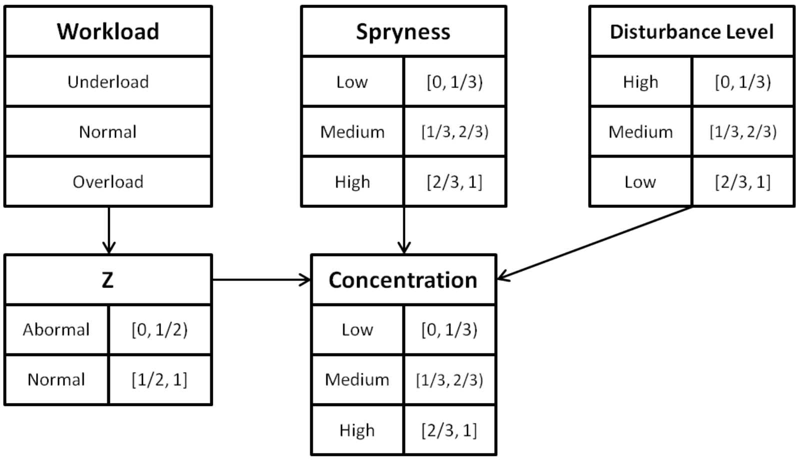


Figure 3.8: Using the auxiliary variable Z to represent the non-monotonic influence of *Workload* on *Concentration*.

3.5.2 Partitioned Expressions

The use of RNM can also be diversified by varying the weight expressions and the related variables in different parts of the CPT. This extension is presented in the documentation of AgenaRisk software [38] in which RNM is implemented.

Table 3.2: CPT of the variable Z in Figure 3.8.

	Workload	Underload	Normal	Overload
Z	Abnormal	1	0	1
	Normal	0	1	0

For example, in the network of Figure 3.8, one could apply WMIN as the weight expression when Z is in the state *Abnormal* and WMEAN when Z is in the state *Normal*. This selection would reflect the effect of the abnormal workload generally making high concentration harder to achieve compared to situations when the workload is normal.

Another way to partition the ranked nodes is to vary the weight expression or the parameters according to the states of various exterior variables that do not need to be ranked nodes. For example, in the network of Figure 3.8, one could give to *Concentration* a new parent node *Employee* with states $\{$*Jack*, *Jill*$\}$ describing which employee, Jack or Jill, is under consideration. For the two states of *Employee*, different sets of parameters of RNM could then be used to generate the CPT of *Concentration*. The other set would reflect the behaviour of Jack while the other would correspond to that of Jill.

In [29], RNM is used with partitioned expressions related to a BN describing a software process. Still, in most of the applications of RNM [24, 23, 28, 56, 15], the extensions of RNM now described have not been needed. In any case, the extensions diversify the use of RNM and by doing so, improve its modeling accuracy.

3.6 Benefits of RNM

There are several good features in RNM that encourage its use. The rough nature of the weight expressions and their use can be explained to a domain expert without a strong mathematical background. One can use verbal descriptions similar to those in Section 3.3.2 and also show illustrative pictures like the ones in Figure 3.3. The role of the weights and the variance parameter can be effectively demonstrated by letting the expert see how the probability distribution of the child node varies when the values of the parameters are altered. This kind of qualitative and visual description of the method helps the expert to understand its general idea which eases up the elicitation of the parameters.

The number of parameters required to assess in RNM is small. To illustrate this, consider a case where there are n parent nodes so that the ith one has m_i states and the child node has m_{n+1} states. In this case, the CPT of the child node consists of $m_{n+1}\prod_{i=1}^{n} m_i$ elements. For example, if $n = 5$ and $m_i = 5,\ \ i = 1, ..., 6$, then there are $5 * 5^5 = 15625$ elements in the CPT of the child node. Assessing each of these directly is unreasonably laborious in practice. As opposed to assessing $m_{n+1}\prod_{i=1}^{n} m_i$ probabilities, using RNM, one can construct the CPT based on $n + 1$ binary decisions concerning the mappings of the states, a single selection concerning the weight expression, and $n + 1$ assessments — or three, if the weight expression is MIXMINMAX — concerning the weights and the variance parameter. Thus, in the example case, the CPT could be constructed based on six binary selections, a single pick out of four options, and six or three numerical assessments. Hence, the total number of any kinds of assessments required would be only 10 or 13 which is drastically less than the number of elements in the CPT. Unlike in most of the existing canonical models, see, e.g., [18] for a review, the amount of elicited parameters in RNM does not grow with the number of states of the parent nodes or the child node. Moreover, in the basic form of RNM, the introduction of a new parent node requires one to assess at most two new parameters — selecting how the states of the parent node are mapped to state intervals and possibly assigning a weight to it. Usually the amount of assessed parameters in canonical models grows only linearly with the number of parent nodes [18]. However, typically more than two new parameters need to be assigned when a new parent node is introduced.

In practice, one may wish to use different weight expressions or change the weights and the variance parameter σ^2 in different parts of the CPT. This increases the number of the parameters in RNM. However, the increase is only linear with the number of partitions of the CPT. Each change of the weights expression requires the assessment of new values for the related weights of the parent nodes and the variance parameter σ^2.

In [25], it is reported that experts see weighting the parent nodes as an intuitive procedure, along with assigning the variance parameter σ^2 as a measure of uncertainty. Also in [13], it is reported that experts find it instinctive to assign relative weights to the parent nodes when quantifying the relative strengths of their influences on the child node. On the other hand, in [49], it is noted that experts dislike the direct estimation of probabilities as they do not think they can provide estimates accurate enough. The relief RNM gives to the workload of the construction process of the CPT can thus be considerable. It requires a small amount of parameters and the experts can also feel more comfortable in providing them instead of assessing probabilities directly.

As the experts can usually provide only a limited amount of time for the elicitation process, factors defining the speed of the process are crucial. In many practical circumstances, it can be the sheer vastness of the elicitation burden that hinders and limits the use of BNs. RNM can be a tremendous aid in resolving issues on this matter. In [11], it is argued that the elicitation of probabilities from experts can be supported to a large extent with an iterative elicitation process based on sensitivity analysis and starting with rough initial probability assessments. The sensitivity analysis provides insight into which probabilities require a high level of accuracy and more careful assessment. Due to the small effort required by the expert while using RNM, it seems as a good method for quickly constructing initial CPTs to be used as the starting point of the sensitivity analysis.

3.7 Issues on RNM

In this chapter, RNM was presented thoroughly enough for implementing it. The discussion of the good features of RNM above notes several benefits that encourage its use as an aid while constructing and quantifying BNs. Despite the apparent usefulness of RNM, there are issues concerning it that have not been addressed earlier. Some of these issues are related to the modeling aspect and others to the computational aspect of RNM.

To begin with, there is a lack of discussion concerning the interpretation of the mapping of the states of the ranked nodes to the state intervals on $[0, 1]$. For example, what do the points on a state interval represent with respect to the associated labelled state? Another theme not addressed earlier concerns the exact interpretation of the weights and how one should try to assess them. Related to both of these themes, there has not been any discussion concerning the use of RNM when the underlying continuous quantities of the ranked nodes are actually defined as ratio or interval scales instead of them being just abstract entities. For example, one might say that the ranked node *Skills* with states $High$, $Medium$, and Low has related to it an underlying continuous quantity that defines the skills of a worker. Yet, there might not be a single physical quantity that depicts the skills exhaustively. As opposed to this, for the ranked node *Salary* with states $High$, $Medium$, and Low, a plausible underlying continuous quantity could be the salary in euro. In some applications [15, 20, 46] of RNM, the states of ranked nodes have been defined like this, i.e., by discretizing ratio or interval scales measuring a physical quantity. However, there is a lack of discussion concerning the properties and implications of using RNM in this way in general.

In addition to the issues related to the modeling aspect of RNM, there are also computational matters that should be investigated. There has not been studies concerning the computational complexity of RNM and the possible limitations set by it to the use of the method. Moreover, the effect of the sample point size as well as the scales of the weights and the variance parameter on the modeling accuracy of RNM has not been studied earlier. The modeling accuracy of RNM is itself something to be explored. This concerns how well CPTs constructed with RNM can actually represent probabilistic relationships associated with different systems and phenomena.

Exploring the above topics is essential to obtain a more profound understanding of the possibilities and limitations related to the use of RNM. This is the main motivation of the study in the thesis at hand. In the next two chapters, RNM is explored from two aspects. Chapter 4 concerns the modeling aspects of RNM and Chapter 5 discusses it from the computational point of view.

Chapter 4

On Modeling Aspect of RNM

This chapter explores RNM from the modeling point of view. Different features of RNM are interpreted and discussed to deepen the existing understanding of its potential and limitations as a method for providing CPTs representing the probabilistic relationships in BNs. Section 4.1 presents and discusses how to interpret the mapping of the states of ranked nodes to state intervals. Section 4.2 provides a novel interpretation for the weights w_i assigned to the parent nodes in the weight expressions. In addition, the utilization of these interpretations in the elicitation of the weights and the conceptual challenges are discussed. Section 4.3 studies through a tentative example how to use RNM when the underlying continuous quantities of the ranked nodes are defined on ratio or interval scales. The chapter ends to Section 4.4 where RNM is compared to other canonical models from the modeling point of view.

4.1 Interpretation of Scale [0, 1]

In Chapter 3, ranked nodes are defined as nodes having discrete ordinal states that can be considered to be abstractions of some kind of underlying continuous quantity. For example, one can think that spryness of a person is a continuous quantity and the states {*High*, *Medium*, *Low*} of the corresponding ranked node *Spryness* are discretizations of the quantity. In RNM, the states of the ranked nodes are identified with intervals of equal width on $[0, 1]$, i.e., the state intervals. Though the mapping may seem intuitive, no exact discussion on its interpretation is provided in [25] that originally presents RNM. The discussion in [29] suggests that identifying the discrete states of ranked nodes with the state intervals is desirable as it reflects the subjectiveness and uncertainty related to the meaning of a given ranked node being in a certain state. However, the meaning of the scale $[0, 1]$ is not addressed there in more detail either. In [26], it is stated that the whole range of possible values of a ranked node is internally defined as the normalized scale $[0, 1]$. Nonetheless, a more detailed discussion of, e.g., the interpretation of the state intervals is lacking there as well. This is set out to do next.

Basically, the use of the normalized scale $[0, 1]$ in RNM stems from the idea to model mathematically the practice of the expert to see the central tendency of the child node forming as some kind of an aggregate of the ordinal states of the parent nodes. If there was no normalized scale, such aggregation could not be performed sensibly. Identifying the discrete states of the ranked nodes with state intervals, rather than with discrete points, on the normalized scale reflects the continuous nature of the underlying quantities of the ranked nodes. In turn, each discrete state of a ranked node can be understood to represent some portion of the underlying continuous quantity. On the other hand, the fact that the discrete states of a ranked node are always identified with state intervals of equal width indicates that each discrete state represents an equal sized portion of the underlying continuous scale. This is visualized in Figure 4.1 for the node *Skills* of the example BN. The interpretation is natural as it implies that if the state of the node is unknown on the underlying continuous scale, then also the discretized states will be under a discrete uniform distribution. This results from integrating the continuous uniform distribution $U(0, 1)$, portraying the ignorance of the state on the underlying continuous scale, over the state intervals on $[0, 1]$.

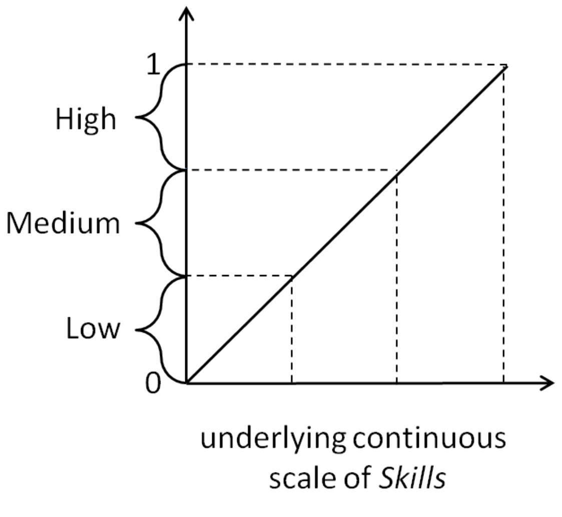


Figure 4.1: Interpreting [0, 1] as the normalized presentation of the underlying continuous scale of *Skills*.

The above interpretation of the normalized scale $[0, 1]$ explains also why the sample points $z_{i,k}$ are taken in RNM. Stating, e.g., that *Skills* is in the state Low means that the skills of the worker are known to be in the lowest third of the skills continuum. However, as the level of skills is not specified more exactly, any point on $[0, 1/3]$ can correspond to the exact state on the underlying continuous scale. Due to this lack of precision related to the fixed state of the ranked node, equidistant sample points are taken from the state interval in RNM. Any combination of sample points in the set $\{z_{1,k}, ..., z_{n,k}\}_{k=1}^{s^n}$ has an equal chance of portraying the exact states of the parent nodes on the normalized scale. Thus, the conditional probability distribution $P(X_{n+1}|X_1 = x_1, ..., X_n = x_n)$ is calculated by taking the average of the distributions $P(X_{n+1}|X_1 = x_1, ..., X_n = x_n; \mu_k)$ obtained with different combinations of sample points, see Equation 3.10.

Formally, the use of the sample points $z_{i,k}$ in RNM corresponds to an approximation of the following hierarchical Bayesian model [27] as means for constructing the CPT of the child node. Let $z_i = [a_i, b_i]$ denote the state interval on $[0, 1]$ related to state x_i of parent node X_i. Moreover, let there be a random variable χ_i identified with parent node X_i so that

$$X_i = x_i \Leftrightarrow \chi_i \sim U(a_i, b_i). \tag{4.1}$$

Now, χ_i represents the exact state of the parent node X_i on $[0, 1]$ when it is known to be in state x_i. As the exact state can be anywhere on the corresponding state interval $z_i = [a_i, b_i]$, a uniform distribution is used for χ_i. Let the mean parameter μ be defined by

$$\mu = f(\chi_1, ..., \chi_n, \boldsymbol{w}), \tag{4.2}$$

where f is the weight expression used, i.e., $f \in$ {WMEAN, WMIN, WMAX, MIXMINMAX}, and $\boldsymbol{w}$ is a vector containing the weights. Because all χ_i are random variables, also μ is a random variable. Let another random variable γ be defined by

$$\gamma = \int_{a_{n+1}}^{b_{n+1}} TNormpdf(u, \mu, \sigma^2, 0, 1)du, \tag{4.3}$$

where $[a_{n+1}, b_{n+1}]$ is the state interval identified with state x_{n+1} of child node X_{n+1} and $TNormpdf(u, \mu, \sigma^2, 0, 1)$ is the probability density function of the doubly truncated normal distribution $TNormal(\mu, \sigma^2, 0, 1)$, see Equations 3.3.5 and 3.7 for clarification. Finally, let the conditional probability $P(X_{n+1} = x_{n+1}|X_1 = x_1, ..., X_n = x_n)$ be calculated as

$$P(X_{n+1} = x_{n+1}|X_1 = x_1, ..., X_n = x_n) = E(\gamma), \tag{4.4}$$

where $E(\gamma)$ is the expected value of γ.

The hierarchical Bayesian model represented by Equations 4.1–4.4 joins the way how the conditional probability distribution of the child node is calculated in RNM, see Equation 3.10, so that

$$\frac{1}{s^n}\sum_{k=1}^{s^n} P(X_{n+1}=x_{n+1}|X_1=x_1,...,X_n=x_n;\mu_k) \xrightarrow{s\to\infty} E(\gamma). \qquad (4.5)$$

That is, as the sample size s used in RNM increases, the value obtained for $P(X_{n+1} = x_{n+1}|X_1 = x_1,...,X_n = x_n)$ converges to the certain value of $E(\gamma)$. The proof of Equation 4.5 is presented in Appendix B.

As the discussion above indicates, the functioning of RNM is rationalized by interpreting the discrete states and the state intervals of the ranked nodes to represent equally sized portions of their underlying continuous quantities. Especially, the discussion above shows the interpretation to be justified when the underlying continuous quantities are abstract in nature, i.e., there is not a scale measuring them exhaustively. This is the case, e.g., for all the nodes in the example BN, see Figure 3.1. In Section 4.3, it is discovered that another type of interpretation of the normalized scale $[0,1]$ is required when applying RNM to ranked nodes whose underlying continuous quantities are measurable on a ratio or interval scale. However, the results presented next are valid independent of the interpretation used.

4.2 Interpretation of Weights

In RNM, the expert needs to assign weights that describe the strengths of influence of the parent nodes on the child node. For the weight expressions WMEAN, WMIN, WMAX, each parent node gets a separate weight. In MIXMINMAX, the weights are associated with to the smallest and largest values in the combination of the sample points $\{z_{1,k},...,z_{n,k}\}$. While it is easy to understand that a larger weight corresponds to a stronger influence, the exact meaning and effect of the weights has not been addressed in the existing literature. In Sections 4.2.1–4.2.4, explicit interpretations of the weights are derived for each of the weight expressions of RNM. The benefits and use of these interpretations in the elicitation of weights is then discussed in Section 4.2.5. Whereas a combination of sample points related to n parent nodes has been marked by $\{z_{i,k}\}_{i=1}^n$ this far, the notation $\{x_i\}_{i=1}^n$ is used in this section for brevity.

4.2.1 WMEAN

Let $x^a = (x_1, ...x_k, ..., x_n)$ and $x^b = (x_1, x_2, ..., , x_k + \Delta x_k, ..., x_n)$ be two combinations of sample points from n parent nodes. Now, using Equation 3.1, one obtains

$$\begin{aligned}
\Delta\mu &:= WMEAN(x^b, w_1, ..., w_n) - WMEAN(x^a, w_1, ..., w_n) \\
&= \frac{\sum_{i=1}^{n} w_i x_i + w_k \Delta x_k}{\sum_{i=1}^{n} w_i} - \frac{\sum_{i=1}^{n} w_i x_i}{\sum_{i=1}^{n} w_i} \\
&= \frac{w_k \Delta x_k}{\sum_{i=1}^{n} w_i} \Leftrightarrow \\
\frac{\Delta\mu}{\Delta x_k} &= \frac{w_k}{\sum_{i=1}^{n} w_i} =: w_k^N .
\end{aligned} \tag{4.6}$$

Thus, when WMEAN is used, the weight of the kth parent node relative to the sum of the weights, w_k^N, equals the ratio between the change in the mean parameter of the child node and the change in the state of the parent node measured on the normalized scale $[0, 1]$. It is enough to have the result only for the relative weights as only they matter in Equation 3.1.

4.2.2 WMIN

Because Equation 3.2 is not linear with respect to the sample points $z_{i,k}$, studying WMIN with $x^a = (x_1, x_2, ..., x_k, ..., x_n)$ and $x^b = (x_1, x_2, ..., x_k + \Delta x_k, ..., x_n)$ does not produce as clear results as for WMEAN above. For WMIN, the difference $\Delta\mu$ depends on the values of the sample points and the weights of the other parent nodes in addition the kth one. However, a clear interpretation for the weights is achieved by considering the sample point combinations $x^a = (c)_{i=1}^{n}$ and $x^b = ((c)_{i=1}^{k-1}, c + \Delta x_k, (c)_{i=k+1}^{n})$ where $c > 0$ and $-c \leq \Delta x_k < 0$.

Substituting x^a to Equation 3.2 gives

$$\mu^a := WMIN(x^a, w_1, ..., w_n) = \min_{i=1,...,n} \left\{ \frac{w_i c + \sum_{j \neq i}^{n} c}{w_i + n - 1} \right\} = c. \tag{4.7}$$

For x^b, the corresponding calculation goes as follows:

$$\begin{aligned}\mu^b &:= WMIN(x^b, w_1, ...w_n)\\ &= \min\left\{\frac{w_k(c+\Delta x_k)+\sum_{i\neq k} c}{w_k+n-1}, \min_{i\neq k}\left\{\frac{w_i c+\sum_{j\neq i} c+\Delta x_k}{w_i+n-1}\right\}\right\}\\ &= \min\left\{\frac{w_k(c+\Delta x_k)+(n-1)c}{w_k+n-1}, \min_{i\neq k}\left\{\frac{w_i c+(n-1)c+\Delta x_k}{w_i+n-1}\right\}\right\}\\ &= \min\left\{\frac{w_k(c+\Delta x_k)+(n-1)c}{w_k+n-1}, \frac{w_s c+(n-1)c+\Delta x_k}{w_s+n-1}\right\}, w_s \leq w_i \;\forall i \neq k.\end{aligned} \tag{4.8}$$

The last line above results from the fact that the function

$$f(x) = \frac{cx+(n-1)c+\Delta x_k}{x+n-1},$$

is strictly increasing when $\Delta x_k < 0$. Let us next explore when the first argument of the minimum operator in Equation 4.8 is smaller than or equal to the second:

$$\begin{aligned}\frac{w_k(c+\Delta x_k)+(n-1)c}{w_k+n-1} &\leq \frac{w_s c+(n-1)c+\Delta x_k}{w_s+n-1} && \Leftrightarrow\\ \frac{(w_k+n-1)c+w_k\Delta x_k}{w_k+n-1} &\leq \frac{(w_s+n-1)c+\Delta x_k}{w_s+n-1} && \Leftrightarrow\\ c+\frac{w_k\Delta x_k}{w_k+n-1} &\leq c+\frac{\Delta x_k}{w_s+n-1} && \Leftrightarrow\\ \frac{w_k}{w_k+n-1} &\geq \frac{1}{w_s+n-1} && \Leftrightarrow\\ w_k(w_s+n-1) &\geq w_k+n-1 && \Leftrightarrow\\ w_k &\geq \frac{n-1}{n-1+\underbrace{w_s-1}_{\geq 0}} \leq 1.\end{aligned}$$

If it is assumed that $w_i \geq 1 \;\forall i = 1, ..., n$ — this assumption is discussed below —, the result above implies the first argument of the minimum operator in Equation 4.8 is never larger than the second one. Thus, the result of Equation 4.8 is

$$\begin{aligned}\mu_b &= \frac{(w_k+n-1)c+w_k\Delta x_k}{w_k+n-1}\\ &= c+\frac{w_k\Delta x_k}{w_k+n-1}.\end{aligned} \tag{4.9}$$

Using the values of μ_a and μ_b in Equations 4.7 and 4.9, one obtains

$$\begin{aligned} \Delta\mu := \mu_b - \mu_a &= c + \frac{w_k \Delta x_k}{w_k + n - 1} - c \\ &= \frac{w_k \Delta x_k}{w_k + n - 1} \Leftrightarrow \\ w_k &= \frac{(n-1)\Delta\mu}{\Delta x_k - \Delta\mu}. \end{aligned} \tag{4.10}$$

Now, Equation 4.10 defines the weight of the kth parent node when using WMIN in terms of Δx_k and $\Delta\mu$. Thus, it is in this sense analogical to Equation 4.6. However, as opposed to 4.6, Equation 4.10 provides the interpretation for the absolute size of weight w_k. This is because in WMIN, it is not enough to know the sizes of the weights relative to each other — their absolute sizes must be known as well.

Above, it is assumed that $w_i \geq 1 \; \forall i = 1, ..., n$. Yet, in [25], the lower bound for the weights is marked to be 0. However, in [25] and [26], it is noted that WMIN and WMAX correspond to taking the arithmetic mean of the states of the parent nodes when all the weights are equal to 1. In both [25] and [26], this remark is presented alongside the note describing the behaviour of WMIN and WMAX when all the weights grow arbitrary large. In this case, WMIN and WMAX correspond to taking the minimum and the maximum of the states of the parent nodes, respectively. This kind of description concerning extreme situations with the weights indicates that 1 is the more appropriate lower bound for the weights than 0. In addition, AgenaRisk software [38] designed by the developers of RNM provides $[1, 5]$ as the default scale for the weights.

4.2.3 WMAX

Being similar to WMIN, the interpretation of the weights for WMAX is derived in the same fashion as above. Let $x^a = (c)_{i=1}^n$ and $x^b = ((c)_{i=1}^{k-1}, c + \Delta x_k, (c)_{i=k+1}^n)$ where $c < 1$ and $0 < \Delta x_k \leq 1 - c$. Substituting x^a to Equation 3.3, one obtains

$$\mu^a := WMAX(x^a, w_1, ..., w_n) = \max_{i=1,...,n} \left\{ \frac{w_i c + \sum_{j \neq i}^n c}{w_i + n - 1} \right\} = c. \tag{4.11}$$

Repeating the same calculation with x^b proceeds as follows:

$$\begin{aligned}\mu^b &:= WMAX(x^b, w_1, ...w_n)\\ &= \max\left\{\frac{w_k(c+\Delta x_k)+\sum_{i\neq k} c}{w_k+n-1}, \max_{i\neq k}\left\{\frac{w_i c+\sum_{j\neq i} c+\Delta x_k}{w_i+n-1}\right\}\right\}\\ &= \max\left\{\frac{w_k(c+\Delta x_k)+(n-1)c}{w_k+n-1}, \max_{i\neq k}\left\{\frac{w_i c+(n-1)c+\Delta x_k}{w_i+n-1}\right\}\right\}\\ &= \max\left\{\frac{w_k(c+\Delta x_k)+(n-1)c}{w_k+n-1}, \frac{w_s c+(n-1)c+\Delta x_k}{w_s+n-1}\right\}, w_s \leq w_i \ \forall i \neq k.\end{aligned} \tag{4.12}$$

Above, the last line results from the fact that the function

$$f(x) = \frac{cx+(n-1)c+\Delta x_k}{x+n-1},$$

is strictly decreasing when $\Delta x_k > 0$. Let us then investigate the condition when the first argument of the maximum operator in Equation 4.12 is larger than or equal to the second:

$$\begin{aligned}\frac{w_k(c+\Delta x_k)+(n-1)c}{w_k+n-1} &\geq \frac{w_s c+(n-1)c+\Delta x_k}{w_s+n-1} && \Leftrightarrow\\ \frac{(w_k+n-1)c+w_k\Delta x_k}{w_k+n-1} &\geq \frac{(w_s+n-1)c+\Delta x_k}{w_s+n-1} && \Leftrightarrow\\ c+\frac{w_k\Delta x_k}{w_k+n-1} &\geq c+\frac{\Delta x_k}{w_s+n-1} && \Leftrightarrow\\ \frac{w_k}{w_k+n-1} &\geq \frac{1}{w_s+n-1} && \Leftrightarrow\\ w_k(w_s+n-1) &\geq w_k+n-1 && \Leftrightarrow\\ w_k &\geq \frac{n-1}{n-1+\underbrace{w_s-1}_{\geq 0}} \leq 1.\end{aligned}$$

Again, if it is assumed that $w_i \geq 1 \ \forall i$ — see the discussion above —, the result above implies that the first argument of the maximum operator in Equation 4.12 is always larger than or equal to the second argument. Thus, it now applies that

$$\begin{aligned}\mu_b &= \frac{(w_k+n-1)c+w_k\Delta x_k}{w_k+n-1}\\ &= c+\frac{w_k\Delta x_k}{w_k+n-1}.\end{aligned} \tag{4.13}$$

Using the values of μ_a and μ_b in Equations 4.11 and 4.13, one now obtains

$$\begin{aligned}\Delta\mu := \mu_b - \mu_a &= c + \frac{w_k \Delta x_k}{w_k + n - 1} - c \\ &= \frac{w_k \Delta x_k}{w_k + n - 1} \Leftrightarrow \\ w_k &= \frac{(n-1)\Delta\mu}{\Delta x_k - \Delta\mu}. \end{aligned} \tag{4.14}$$

Equation 4.13 concerning WMAX turns out to be the same as Equation 4.9 related to WMIN. This describes the symmetry of WMIN and WMAX as weight expressions. It also indicates that similarly to WMIN, the absolute sizes of the weights must be known in WMAX as well.

4.2.4 MIXMINMAX

In MIXMINMAX, the assigned weights w_{MIN} and w_{MAX} are used to calculate the weighted average of the minimum and maximum values of a sample point combination $(x_i)_{i=1}^n$, see Equation 3.4. As with WMEAN, also for MIXMINMAX it is enough to know only the relative weights to apply the weight expression. Reorganizing Equation 3.4 with the substitutions

$$\begin{cases} x_{MIN} = \min_{i=1,\ldots,n}\{x_i\} \\ x_{MAX} = \max_{i=1,\ldots,n}\{x_i\} \\ w_{MIN}^N = \frac{w_{MIN}}{w_{MIN}+w_{MAX}} \\ w_{MAX}^N = \frac{w_{MAX}}{w_{MIN}+w_{MAX}} \end{cases}, \tag{4.15}$$

one obtains

$$\begin{cases} w_{MIN}^N = \frac{\mu - x_{MAX}}{x_{MIN}-x_{MAX}} \\ w_{MAX}^N = 1 - w_{MIN}^N \end{cases}, \tag{4.16}$$

which defines the relative weights related to MIXMINMAX.

4.2.5 Use of Interpretations in Elicitation of Weights

Equations 4.6, 4.10, 4.14, and 4.16 give exact interpretations for the weights related to the weight expressions of RNM. The motivation for the derivation of the equations has been the desire to gain a better understanding of the exact meaning of the weights. In principle, these interpretations can be used to support the elicitation of the weights when using RNM. Instead of assessing the weights directly, their values can be solved from Equations 4.6, 4.10, 4.14, and 4.16 based on assertions about the mode of the child node in different scenarios. Note that while μ is the mean parameter of the doubly truncated normal distribution $TNormal(\mu, \sigma^2, 0, 1)$, it is not necessarily the expected value of the distribution. For example, the expected value of $TNormal(0, \sigma^2, 0, 1)$ is greater than zero even though the mean parameter $\mu = 0$. However, the mode of $TNormal(\mu, \sigma^2, 0, 1)$ is always μ.

For WMEAN and WMAX, the weights of the parent nodes could be elicited by, e.g., asking the expert to specify to which point μ^b does the mode of the child node rise when $x_a = (0)_{i=1}^n$ changes into $x_b = (0, ..., 0, x_k = 1, 0, ..0)$. Now, Equation 4.6 implies that the specified value of μ^b is the relative weight of the kth parent node, w_k^N, when using WMEAN. In the case of WMAX, the weight w_k would be obtained by substituting the answer to Equation 4.14.

When using WMIN, the expert could be asked to which value μ^b the mode of the child node drops from $\mu^a = 1$, when initially all the parent nodes have been in the extreme states $x_a = (1)_{i=1}^n$ and then, the kth parent node has shifted to the extreme state corresponding to $x_k = 0$, i.e., $x_b = (1, ..., 1, x_k = 0, 1, ..1)$. The weight w_k would now be obtained from Equation 4.10. Naturally, the same form of question could be used with WMEAN as well. In this case, Equation 4.6 implies that $w_k^N = 1 - \mu_b$.

For MIXMINMAX, the weights could be elicited by asking the expert to consider a case where $\max\{x_i\}_i^n = 1$ and $\min\{x_i\}_i^n = 0$ and then asking her opinion of the mode μ of the child node on $[0, 1]$. Substituting these values to Equation 4.16 gives

$$\begin{cases} w_{MIN}^N = 1 - \mu \\ w_{MAX}^N = \mu \end{cases} . \tag{4.17}$$

The forms of questions presented above are all only examples of how one can use Equations 4.6, 4.10, 4.14, and 4.16 while eliciting the weights in RNM. The derivation of the equations reveals that it is not necessary to consider only scenarios where the sample points of the parent nodes are always on either of the extreme values 0 and 1. The possibility to modify the forms of the elicitation questions may be used as means to further ease up the elicitation process. For example, the questions can be formed to depict situations that are considered the most easiest to answer.

4.2.6 Conceptual Challenges

As noted above, Equations 4.6, 4.10, 4.14, and 4.16 provide means to elicit the values of the weights indirectly in a consistent way. However, the abstract nature of the ranked nodes may cause some conceptual challenges when using this approach in the elicitation. When answering the elicitation questions concerning the mode of the child node on $[0, 1]$, one might find it hard to specify the necessary value due to the difficulty in conceiving its meaning with respect to the qualitative ordinal scale. For example, consider eliciting the weight of *Skills* in the example BN. Suppose the weight expression is WMIN and the form of question presented in the above section is used. Then, the question to be answered is:

"Think that initially *Skills*, *Spryness*, and *Disturbance Level* are all at the best possible level corresponding to value 1 on the normalized scale [0, 1]. This implies that *Productivity* is most probably at the best possible level corresponding to value 1 on [0, 1]. Consider then that *Skills* drops to the worst possible level corresponding to value 0 while the two other parent nodes remain on the best possible levels. What is now μ_b, the most probable level of *Productivity* on the normalized scale [0, 1]?"

Even though one would understand that the qualitative states {*High*, *Medium*, *Low*} correspond to the subintervals of $[0, 1]$ according to Figure 3.1, it might still feel hard to assess a precise value to μ_b. Even defining some proper interval for the value of μ_b might turn out to be difficult. This is because the challenge now is not about knowing the correct value for μ_b, but rather, understanding the meaning of any value of μ_b with respect to the qualitative ordinal scale. When there is no exact scale representing the underlying continuous quantity of the ranked node, it is impossible to have exact meaning to the points on the normalized scale $[0, 1]$. For example, there is not a proper quantity that could be said to measure the skills or productivity of a worker exhaustively. Thus, the horizontal axes in Figure 4.1 are abstract in nature and the mappings in the graphs have only conceptual meaning.

As the mapping from the underlying continuous quantity to the normalized scale is ill-defined in the exact sense, one might think whether this jeopardizes the whole idea of RNM. When considering this question, it is good to remember that RNM is developed to quantify the type of probabilistic relationships that human experts tend to see between a child node and its parent nodes with qualitative ordinal states. In the lack of clear measures and with states that leave room for subjective interpretation, one cannot be expected to describe the probabilistic dependence between the nodes with great precision. Rather, one might think that in such contexts, depicting the probabilistic relationships between the nodes in a coarser manner is best that can be achieved and it is acceptable to use a workable heuristic. The use of RNM in various applications indicates that it provides satisfactory means for this kind of coarse modeling. Thus, though there is conceptual difficulty related to the ill-defined mappings from the discrete states of the ranked nodes to the state intervals, it can be regarded to be a heuristic feature of RNM rather than a fatal flaw.

Discussing still the conceptual challenges related to the elicitation of weights, it should be remembered that, as opposed to assessing only a single weight, RNM always requires specification of two or more weights that reflect the strength of influence of the parent nodes to the child node relative to each other. As the weights are elicited by sequentially considering the behaviour of the child node on the normalized scale, it could be expected to help to set the weights of the parent nodes more coherently than direct estimation. This is a topic that should be investigated empirically to be able to say some concrete conclusions about the matter. However, planning and executing such research is beyond the scope of this thesis. Instead, the discussion next continues with another topic related to use of the interpretations of the weights. This concerns how to apply RNM to nodes whose underlying continuous quantity is not an abstract concept but a variable naturally measured on a ratio or an interval scale.

4.3 Variables with Interval or Ratio Scales as Ranked Nodes: Illustrative Example

As discussed in Section 3.1, the emergence of RNM is related to cases where the underlying continuous quantities of the ranked nodes are abstract in nature [25]. This is also the situation in the first applications of RNM [23, 24]. The use of discrete ordinal scales such as $\{Low,\ Medium,\ High\}$ is a natural way to describe states of continuous quantities that are abstract, like spryness or disturbance level. But similarly, one can also discretize continuous quantities measured readily on interval or ratio scales into intervals identified with descriptive ordinal states. For example, the scale $\{Very\ Small,\ Small,\ Medium,\ Large,\ Very\ Large\}$ can be used to describe the size of an apartment so that the states correspond to consecutive intervals of surface area measured in square metres. In some applications [15, 20, 46] of RNM, the states of ranked nodes have been defined like this. However, there has not been any discussion on the properties and implications concerning the use of RNM in this way. The matter is next addressed through an illustrative example.

The BN in Figure 4.2 describes how the monthly rent, depicted by *Rent*, of an apartment is dependent on three other features. *Surface Area* represents the surface area of the apartment. *Distance to Centre* depicts the distance from the apartment to the centre of the city. *Time from Last Overhaul* portrays the time that has passed since the last overhaul of the apartment, or its construction, if there has not been any overhauls.

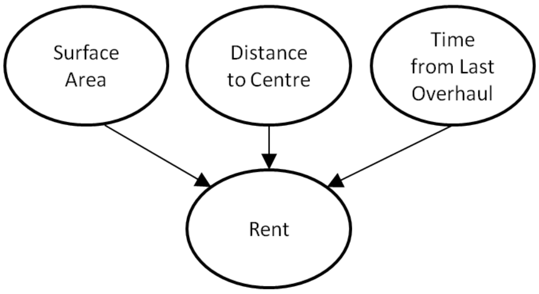


Figure 4.2: Example BN related to the discussion on the application of RNM to nodes with an interval scale.

Each of the nodes in the BN in Figure 4.2 can be considered to have a continuous interval or ratio scale. The scale types, the units, as well as the lower and upper limits of the scales are presented in Table 4.1. Suppose that the rent is known to increase for an increasing surface area and decrease when the distance from the centre or the time from the last overhaul are increasing. The monotonic influence of the parent nodes to *Rent* is consistent with the functioning of RNM. Moreover, the nodes could be considered as ranked nodes by appropriately discretizing the ratio and interval scale. Thus, in principle, RNM could be used to construct a CPT representing the probabilistic relationship between the nodes. How to do this in practice is discussed next.

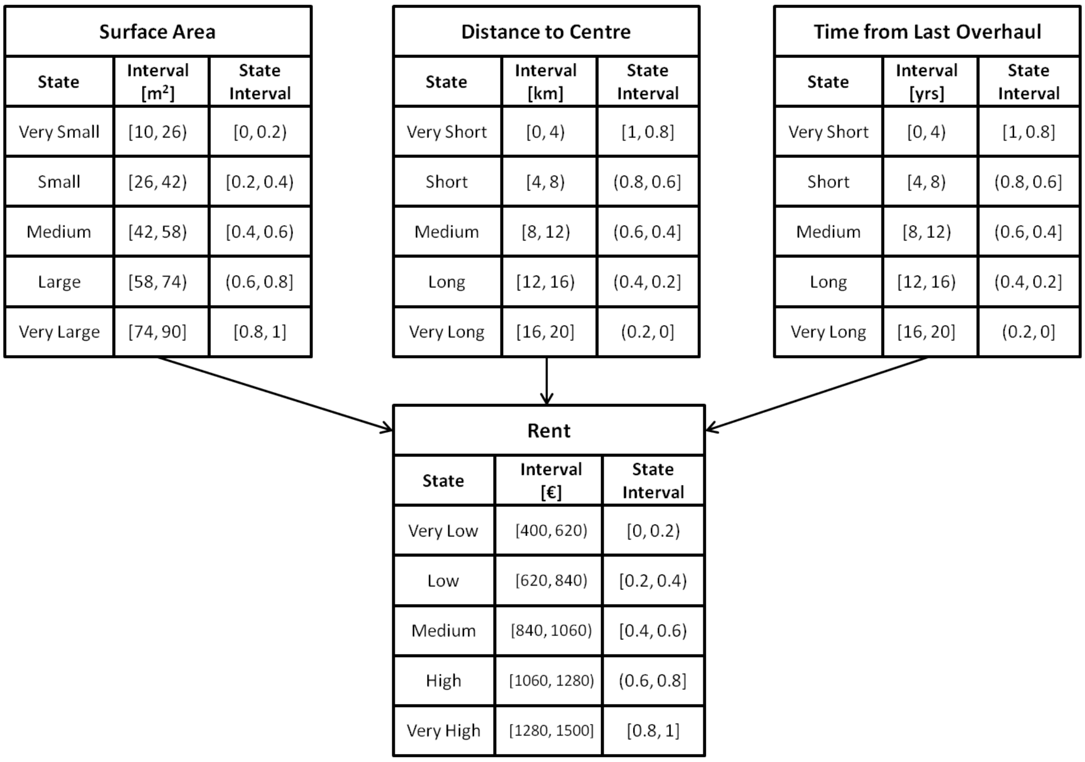
4.3.1 Discretization of Interval and Ratio Scales

In order to apply RNM to the example BN presented above, the discrete states of the nodes need to be defined and identified with state intervals on $[0, 1]$. In Section 4.1, it is discussed that the state intervals of the ranked nodes correspond to equal sized portions of their underlying continuous quantities. Thus, the discrete

Table 4.1: Units and the limits of the interval and ratio scales of the nodes in Figure 4.2.

	Surface Area	Distance to Centre	Time from Last Overhaul	Rent
Scale Type	Interval	Ratio	Ratio	Interval
Unit	m^2	km	yrs	€
Upper Limit	90	20	20	1500
Lower Limit	10	0	0	400

states of the ranked nodes should now be readily achieved by dividing the ratio and interval scales into desired numbers of intervals of equal width. For example, if a 5 point scale, such as $\{Very\ Low, Low, Medium, High, Very\ High\}$, is used for all the nodes in Figure 4.2, the resulting ranked nodes are those displayed in Figure 4.3.



Surface Area

State	Interval [m^2]	State Interval
Very Small	[10, 26)	[0, 0.2)
Small	[26, 42)	[0.2, 0.4)
Medium	[42, 58)	[0.4, 0.6)
Large	[58, 74)	(0.6, 0.8]
Very Large	[74, 90]	[0.8, 1]

Distance to Centre

State	Interval [km]	State Interval
Very Short	[0, 4)	[1, 0.8]
Short	[4, 8)	(0.8, 0.6]
Medium	[8, 12)	(0.6, 0.4]
Long	[12, 16)	(0.4, 0.2]
Very Long	[16, 20]	(0.2, 0]

Time from Last Overhaul

State	Interval [yrs]	State Interval
Very Short	[0, 4)	[1, 0.8]
Short	[4, 8)	(0.8, 0.6]
Medium	[8, 12)	(0.6, 0.4]
Long	[12, 16)	(0.4, 0.2]
Very Long	[16, 20]	(0.2, 0]

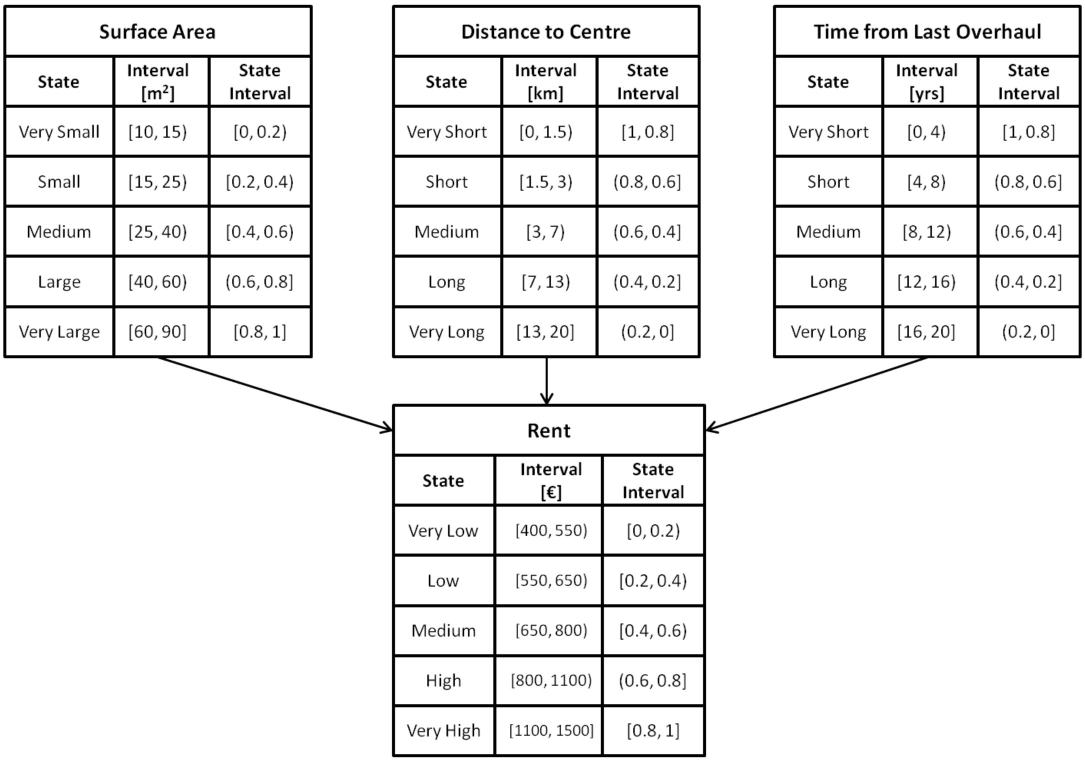
Rent

State	Interval [€]	State Interval
Very Low	[400, 620)	[0, 0.2)
Low	[620, 840)	[0.2, 0.4)
Medium	[840, 1060)	[0.4, 0.6)
High	[1060, 1280)	(0.6, 0.8]
Very High	[1280, 1500]	[0.8, 1]

Figure 4.3: Identifying subintervals of equal width on the interval and ratio scales with different ordinal states and state intervals.

The directions of the state intervals presented in Figure 4.3 are such that they correctly depict the influence of the parent nodes to *Rent*. For example, each of the states $Very\ Large$ of *Surface Area*, $Very\ Short$ of *Distance to Centre*, and $Very$

$Short$ of *Time from Last Overhaul* promotes the state $Very\ High$ of *Rent*. This is because they are all identified to the state interval $[0.8, 1]$ and a given sample point from the normalized scale of a parent node always promotes the same value to be the mode of the child node on $[0, 1]$, see Section 3.5.1. However, when reviewing which subintervals of the ratio and interval scales correspond to which labelled states, one might feel there to be inaccuracy. For example, as opposed to the discretization of the underlying ratio scale of *Distance to Centre* displayed in Figure 4.3, one might claim that a distance of 4 km from the centre cannot be described as "very short". Thus, based on the labelled states of the nodes, one might readily discretize the underlying ratio and interval scales of the ranked nodes into subintervals of different widths. In the case of the ongoing example, the resulting discretizations could then be, e.g., those presented in Figure 4.4. In [20] and [46], RNM is applied so that the discrete states of some ranked nodes correspond to different sized portions on the underlying ratio scales. However, the rationales behind the discretizations are not discussed in detail.



Surface Area

State	Interval [m^2]	State Interval
Very Small	[10, 15)	[0, 0.2)
Small	[15, 25)	[0.2, 0.4)
Medium	[25, 40)	[0.4, 0.6)
Large	[40, 60)	(0.6, 0.8]
Very Large	[60, 90]	[0.8, 1]

Distance to Centre

State	Interval [km]	State Interval
Very Short	[0, 1.5)	[1, 0.8]
Short	[1.5, 3)	(0.8, 0.6]
Medium	[3, 7)	(0.6, 0.4]
Long	[7, 13)	(0.4, 0.2]
Very Long	[13, 20]	(0.2, 0]

Time from Last Overhaul

State	Interval [yrs]	State Interval
Very Short	[0, 4)	[1, 0.8]
Short	[4, 8)	(0.8, 0.6]
Medium	[8, 12)	(0.6, 0.4]
Long	[12, 16)	(0.4, 0.2]
Very Long	[16, 20]	(0.2, 0]

Rent

State	Interval [€]	State Interval
Very Low	[400, 550)	[0, 0.2)
Low	[550, 650)	[0.2, 0.4)
Medium	[650, 800)	[0.4, 0.6)
High	[800, 1100)	(0.6, 0.8]
Very High	[1100, 1500]	[0.8, 1]

Figure 4.4: Identifying subintervals of different widths on the interval and ratio scales with different ordinal states and state intervals.

Independent of how the discretizations of the underlying scales are performed, it should be checked that they are compatible with the functioning of RNM in the following sense. Referring to Figure 4.4, consider an apartment that has a surface area between 60 and 90 m^2, is located between 0 to 1.5 km from the centre, and has 0 to 4 years from the last overhaul. Then, this apartment is expected to have a rent somewhere between 1100 and 1500 € because all these intervals correspond to the state interval $[0.8, 1]$. This exemplifies that when sample points of the parent nodes are taken from the same state interval in RNM, also the mode of the child node lies on that interval for all the combinations of the sample points. The feature is independent of the weight expression or the weights which can be verified by exploring Equations 3.1–3.4. If this type of verification of the discretizations is found to reveal incorrect implications concerning the probabilistic relationships of the nodes, the discretizations should be revised.

The verification of discretizations is straightforward to carry out when all the nodes have the same amount of discrete states. That is, the parent nodes are repeatedly fixed to be in states corresponding to a given state interval. For each combination, it is checked whether the corresponding subintervals of the parent nodes on their underlying scales really promote the child node to lie on the corresponding subinterval on its underlying scale. If the implications are deemed to be incorrect, the discretizations are adjusted. If the nodes have varying number of discrete states, the same routine cannot be used and the verification of the discretizations is harder to perform. Hence, it is favourable to use the same amount of discrete states for each node. If a suitable discretization for all the nodes seems impossible to find, it indicates that the RNM does not provide adequate means to construct the CPT representing the probabilistic relationship between the parent nodes and the child node.

4.3.2 Revised Interpretation of Scale [0, 1]

The discussion in Section 4.3.1 indicates that contrary to the interpretations made in Section 4.1, the state intervals in RNM do not necessarily represent equal portions of the underlying continuous quantities of the ranked nodes. Thus, the interpretation of the normalized scale $[0, 1]$ requires revision for the case where the continuous quantities are represented by ratio and interval scales.

In Section 4.3.1, the discretizations of the ratio or interval scales of the ranked nodes are recommended to be adjusted so that they are compatible with the functioning of RNM. This corresponds to having the discrete states of the parent nodes associated to given values on $[0, 1]$ promoting discrete states of the child node associated to the same values. Performing the discretizations in this way corresponds to making elementary statements about the probabilistic relationship between the parent nodes and the child node. These statements set the framework for the more detailed description using a weight expression, weights, and a variance parameter. Thus, the normalized scale $[0, 1]$ can be generally interpreted as an interface that provides the basis for any description of the probabilistic relationship between the parent nodes and the child node. Related to this, the state intervals of the ranked nodes can be viewed as representing distinguishable portions of their underlying scales through which their probabilistic relationship with the other nodes can be described.

4.3.3 Elicitation of Weights

After verifying the discretizations of the interval or ratio scales of the ranked nodes, the use of RNM can be continued in the normal way, as described in Chapter 3. Thus, one would first select the weight expression considered the most appropriate and then assign the weights to the parent nodes. In accordance with the description of RNM in Chapter 3, the weights can be assigned directly and then be refined if the verification of the resulting CPT indicates need for it. As opposed to this, the elicitation could also be based on the interpretations of the weights derived in Section 4.2. Recall that in Section 4.2.5, it is introduced how the elicitation of weights in RNM can be supported by the use of Equations 4.6, 4.10, 4.14, and 4.16. In 4.2.6, the abstract nature of the underlying continuous quantities of the ranked nodes is suspected to introduce some cognitive challenge to the elicitation of the weights based on the interpretations. This matter is now revised for the case where the ranked nodes have interval or ratio scales by considering the BN in Figure 4.4.

In Figure 4.4, consecutive subintervals on the interval scale of a ranked node correspond to consecutive state intervals on $[0, 1]$. Based on this, one can think that there is a continuous monotonic mapping between the scales. A piecewise linear mapping is an example of such a mapping. Figure 4.5 displays piecewise linear mappings between the underlying and the normalized scales of *Surface Area* and *Rent* that are in accordance to the discretizations presented in Figure 4.4.

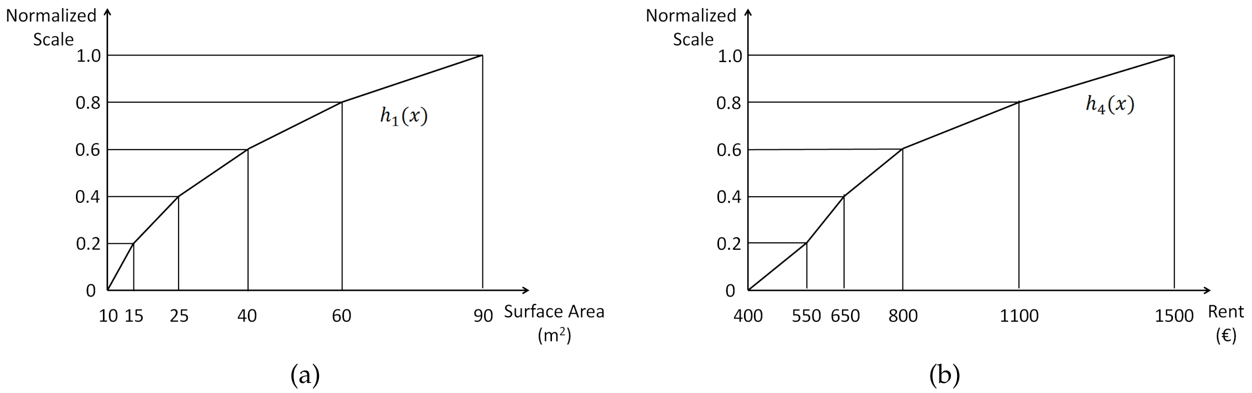


Figure 4.5: Piecewise linear mapping of the underlying scale of (a) *Surface Area* and (b) *Rent* into $[0, 1]$ in accordance to the discretizations displayed in Figure 4.4.

The functional form of the mapping for *Surface Area* in Figure 4.5a is

$$h_1(x) = \begin{cases} \frac{x-10}{25}, & 10 \leq x < 15 \\ 0.2 + \frac{x-15}{50}, & 15 \leq x < 25 \\ 0.4 + \frac{x-25}{75}, & 25 \leq x < 40 \\ 0.6 + \frac{x-40}{100}, & 40 \leq x < 60 \\ 0.8 + \frac{x-60}{150}, & 60 \leq x \leq 90 \end{cases}, \tag{4.18}$$

where x is the surface area in m^2 and $h_1(x)$ is the corresponding value on [0, 1]. Correspondingly, the functional form of the mapping for *Rent* in Figure 4.5b is

$$h_4(x) = \begin{cases} \frac{x-400}{750}, & 400 \leq x < 550 \\ 0.2 + \frac{x-550}{500}, & 550 \leq x < 650 \\ 0.4 + \frac{x-650}{750}, & 650 \leq x < 800 \\ 0.6 + \frac{x-800}{1500}, & 800 \leq x < 1100 \\ 0.8 + \frac{x-1100}{2000}, & 1100 \leq x \leq 1500 \end{cases}, \tag{4.19}$$

where x is the rent in euros and $h_4(x)$ is the corresponding value on $[0, 1]$.

Suppose piecewise linear mappings similar to those presented in Figure 4.5 are defined for all the nodes in Figure 4.4. Then, these mappings link together points on the underlying scales of the nodes in the same way as common state intervals of the nodes are discussed to link together subintervals on their underlying scales in Section 4.3.1. This means that the elicitation of the weights by means introduced in Section 4.2.5 can be carried out by relating the elicitation questions to the interval and ratio scales of the nodes. For example, suppose that the dependence of *Rent* from its parent nodes is characterized by either the weight expression WMEAN or WMIN. In this case, the weight of *Surface Area* can be asked with the following question:

Q1a: *"Think initially of an apartment with the surface area 90 m^2 that is right in the centre of the city and is brand new. The rent of this kind of apartment is most probably 1500 €. Consider then another apartment that is as new and located in the same place but has only 10 m^2 of surface area. What is the rent of this kind of apartment most probably?"*

The apartment that is initially described in Q1a corresponds to each of the parent nodes being at point $c = 1$ on the normalized scales, see Figure 4.4. Thus, in RNM, the initial mode of *Rent* on the normalized scale is $\mu_a = c = 1$ that corresponds to $h_4^{-1}(1) = 1500$ € based on Equation 4.19. The second apartment corresponds to change $\Delta x_1 = -1$ on the normalized scale of *Surface Area*. Suppose that the answer obtained to Q1a is 500 €. This corresponds to the value $\mu_b = h_4(500) = 0.133$ on the normalized scale of *Rent*. Thus, the change of mode on the normalized scale of *Rent* is $\Delta\mu = \mu_b - \mu_a = -0.867$.

If the weight expression WMEAN is used, the relative weight of *Surface Area*, w_1^N, according to Equation 4.6 is

$$w_1^N = \frac{\Delta\mu}{\Delta x_1} = \frac{-0.876}{-1} = 0.876. \tag{4.20}$$

As discussed in Section 4.2.1, only the relative weights of the parent nodes have effect in WMEAN. Thus, the value of w_1^N obtained contains the required information of the effect of *Surface Area* to *Rent* when WMEAN is used.

When WMIN is the weight expression, the weight of *Surface Area*, w_1, is given by Equation 4.10 with the substitution $n = 3$:

$$w_1 = \frac{(n-1)\Delta\mu}{\Delta x_1 - \Delta\mu} = \frac{2 * -0.867}{-1-(-0.867)} = 13.0. \tag{4.21}$$

In the case of WMAX, w_1 could be elicited with the question

Q2: *"Think initially of an apartment with the surface area 10 m^2 that is 20 km from the centre of the city and has not had an overhaul for 20 years. The rent of this kind of apartment is most probably 400 €. Consider then another apartment that is located in the same place and has the same time from the last overhaul but has 90 m^2 of surface area. What is the rent of this kind of apartment most probably?"*

Now, the apartment that is initially described in Q2 corresponds to each of the parent nodes being at point $c = 0$ on the normalized scales, see Figure 4.4. This yields the initial mode of *Rent* on the normalized scale to be $\mu_a = c = 0$ that corresponds to $h_4^{-1}(0) = 400$ €. The second apartment corresponds to change $\Delta x_1 = 1$ on the normalized scale of *Surface Area*. Suppose that the answer obtained to Q2 is 1200 €. This corresponds to the value $\mu_b = h_4(1200) = 0.85$ on the normalized scale of *Rent*. Thus, the change of the mode on the normalized scale of *Rent* is $\Delta\mu = \mu_b - \mu_a = 0.85$ and the value of w_1 given by Equation 4.14 with the substitution $n = 3$ is

$$w_1 = \frac{(n-1)\Delta\mu}{\Delta x_1 - \Delta\mu} = \frac{2 * 0.85}{1 - 0.85} = 11.3. \tag{4.22}$$

If MIXMINMAX is the weight expression, w_1 can be elicited by asking the following question:

Q3:*"Consider an apartment with the surface area 90 m^2 and is right in the centre of the city but has 20 years since its last overhaul. What is the rent of the apartment most probably?"*

The apartment described in Q3 corresponds to *Surface Area* and *Distance from the Centre* to have the value $x_{MAX} = 1$ on the normalized scales and *Time from Last Overhaul* to have the value $x_{MIN} = 0$, see Figure 4.4. Suppose the answer received to Q2a is 1000 €. Then, the value of w_{MIN}^N obtained from Equation 4.16 is

$$w_{MIN}^N = \frac{\mu - x_{MAX}}{x_{MIN} - x_{MAX}} = \frac{h_4^{-1}(1000) - 1}{0 - 1} = \frac{0.7333 - 1}{-1} = 0.27, \tag{4.23}$$

and the corresponding value of w_{MAX}^N is $1 - w_{MIN}^N$=0.73.

4.3.4 Inconsistency in Elicitation

In the end of Section 4.2.5, it is discussed that a given weight can be elicited with different forms of questions. This is seen as a facilitation to the elicitation process as one can modify the questions such that they feel easy to answer. The same feature can also be used as means to reveal inconsistencies in the RNM representation by asking the same weight with different elicitation questions. For example, instead of eliciting the weight of *Surface Area* with Q1a, one could just as well use the following question:

Q1b:*"Think initially of an apartment with the surface area 40 m^2 that is 3 km from the centre and is 8 years old. The rent of this kind of apartment is most probably 800 €. Consider then another apartment that is as old and located in the same place but has only 30 m^2 of surface area. What is the rent of this kind of apartment most probably?"*

In the question Q1b, the apartment that is first described corresponds to each of the parent nodes being at point $c = 0.6$ on the normalized scales, see Figure 4.4. In RNM, this implies that the initial mode of *Rent* on the normalized scale is $\mu_a = c = 0.6$ that corresponds to $h_4^{-1}(0.6) = 800$ €. The surface area of the second apartment type corresponds to the point $x_1^b = h_1(30) = 0.467$ on the normalized scale. Suppose now that the answer obtained to the question Q1b is 720 €. Then, substituting the values

$$\begin{cases} n & = 3 \\ \Delta x_1 & = x_1^b - c = 0.467 - 0.6 = -0.133 \\ \Delta \mu & = h_4(720) - c = 0.4933 - 0.6 = -0.1067 \end{cases}, \tag{4.24}$$

to Equation 4.10 gives

$$\tilde{w}_1 = \frac{(n-1)\Delta\mu}{\Delta x_1 - \Delta\mu} = \frac{2 * -0.1067}{-0.133 - (-0.1067)} = 8.0, \tag{4.25}$$

which differs from the value $w_1 = 13.0$ obtained in Equation 4.21. Hence, the answers to Q1a and Q1b seem to be inconsistent. The origin and ways to deal with such inconsistency is next discussed.

Modeling abilities of any canonical model are limited by its underlying assumptions. Therefore, they cannot be expected to flawlessly represent the probabilistic relationships in a given real-life phenomenon. The inconsistency between w_1 and $\tilde{w}_1$ above is an example of such imperfection in RNM. The origin of the emerging inconsistency cannot be said to be a specific feature of RNM but the functioning of the method as a whole. Accordingly, there are various ways to resolve the inconsistency. The simplest way is that the answer to either of the questions Q1a or Q1b is refined. The answer to Q1b that is consistent with the answer 500 € to Q1a and the related weight $w_1 = 13.0$ can be calculated, using Equation 4.10, as follows:

$$\begin{aligned}
w_1 &= \frac{(n-1)\Delta\mu}{\Delta x_1 - \Delta\mu} \Leftrightarrow \\
13.0 &= \frac{2(h_2(x) - 0.6)}{-0.133 - (h_2(x) - 0.6)} \Leftrightarrow \\
h_4(x) &= \frac{13.0(-0.133 + 0.6) + 0.6 * 2}{2 + 13.0} \\
&= 0.4844 \Leftrightarrow \\
x &= h_4^{-1}(0.4844) \\
&= 713.33. \qquad (4.26)
\end{aligned}$$

Thus, if it is deemed acceptable to change the answer to Q1b from 720 € to 713.33 €, the inconsistency in the value of w_1 is resolved.

Another way to deal with the inconsistency between w_1 and $\tilde{w}_1$ is to ask the answers to questions Q1a and Q1b as intervals and take the value of w_1 from the possible intersection of the corresponding intervals obtained for w_1. For example, suppose the answers to Q1a and Q1b are the intervals $[480 \text{ €}, 520 \text{ €}]$ and $[710 \text{ €}, 730 \text{ €}]$, respectively. The corresponding intervals for w_1 are calculated by similar means as the initial point estimates in Equations 4.21 and 4.25 and the results are $w_1 \in [10.5, 16.8]$ for Q1a and $w_1 \in [4.7, 18.0]$ for Q1b. Thus, as a compromise, one could select the value of w_1 to be, e.g., $w_1 = 14.0$ that is found from the intersection of the intervals for w_1.

If the inconsistency between w_1 and $\tilde{w}_1$ cannot be resolved by either of the two means presented above — one is not willing to change the answers to Q1a and Q1b or the intervals obtained for w_1 do not intersect —, partitioned expressions, see Section 3.5.2, could be used as an aid. Now, the CPT of *Rent* could be generated by using weight $w_1 = 13.0$ when *Surface Area* is in state $Very\ Small$ and weight $\tilde{w}_1 = 8.0$ when the node is in state $Medium$. This would then reflect the idea that the strength of influence of *Surface Area* to *Rent* is not constant but varies depending on the states of the former.

In principle, the inconsistency between w_1 and $\tilde{w}_1$ can also be handled by refining the discretizations of the nodes *Surface Area* and *Rent*. For example, consider altering the discretization of the interval scale of *Surface Area* displayed in Figure 4.4 so that the states $Small$ and $Medium$ correspond to subintervals $[15 \text{ m}^2, 23.5 \text{ m}^2]$ and $[23.5 \text{ m}^2, 40 \text{ m}^2]$, respectively. In this case, the piecewise linear mapping $h_1(x)$

defined by Equation 4.18 changes into $\tilde{h}_1(x)$ defined by

$$\tilde{h}_1(x) = \begin{cases} \frac{x-10}{25}, & 10 \leq x < 15 \\ 0.2 + \frac{x-15}{42.5}, & 15 \leq x < 23.5 \\ 0.4 + \frac{x-23.5}{82.5}, & 23.5 \leq x < 40 \\ 0.6 + \frac{x-40}{100}, & 40 \leq x < 60 \\ 0.8 + \frac{x-60}{150}, & 60 \leq x \leq 90 \end{cases}, \tag{4.27}$$

Now, the value of Δx_1 in Equation 4.25 becomes $\Delta x_1 = \tilde{h}_1(30) - 0.6 = 0.4769 - 0.6 = -0.1231$ which yields

$$\tilde{w}_1 = \frac{(n-1)\Delta\mu}{\Delta x_1 - \Delta\mu} = \frac{2 * -0.1067}{-0.1231 - (-0.1067)} = 13.0, \tag{4.28}$$

and resolves the inconsistency concerning the weight of *Surface Area*.

The inconsistency between w_1 and $\tilde{w}_1$ may also be dealt with by redefining the mappings $h_1(x)$ and $h_4(x)$. As discussed above, piecewise linear functions are just an example of the kind of continuous monotonic mapping that can be considered to exist between the interval or ratio scales and the normalized scales of the nodes. To make the RNM representation more consistent with the answers of the elicitation question regarding the weights, the mappings can be modified freely. For example, $h_1(x)$ can be modified so that the value of $x_1^b = h_1(30)$ in Equation 4.24 yields the result $\tilde{w}_1 = w_1 = 13.0$ from Equation 4.25. The desired value for $h_1(30)$ can be calculated from Equation 4.10 by

$$\begin{aligned} \tilde{w}_1 &= \frac{(n-1)\Delta\mu}{\Delta x_1 - \Delta\mu} \\ &= \frac{(n-1)\Delta\mu}{h_1(30) - h_1(40) - \Delta\mu} \Leftrightarrow \\ 13.0 &= \frac{2 * -0.1067}{h_1(30) - 0.6 - (-0.1067)} \Leftrightarrow \\ h_1(30) &= 0.4769. \end{aligned} \tag{4.29}$$

Contrary to the result above, the definition for $h_1(x)$ in Equation 4.18 yields $h_1(30) = 0.467$. The result of Equation 4.29 can be taken into account in $h_1(x)$ by splitting the linear part between $x \in [25, 40]$ into two linear parts between $x \in [25, 30]$ and $x \in [30, 40]$ so that $h_1(30) = 0.4769$. This is illustrated in Figure 4.6.

Though there are number of ways to remove inconsistencies arising in the elicitation of the weights, one cannot expect to able to remove all of them. Handling

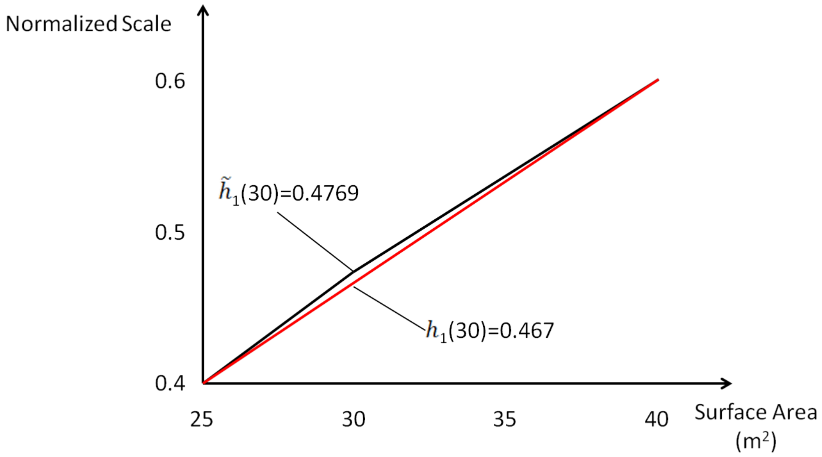


Figure 4.6: Function $h_1(x)$ is linear when $x \in [25, 40]$ and function $\tilde{h}_1(x)$ is linear when $x \in [25, 30]$ and $x \in [30, 40]$.

successfully one of them might introduce a new one leading to an endless cycle of corrections. The emergence of the inconsistencies can be regarded as an embodiment of the lack of accuracy concerning the use of RNM to construct a CPT representing the probabilistic relationships in the phenomenon at hand. Then, the crucial question is whether a satisfying accuracy can be reached and whether it is possible with an acceptable amount of effort. As discussed in Section 3.6, RNM is regarded as a method to quickly construct initial CPTs that are used the starting point of an iterative elicitation process based on, e.g., sensitivity analysis. In the sensitivity analysis of a BN, one or more conditional probabilities are varied and the effect on a probability of interest is observed. This indicates which conditional probabilities are the most important to get correct. These values can then be refined separately from the other elements in the CPTs. Hence, one should not get tangled to some specific inconsistencies regarding the use of RNM. Removing them might have a minimal effect on the overall quality of the BN or they might represent flaws that are better corrected by other means than revised use of RNM.

4.4 Comparison to Other Canonical Models

In RNM, the central tendency of a child node is determined as a kind of weighted aggregate of the states of parent nodes by using the normalized scale $[0, 1]$. The selection of the weight expression and assessment of the weights and the variance parameter basically corresponds to describing the nature of the simultaneous effect of the parent nodes on the child node. This way of RNM for considering the causal influence of the parent nodes differs from most of the canonical models. Usually, e.g., in noisy-OR [45] and noisy-MAX [17, 52], the basic idea is that parent nodes are causes which can affect the child node independent of each other. This idea is known as independence of causal influence (ICI) in the literature [31, 30, 18]. In ICI models, each parent node X_i is assumed to have a so-called distinguished state x_i^* in which it does not affect the child node. In principle, if all the parent nodes $X_1, ..., X_n$ are in the distinguished states, then the child node X_{n+1} will be in its distinguished state x_{n+1}^* for sure. If the state of a single parent node X_i changes from the distinguished state x_i^*, it can disturb the child node away from the distinguished state x_{n+1}^*. The parameters to be assessed in ICI models are usually the conditional probabilities of the states of the child node given that all parent nodes except one are in the distinguished states, i.e.,

$$P(X_{n+1} = x_{n+1} | X_j = x_j^*, X_i = x_i^* \forall i \neq j), i = 1, ..., n. \tag{4.30}$$

Thus, contrary to RNM, one does not need to consider the combined influence of parent nodes acting simultaneously but the effect of one parent node at a time.

The different ways for considering the nature of the probabilistic relationship between the nodes in the ICI models and RNM introduces a considerable difference to the amount of parameters that need to be assessed in order to construct CPTs. Suppose there are n parent nodes with the ith one having m_i states and a child node having m_{n+1} states. In this case, there are $m_{n+1} \prod_{i=1}^{n} m_i$ elements in the CPT of the child node. As discussed in Section 3.6, the construction of the CPT of the child node with RNM requires making

$$N_{RNM}^{sel} = n + 2, \tag{4.31}$$

selections concerning the mapping of the states of the nodes to the state intervals and the weight expression. In addition, one has to assign

$$N_{RNM}^{ass} = \begin{cases} n+1 & \text{if } f \in \{\text{WMEAN, WMIN, WMAX}\} \\ 3 & \text{if } f = \text{MIXMINMAX} \end{cases}, \tag{4.32}$$

parameters specifying the weights related to the weight expression f as well as the value of the variance parameter. In turn, the use of noisy-MAX requires one to make

$$N_{noisy-MAX}^{sel} = n + 1, \tag{4.33}$$

Table 4.2: Distinguished states of the nodes in the example BN for the use of noisy-MAX.

Node	Skills	Spryness	Disturbance Level	Productivity
Distinguished state	Low	Low	High	Low

selections concerning the distinguished states of the nodes and to assign

$$N_{noisy-MAX}^{ass} = m_{n+1} \sum_{i=1}^{n} m_i, \tag{4.34}$$

probabilities. Equations 4.31 and 4.33 indicate that the amount of decisions in RNM and noisy-MAX differs only by one in favour of the latter. However, the comparison of Equations 4.32 and 4.34 indicates that RNM requires significantly smaller amount of numerical assessments than noisy-MAX.

To illustrate the difference in the amount of parameters between noisy-MAX and RNM, consider the construction of the CPT of *Productivity* in the example BN displayed in Figure 2.1. This CPT has $3^4 = 81$ elements, see Table A-4 in Appendix A. Using noisy-MAX, one has to specify now $N_{noisy-MAX}^{sel} = 4$ distinguished states that could be those presented in Table 4.2. That is, if all the parent nodes are in the worst states, also *Productivity* is in the worst state for sure. In addition, one is required to make $N_{noisy-MAX}^{RNM} = 18$ probability assessments corresponding to the entries of Table 4.3. The rows in Table 4.3 correspond to situations where two parent nodes are in the distinguished states but one of them is not. That is, one has to consider how a change in the state of a single parent node affects *Productivity* when the other parent nodes are not promoting any change. The remaining probability distributions of the full CPT of *Productivity* are deterministically generated based on the entries of Table 4.3, see, e.g., [17, 52] for details. As presented in Chapter 3, the construction of the CPT of *Productivity* with RNM requires $N_{RNM}^{sel} = 5$ selections concerning the mappings of the states of the nodes and the weight expression. In addition to these, $N_{RNM}^{ass} = 3$ or $N_{RNM}^{ass} = 2$ numerical assessments concerning the weights are needed.

The small amount of parameters to be elicited makes RNM seem appealing compared to noisy-MAX or other canonical models. Moreover, avoiding the assessment of probabilities may be considered desirable, as discussed in Section 3.6. However, the selection of the canonical model cannot be based solely on the amount of parameters to be elicited. Stress must be put on the ability of the

Table 4.3: Conditional probability distributions of *Productivity* that need to be elicited from the expert in noisy-MAX.

Parents			Productivity		
Skills	Spryness	Disturbance level	High	Medium	Low
High	Low	High			
Medium					
Low	Medium				
	High				
	Low	Medium			
		Low			

model to represent the probabilistic relationship between the nodes. There is no single canonical model that would always be the best choice because the models have been designed by assuming different kinds of probabilistic relationships. When constructing a CPT, one should try to discover which one of the models best portrays the probabilistic relationship of the nodes at hand. For example, if the behaviour of *Productivity* can be described as some kind of weighted average of the states of its parent nodes, RNM could be handy in constructing the CPT. If, on the other hand, the interaction is perceived better using the idea of, e.g., noisy-MAX, one should use that instead. In [25], it is stated that RNM is complementary to other elicitation methods. The same idea applies to canonical models in general. Different canonical models should be seen as tools supplementing each other and one should always seek to use the one that best corresponds to the given context.

Chapter 5

On Computational Aspect of RNM

This Chapter explores RNM through two experimental studies related to the computational properties of the method. The studies provide additional knowledge concerning the practical use of RNM and complement the discussion of its modeling aspect presented in Chapter 4. In Section 5.1, the computational complexity of RNM is studied by measuring the calculation times of CPTs of various sizes with two implementations of RNM. The calculations are carried out with Intel Core i5 CPU 2.40 GHz processor, 4 GB RAM, and Windows Vista — a set-up which is considered to be a standard desktop computer. In Section 5.2, the modeling accuracy of RNM and the effect of the sample point size s are studied by measuring how well RNM can approximate CPTs found in various BNs used in real-life applications.

5.1 Experimental Study on Computational Complexity of RNM

The generation of a CPT by RNM contains several computational operations. If there are n parent nodes and the ith one has m_i states, then there are in total $\prod_{i=1}^{n} m_i$ combinations of states of the parent nodes. With the sample point size s, any combination of states $(x_1, ..., x_n)$ requires constructing s^n combinations of sample points. For each combination of the sample points $(z_{1,k}, ..., z_{n,k})$, the value of μ_k is calculated and the doubly truncated normal distribution $TNormal(\mu_k, \sigma^2, 0, 1)$ is integrated over the state intervals of the child node. If the child node has m_{n+1} states, the definite integral of $TNormal(\mu_k, \sigma^2, 0, 1)$ is calculated over $m_{n+1} - 1$ state intervals. For example, consider a case where $n = 5$, $m_1, ..., m_5 = 5$, $s = 5$, and $m_{n+1} = 5$. Then, while generating the CPT, $\prod_{i=1}^{5} 5 * 5^5 = 9765625$ combinations of sample points are constructed and the value of μ_k is calculated as many times. Moreover, the definite integral of a doubly truncated normal distribution is calculated $(5 - 1) * 9765625 = 39062500$ times.

The numbers above may seem large but does this cause any practical problems when the CPT is generated by a computer? When testing the above example in AgenaRisk [38] software implementing RNM, the program warns the user of a possibly slow calculation time and, indeed, the generation of the CPT lasts several minutes. As RNM is meant to be a tool in probability elicitation, one is interested in whether the CPT can be generated reasonably fast. To gain insight into the computational complexity of RNM from this practical point of view, an experimental study is performed. The study concerns generating CPTs with RNM for different sets of controlled variables and measuring calculation times. Section 5.1.1 presents the experimental design of the study. Section 5.1.2 presents the results. Summary of the experiment is given in Section 5.1.3.

5.1.1 Experimental Design

To measure the calculation times of CPTs by RNM, the method is implemented with MATLAB [39]. By carrying out the investigation with a self programmed implementation, one can modify and explore the computational properties of RNM with greater invisibility and detail than is possible with AgenaRisk. MATLAB is used as it has several built-in functions that ease up the programming task and help in investigating the properties of the implementation. In addition, MATLAB is considered to be a good example of a software that can be used to develop a custom-built program that utilizes RNM. Thus, implementing and investigating RNM with MATLAB provides information that can be used in planning and executing such a custom-built program.

In the experiment, CPTs for a child node are generated with RNM using different sets of controlled variables and calculation times are recorded. The controlled variables that are varied in the experiment are the number of parent nodes n, the numbers of states of the nodes m_i, the sample point size s, and the weight expression f. To keep the number of combinations of the controlled variables manageable, the number of states of the parent nodes and the child node is always kept mutually equal, i.e., $m_i = m \; \forall i = 1, ..., n + 1$. The ranges for the controlled variables in the experiment are the following:

- $m \in \{2, ..., 7\}$
- $s \in \{1, ..., 5\}$
- $n \in \{2, ..., n_{ub}\}$
- $f \in$ {WMEAN, WMIN, WMAX, MIXMINMAX}

For m, the lower bound $m_{lb} = 2$ is the minimum amount of states that a random variable can have. The upper bound $m_{ub} = 7$ is selected because in current practice the ordinal scales typical to RNM — such as $\{Very\ Low$, Low, $Medium$, $High$, $Very\ High\}$ — usually consist of five or seven categories [48]. The lower bound for s is set to $s_{lb} = 1$ which is the minimum number of sample points that can be taken in practice. When using $s = 1$, the middle points of the state intervals of the parent nodes are used as sample points. For larger values of s, the end points of the state interval are always included in the sample points, as discussed in Section 3.3.5. The upper bound of s is set to be $s_{ub} = 5$ which is the default sample size in AgenaRisk and therefore a good reference point. Larger sample sizes are ruled out of the experiment because the interest is more on studying how small the elapsed times can be as opposed to how large they can become. As the idea in RNM is to define the central tendency of the child node by "weighing" the states of the parent nodes, the case of a single parent node, $n = 1$, is not deemed to be interesting. Therefore, the lower bound for n is set to be $n_{lb} = 2$. The upper bound n_{ub} is not set beforehand but measurements are conducted for different $n \geq 2$ until a suitable limit is reached.

Two implementations of RNM are used in the experiment. In the first implementation, referred to as A, the CPT of the child node is generated in two separate steps. In the first step, all possible combinations of sample points $(z_{1,k}, ..., z_{n,k})_{k=1}^{s^n}$ for each combination of states of the parent nodes $(x_1, ..., x_n)$ are constructed and stored. In the second step, the CPT is generated by using the stored combinations of sample points. The idea of dividing the generation of the CPT into two separate steps stems from practical considerations. As all required combinations of sample points are constructed and stored beforehand, they do not need to be constructed repeatedly during an elicitation session where the CPT of the child node may need to be refined and regenerated several times.

In the second implementation, referred to as B, the combinations of sample points $(z_{1,k}, ..., z_{n,k})_{k=1}^{s^n}$ are not generated beforehand for each combination of states of the parent nodes. Instead, the CPT is generated column by column so that only the combinations of sample points $(z_{1,k}, ..., z_{n,k})_{k=1}^{s^n}$ related to a single combination of states of the parent nodes $(x_1, ..., x_n)$ are in the memory storage at a time. Thus, as opposed to A, the B reconstructs the combinations of sample points each time the CPT is regenerated in an elicitation session. This increases the calculation times of the CPTs. However, the second implementation requires less memory which might turn out to be a desirable feature.

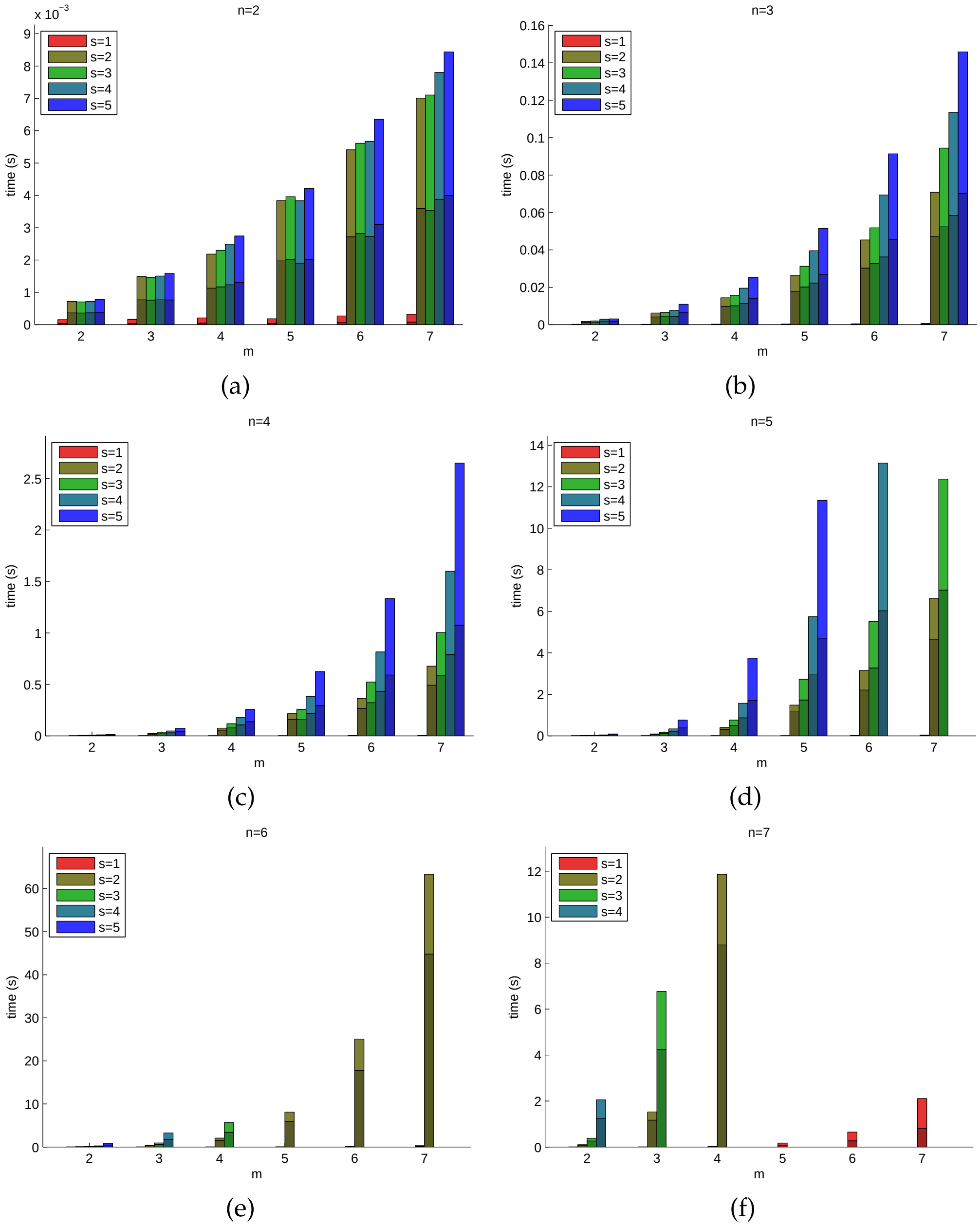
To increase the reliability of the results, the experiment is carried out by making 50 repetition trials for each set of the controlled variables. In each repetition, the weights $(w_1, ..., w_n)$ or (w_{MIN}, w_{MAX}) and the variance parameter σ^2 are let vary arbitrary in the default ranges defined in AgenaRisk. By not fixing these parameters to be the same in each repetition indicates better the possible range of calculation times of the implementations for a fixed set of the controlled variables than using the same weights and variance parameter in each repetition.

The computer set-up used in the experiment is considered to be a standard desktop computer. This set-up is used to get an idea of what kinds of calculation times can be expected in a real-life elicitation session. In order to be able to check the effectiveness of the implementations of RNM used in the experiment, some results obtained with them are compared to manually measured calculation times obtained with AgenaRisk.

5.1.2 Results

For both implementations A and B, the calculation times for a fixed set of the controlled variables do not vary dramatically in the repetitions — the standard deviations of the repetitions are typically about 10 percent or less of the average values. This degree of variation is considered small enough to focus the analysis of the results on the average values of the repetitions. Moreover, the results obtained with different weight expressions are found to be similar with each other for both implementations. This is illustrated in Figures C-1 and C-2 in Appendix C for A and B, respectively. Figures C-1 and C-2 display the average calculation times of CPTs for different sets of the controlled variables. It can be observed that the average calculation times obtained with different weights expressions are similar when the other controlled variables are fixed. This similarity of the results with different weight expressions indicates that the duration of calculating μ_k is of the same magnitude for all of them. Because of the results obtained with different weight expressions are so similar, further analysis is performed only for the results obtained with one of them, i.e., WMEAN.

Figure 5.1 displays the average calculation times obtained with implementation



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.1: Average calculation times of a CPT with implementation A for varying number of parent nodes n, number of states of the nodes m, and number of sample points s using WMEAN as the weight expression.

A for different values of n, m, and s when using WMEAN as the weight expression. In Figure 5.1, each bar consists of two superimposed segments that correspond to the two steps in implementation A. The lower segments represent the proportion of the calculation time spent on constructing and storing all possible combinations of sample points $(z_{1,k}, ..., z_{n,k})_{k=1}^{s^n}$ for each combination of states of the parent nodes $(x_1, ..., x_n)$. Correspondingly, the upper segments represent the proportion taken by the generation of the CPT using the stored combinations of sample points. Some of the charts in Figure 5.1 are lacking bars related to certain values of s or m. These omissions correspond to cases where the computer ran out of memory while constructing the combinations of sample points.

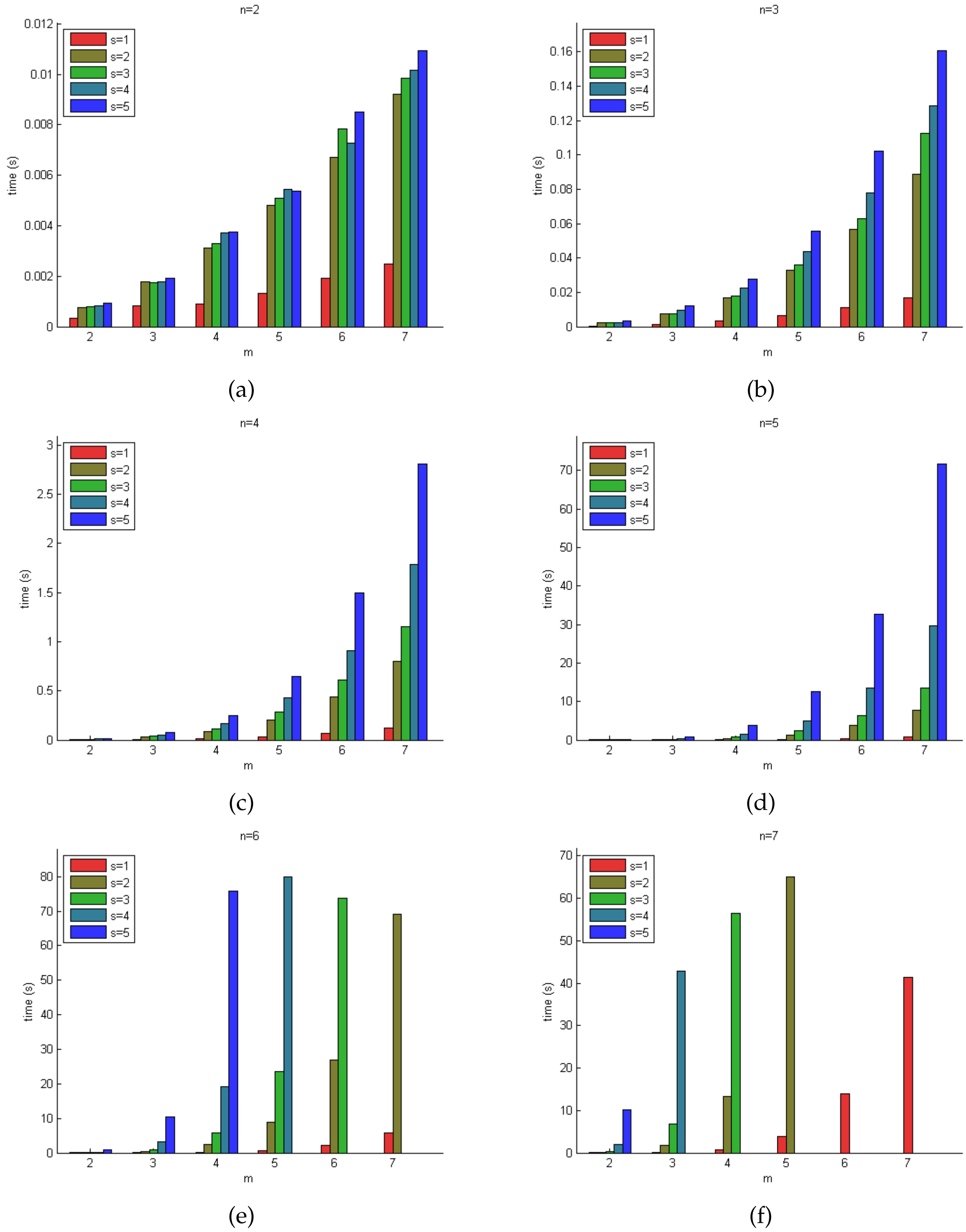
It can be observed in Figure 5.1 that the calculation times tend to increase for increasing n, m, and s. This is expected as an increase in any of these variables increases the size of the computing task related to producing the CPT. Results contrary to the general trend can be seen only in Figure 5.1a. This anomaly is considered to be the result of the fact that once the elapsed time are only thousandths of a second, some computational mechanisms of MATLAB consume time enough to obscure the differences between the different measurements. It can also be noticed that increasing n with one unit increases the elapsed time considerably for any fixed values of m and $s \in \{2, 3, 4, 5\}$. For example, for the combination of variables $(n = 2, m = 4, s = 5)$, increasing n with one unit always makes the total calculation time to roughly tenfold in Figures 5.1a–5.1d. In the case $s = 1$, the effect of increasing n is harder to notice as the calculation times are so much smaller than with other sample sizes. This is discussed more below. The effect of increasing either s or m with one unit while the other two variables are held fixed is not nearly as drastic as with n. For example, starting from the situation $(n = 2, m = 4, s = 5)$, see Figure 5.1b, and increasing m unit by unit seems to increase the total calculation time always with only some thousandths of a second. The effect is even smaller if one starts to decrease the value of s while keeping $n = 2$ and $m = 4$. The effect of n, m, and s on calculation time could be studied more thoroughly by, e.g., considering how the number of floating point operations in the scripts of the implementations depends on them. However, such investigation is not in the focus of the current study and is therefore now omitted.

Especially Figures 5.1a–5.1d imply that for given values of n and m the generation times are roughly on the same scale when $s \in \{2, 3, 4, 5\}$ but the times related to $s = 1$ are always noticeably smaller. The main reason for this is related to the number of combinations of sample points that are constructed while generating the CPT. For a given combination of states of the parent nodes $(x_1, ..., x_n)$, s^n combinations of sample points $(z_{1,k}, ..., z_{n,k})_{k=1}^{s^n}$ need to be constructed. While the term s^n increases rapidly for increasing n when $s > 1$, in the case $s = 1$, only one combination of sample points is constructed, independent of the number of the parent nodes. On the other hand, in the case $s > 1$, there are several doubly truncated normal distributions $\{TNormal(\mu_k, \sigma^2, 0, 1)\}_{k=1}^{s^n}$ that are integrated over the state intervals of the child node for a given combination of states of the parent nodes $(x_1, ..., x_n)$. When $s = 1$, only one doubly truncated normal distribution is used. In implementation A, these circumstances mean that the same computational routine, that produces the whole CPT in the case $s = 1$, is repeated for each combination of states of the parent nodes $(x_1, ..., x_n)$ when $s > 1$. This is why the average calculation times in Figure 5.1 are small when $s = 1$ compared to the cases $s \in \{2, 3, 4, 5\}$.

Figure 5.2 displays the average calculation times obtained with implementation B for different values of n, m, and s when using WMEAN as the weight expression. Some of the charts in Figure 5.2 are lacking bars related to different combinations of n, m, and s. These omissions correspond to cases where the calculation times of the CPT are either measured to be several minutes in initial test runs or concluded to be such based on preceding measurements. These cases are left out from the final experiment as recording the repetitions would have been too slow. On the other hand, in these cases, the calculation times are readily considered too long from the practical point of view. These practical aspects are discussed more below.

The results depicted in Figure 5.2 show similarity to those presented in Figure 5.1. Just like in Figure 5.1a, also in Figure 5.2a there are some anomalies. Again, as the magnitude of the calculation times is only thousandths of a second, it is assumed that these anomalies are caused by some overhead in MATLAB that obscures the differences between the different measurements. In general, the bars corresponding to the same set of variables (n, m, s) in Figures 5.1 and 5.2 represent similar times. This can be noticed especially when comparing the graphs in Figures 5.1a–5.1c and 5.2a-5.2c with each other. In these graphs, the range of the time axes are almost the same and the heights of the bars are easy to compare. For the bars presented in Figures 5.2d–5.2f, the similarity to their counterparts in Figures 5.1d–5.1f is harder to observe because the range of the time axes are different. This is because in Figures 5.2d–5.2f, there are bars related to those cases that caused MATLAB run out memory when using implementation A.

The graphs in Figures 5.1 and 5.2 also reveal that the calculation times with implementation B are generally slightly larger than with A for a fixed set of parameters (n, m, s). Thus, it appears that A is computationally a bit lighter than B. The difference between the results of the two implementations is largest in the cases with $s = 1$. This stems from the differences in the operation of the implementations. When $s = 1$, in the second step of implementation A, the built-in functions of matrix calculations in MATLAB can be used effectively to handle the combinations of sample points saved in the first step of the implementation. Because of the same reason, the results obtained with $s = 1$ using A are better than in cases $s > 1$, as discussed earlier. In implementation B, where the combinations of sam-



(a)

(b)

(c)

(d)

(e)

(f)

Figure 5.2: Average calculation times of a CPT with implementation B for varying number of parent nodes n, number of states of the nodes m, and number of sample points s using weight expression WMEAN.

ple points are not constructed beforehand for each combination of states of the parent nodes, the built-in functions of MATLAB cannot be used as effectively as in A when $s = 1$. This is why the calculation times with implementation B are larger than with A when $s = 1$.

Even though the total calculation times are similar for implementations A and

B, the principal difference in their operation makes A faster in practical use. When the CPT is refined and regenerated repeatedly, the combinations of the sample points do not need to be reconstructed in implementation A. Thus, on the repetitive use, the calculation times are only those represented by the upper segments of the bars in Figure 5.1. For example, for the combination of variables $(n = 6, m = 6, s = 2)$, the average recalculation time of the CPT with implementation A corresponding to the upper segment of the highest bar in Figure 5.1f is about 3 s. The corresponding time with implementation B displayed in Figure 5.2f is about 13 s.

The potential advantage that implementation B has over A is that the latter might cause a memory shortage when storing all possible combinations of sample points. However, based on the comparison of Figures 5.1d – 5.1f and 5.2d – 5.2f, it can be seen that for those combinations of (n, m, s) that cause MATLAB run out of memory with implementation A and are included in the experiment with B, the average calculation times of B are ranging from about 30 s, e.g., $(n = 5, m = 6, s = 5)$ in Figure 5.2d, to about 80 s, e.g., $(n = 6, m = 5, s = 4)$ in Figure 5.2f. As already noted, for those combinations of (n, m, s) not included in the experiment with B, the calculation times are measured to be several minutes in some initial tests. Thus, though implementation B provides means to increase the sample point size s above the critical limits of A, the calculation times in these cases become tens of seconds or minutes.

In real-life elicitation sessions, where the parameters of RNM and the CPTs most probably need to be refined repeatedly, calculation times of only few seconds are desirable. Initially one might think that calculation times of about, e.g., 10 seconds are not too uncomfortable. However, if the regeneration needs to be performed several times during the elicitation session, e.g., when adjusting the value of the weight of a parent node, pauses of this magnitude might become irritating and make the elicitation of the parameters feel cumbersome and demotivating. In this light, implementation A is considered preferable to B in practical use because the former is faster inside the range of acceptable calculation times.

The experiment provides knowledge on the computational complexity of RNM. Naturally, the calculation times depend on the speed of the computer and the way the implementation is programmed. If the experiment is carried out with a more effective computer or with more effective implementations, the calculation times would be smaller. However, as explained in the end of Section 5.1.1, the computer set-up of the experiment provides idea of what types of calculation times could be expected in a real-life elicitation session utilizing a standard desktop computer. To gain some insight into the level of effectiveness of the implementations used, some tentative hand-made timer measurements of the calculation times with AgenaRisk are compared to the average calculation times of implementation A in Figure 5.1. In all comparisons, the total times obtained with implementation A are smaller than the ones obtained with AgenaRisk. For example, for the parameter combination $(n = 4, m = 6, s = 5, f =$WMEAN), the average calculation time obtained with implementation A is 1.6 s whereas with AgenaRisk the corresponding calculation times are each about two minutes in three repetitions. Based on these comparisons, the results obtained in the experiment do not give too conservative view of the calculation times of CPTs with RNM in real-life elicitation sessions.

5.1.3 Summary

The results of the experiment indicate that RNM can well be implemented so that its use is sensible from the point of view of generating CPTs fast enough. In various applications of BNs, e.g., [1, 3, 4, 6, 7, 15, 20, 21, 22, 23, 24, 28, 29, 43, 46, 56], the number of parent nodes n for a given child node is typically 1–4 and the numbers of states of the nodes m_i are usually ranging between 2 and 7. For these cases, the implementations of RNM investigated can generate CPTs in just a few seconds independent of the sample size s used. For larger values of n and m_i, the calculation time of RNM might become impractically large or cause memory problems with increasing values of s. As already stated, the practical effect of this feature depends on the computer and the implementation used. Nevertheless, the results of the experiment rise the question of the importance of the value of s used. In Section 4.1, the use of the sample points $z_{i,k}$ is shown to correspond to the approximation of the hierarchical Bayesian model characterized by Equations 4.1–4.4. The accuracy of the discrete approximation increases with increasing s. However, the effect of the sample point size on the modeling performance of RNM in the practical use of the method is unclear. This matter is investigated in the next section as a part of a study concerning the modeling accuracy of RNM.

5.2 Experimental Study on Modeling Accuracy of RNM

As discussed in Chapter 3, RNM is developed to represent quantitatively the weighting schemes through which experts seem to consider the probabilistic relationship between parent nodes and a child node in various applications. In [25], the four weight expressions available in RNM are said to be sufficient for most applications. Over the years, RNM has been utilized mostly in software quality predicting, e.g., [23, 24] but also in other fields, e.g., human reliability analysis [3] and association football gambling [7]. However, there has not been any studies on how RNM performs with BNs in general. That is, when all parent nodes and the child node can be considered as ranked nodes, does RNM provide means to generate a CPT representing the probabilistic relationship between the nodes? The issue is studied in this section experimentally by approximating CPTs in existing real-life benchmark BNs with RNM. This approach for studying the modeling accuracy of RNM is similar to that used in [58] for studying the modeling accuracy of another canonical model, noisy-MAX [17, 52]. In the study now performed, the effects of the sample point size as well as the ranges of the weights and the variance parameter on the modeling accuracy of RNM are studied in particular. Moreover, the effect of using partitioned expressions is investigated. The experimental design of the study is presented in Section 5.2.1. The results are presented in Section 5.2.2. Summary of the experiment is given in Section 5.2.3.

5.2.1 Experimental Design

There are several measures that are used to describe the similarity of two discrete probability distributions [37]. However, the discussion in [58] indicates that there are not any well-established standard ways to measure the similarity of two CPTs. One way is to calculate the similarity for each of the corresponding conditional probability distributions in the CPTs with some standard measure and then use the sum of the values to describe the similarity of the CPTs. However, in [58], it is pointed out that the similarity values related to different conditional probability distributions can be considered of different importance depending on how probable the corresponding combinations of states of the parent nodes are. This corresponds to taking a weighted sum of the similarity measures. Nonetheless, in this study, it is irrelevant how probable the conditioning combinations of states of the parent nodes are. This is because the interest now is how well CPTs constructed with RNM can represent the probabilistic relationship of the parent nodes and the child node.

In the experiments, the similarity of two CPTs Θ and $\tilde{\Theta}$ with $N = m_{n+1} * \prod_{i=1}^{n} m_i$ elements is measured by considering them as points in $\mathbb{R}^N$ and calculating the Euclidean distance between them by

$$\|\Theta - \tilde{\Theta}\| = \sqrt{\sum_{j=1}^{\prod_{k=1}^{n} m_k} \sum_{i=1}^{m_{n+1}} (\Theta_{i,j} - \tilde{\Theta}_{i,j})^2}, \tag{5.1}$$

where $\Theta_{i,j}$ and $\tilde{\Theta}_{i,j}$ are the elements on the ith row and jth column in the CPTs Θ and $\tilde{\Theta}$, respectively, and the columns of the CPTs are conditional probability distributions. This measure is one of those used in [58] to study the modeling accuracy of noisy-MAX.

Minimization Problems

Finding the optimal approximation for a given CPT Θ corresponds to solving the minimization problem

$$\underset{\boldsymbol{p} \in \boldsymbol{P}}{\text{minimize}} \|\Theta - \tilde{\Theta}(\boldsymbol{p}; s)\|, \tag{5.2}$$

where $\boldsymbol{p}$ is a parameter configuration of RNM, $\boldsymbol{P}$ is the set of feasible parameter configurations, $\tilde{\Theta}(\boldsymbol{p}; s)$ is the CPT generated with RNM using $\boldsymbol{p}$ and the sample point size s. Though s is a parameter of RNM, it is not included in the decision variable $\boldsymbol{p}$ in the basic formulation of the optimization problem because its effect is studied separately.

Two different forms of Problem 5.2 are studied. The first one corresponds to the situation where the whole CPT is generated with a single set of parameters, which was discussed in Section 3.3. The second form corresponds to generating the CPT in parts using partitioned expressions in the way presented in Section 3.5.2. The motivation for studying the second form of the minimization problem is to gain better understanding of the utility and importance of the partitioned expressions as a part of the effective use of RNM. The experimental designs for the two forms of the minimization problem are explained in detail below.

In the first form of Problem 5.2, the decision variable $\boldsymbol{p}$ consists of four elements, $\boldsymbol{p}$=$(\boldsymbol{\phi}, f, \boldsymbol{w}, \sigma^2)$. $\boldsymbol{\phi}$ is a vector of n binary elements that define how the states of the parent nodes are mapped to the state intervals. This corresponds to the decision on whether the first state of the parent node is identified to a state interval whose lower bound is 1 or to a state interval whose upper bound is 1, see Figure 3.1. In the experiment, the first state of the child node is always mapped to a state interval having 1 as the upper bound. $f \in F = \{$WMEAN, WMIN, WMAX, MIXMINMAX} is the weight expression used. $\boldsymbol{w}$ and σ^2 are the weights and the variance parameter related to f, respectively. The first form of Problem 5.2 can be written as

$$\begin{aligned} \underset{\boldsymbol{\phi},f,\boldsymbol{w},\sigma^2}{\text{minimize}} \quad & \|\Theta - \tilde{\Theta}(\boldsymbol{\phi}, f, \boldsymbol{w}, \sigma^2; s)\| \\ \text{subject to} \quad & \boldsymbol{\phi} \in \Omega, \\ & f \in F, \\ & \underline{w} \leq w_i \leq \overline{w} \ \ \forall i = 1..., |\boldsymbol{w}|, \\ & \underline{\sigma}^2 \leq \sigma^2 \leq \overline{\sigma}^2, \end{aligned} \tag{5.3}$$

where Ω is a set of 2^n elements that define all possible combinations of mappings of the states of the parent nodes to the normalized scale $[0, 1]$. In addition, $\underline{w}$, $\overline{w}$, $\underline{\sigma}^2$, and $\overline{\sigma}^2$ are the lower and upper bounds of the weights $\boldsymbol{w} = (w_1, ..., w_{|\boldsymbol{w}|})$ and the variance parameter σ^2, respectively. $|\cdot|$ is the cardinality measure, i.e., it tells the number of elements in an array. For example, when f =MIXMINMAX, $|\boldsymbol{w}| = 2$ but for other values of f, $|\boldsymbol{w}| = n$, i.e., equal to the number of the parent nodes.

In the second form of Problem 5.2, the partitioning of the CPT is performed with respect to the states of one parent node. In this case, the decision variable $\boldsymbol{p}$ consists of five elements, $\boldsymbol{p}$=$(X, \boldsymbol{\phi}, \boldsymbol{f}, W, \boldsymbol{\sigma}^2)$. X is the parent node according to whose states the CPT is divided into m_X segments and generated with partitioned expressions. $\boldsymbol{\phi}$ is vector that defines the mappings of the states of the parent nodes to the state intervals. $\boldsymbol{f} = (f_1, ..., f_{m_X})$ defines the weight expressions used in the partitioned expressions while $W = (\boldsymbol{w}^1, ..., \boldsymbol{w}^{m_X})$ and $\boldsymbol{\sigma^2} = (\sigma_1^2, ..., \sigma_{m_X}^2)$ contain the weights and variance parameters, respectively. Hence, the minimization problem is

$$\begin{aligned} \underset{X,\boldsymbol{\phi},\boldsymbol{f},W,\boldsymbol{\sigma}^2}{\text{minimize}} \quad & \|\Theta - \tilde{\Theta}(X, \boldsymbol{\phi}, \boldsymbol{f}, W, \boldsymbol{\sigma}^2; s)\| \\ \text{subject to} \quad & X \in \{X_1, ..., X_n\}, \\ & \boldsymbol{\phi} \in \Omega, \\ & f_i \in F \ \ \forall i = 1, ..., m_X, \\ & \underline{w} \leq w_j^i \leq \overline{w} \ \ \forall j = 1, ..., |\boldsymbol{w^i}|, \forall i = 1, ..., m_X, \\ & \underline{\sigma}^2 \leq \sigma_i^2 \leq \overline{\sigma}^2, \ \ \forall i = 1, ..., m_X, \end{aligned} \tag{5.4}$$

where w_i^j is the ith element of the vector $\boldsymbol{w}^j$.

Effect of Sample Point Size and Parameter Ranges

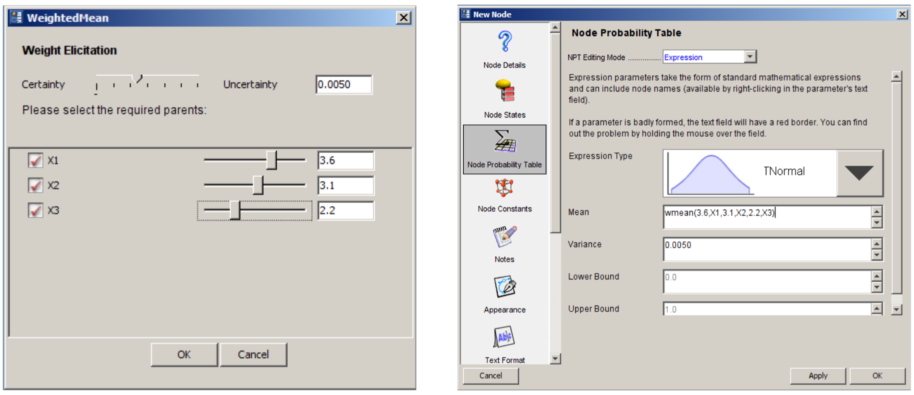
Problems 5.3 and 5.4 are solved separately for different values of the sample point size s in order to investigate its effect on the modeling accuracy of RNM. In Section 4.1, the use of sample points $z_{i,k}$ in RNM is shown to correspond to the approximation of the hierarchical Bayesian model characterized by Equations 4.1–4.4. It is shown that the approximation becomes more accurate with the increasing value of sample point size s, see Equation 4.5. Thus, for a fixed parameter configuration $\boldsymbol{p}$, the measure $\|\Theta - \tilde{\Theta}(\boldsymbol{p}; s)\|$ should also converge for an increasing value of s. From the point of view of the minimization problem 5.2, this corresponds to the phenomenon that the optimal configuration $(\boldsymbol{p}^*; s)$ converges to some specific limit $\boldsymbol{p}^*$.

In addition to studying the effect of the sample point size s on the modeling performance of RNM, the effect of the parameter ranges is investigated as well. For each value of s used, Problems 5.3 and 5.4 are solved using two values for the bounds $\{\underline{w}, \overline{w}, \underline{\sigma}^2, \overline{\sigma}^2\}$. The first set is

$$\begin{cases} \underline{w} &= 1, \\ \overline{w} &= 5, \\ \underline{\sigma}^2 &= 5 * 10^{-4}, \\ \overline{\sigma}^2 &= 0.5. \end{cases} \tag{5.5}$$

These values correspond to the bounds set to the weights and the variance parameter in the elicitation window of AgenaRisk that is displayed in Figure 5.3a. In this window, one can set the values of the parameters using scroll bars. The values of the parameters can also be set beyond these ranges in another elicitation window of AgenaRisk that is displayed in Figure 5.3b. In this window, the values are set directly without any visual support. Because of this, one may easily prefer to use the elicitation window displayed in Figure 5.3a. In turn, this may cause the user to get anchored to the restricted ranges available. That is, the user may settle to use only those ranges of the parameters that are available in the elicitation window and not consider values beyond these bounds.

To investigate how severe effects anchoring to the bounds in Equation 5.5 can have on the modeling accuracy of RNM, Problems 5.3 and 5.4 are solved also



WeightedMean

Weight Elicitation

Certainty / Uncertainty: 0.0050

Please select the required parents:

- ☑ X1: 3.6
- ☑ X2: 3.1
- ☑ X3: 2.2

(a)

New Node

Node Probability Table

NPT Editing Mode .... Expression

Expression parameters take the form of standard mathematical expressions and can include node names (available by right-clicking in the parameter's text field).

If a parameter is badly formed, the text field will have a red border. You can find out the problem by holding the mouse over the field.

Expression Type: TNormal

Mean: wmean(3.6,X1,3.1,X2,2.2,X3)

Variance: 0.0050

Lower Bound: 0.0

Upper Bound: 1.0

(b)

Figure 5.3: Windows for setting parameters of RNM in AgenaRisk software.

with the bounds

$$\begin{cases} \underline{w} & = 1, \\ \overline{w} & = \infty, \\ \underline{\sigma}^2 & = 10^{-6}, \\ \overline{\sigma}^2 & = \infty. \end{cases} \tag{5.6}$$

These correspond to bounds that are plausible from the theoretical point of view. Being the variance of a doubly truncated normal distribution, σ^2 can have any value in $[\epsilon, \infty)$, where $\epsilon > 0$. The value $\epsilon = 10^{-6}$ is now used as the lower bound for σ^2. Similarly to the variance parameter, there does not need to be any finite upper limit for the weights of the parent nodes in RNM. Thus, the value of $\overline{w} = \infty$. The lower limit $\underline{w}$ is set to 1 due to the reasons discussed in Section 4.2.2.

Solving the Minimization Problems

Problem 5.3 is solved by finding the optimal values of $\boldsymbol{w}$ and σ^2 for each combination of $\boldsymbol{\phi}$ and f and taking the best one of the resulting set of solutions. That is, one first solves

$$\begin{aligned} \underset{\boldsymbol{w},\sigma^2}{\text{minimize}} \quad & \|\Theta - \tilde{\Theta}(\boldsymbol{w}, \sigma^2; \boldsymbol{\phi}, f, s)\| \\ \text{subject to} \quad & \underline{w} \leq w_i \leq \overline{w} \ \ \forall i = 1, \ldots, |\boldsymbol{w}|, \\ & \underline{\sigma}^2 \leq \sigma^2 \leq \overline{\sigma}^2, \end{aligned} \tag{5.7}$$

for each $\boldsymbol{\phi} \in \Phi$ and $f \in F$. Then, the final solution $\|\Theta - \tilde{\Theta}; s\|^*$ is obtained by

$$\|\Theta - \tilde{\Theta}; s\|^* = \min_{\boldsymbol{\phi} \in \Phi, f \in F} \|\Theta - \tilde{\Theta}; \boldsymbol{\phi}, f, s\|^*, \tag{5.8}$$

where $\|\Theta - \tilde{\Theta}; \boldsymbol{\phi}, f, s\|^*$ is the set of solutions to Problem 5.7.

Problem 5.4 is solved by finding the optimal values of $\boldsymbol{f}$, W and $\boldsymbol{\sigma}^2$ for each combination of X and $\boldsymbol{\phi}$ and taking the best one of the resulting set of solutions. For a given combination of X and $\boldsymbol{\phi}$, the optimization proceeds as follows. First, in each segment Θ^i of the partition and with each value of $f_i \in F$, one solves the problem

$$\begin{aligned} \underset{\boldsymbol{w}^i, \sigma_i^2}{\text{minimize}} \quad & \|\Theta^i - \tilde{\Theta}^i(\boldsymbol{w}^i, \sigma_i^2; X, \boldsymbol{\phi}, f_i, s)\| \\ \text{subject to} \quad & \underline{w} \le w_j^i \le \overline{w} \;\; \forall j = 1, ..., |\boldsymbol{w}^i|, \\ & \underline{\sigma}^2 \le \sigma_i^2 \le \overline{\sigma}^2. \end{aligned} \tag{5.9}$$

Next, the best solutions over the weight expressions in each segment $\|\Theta^i - \tilde{\Theta}^i; X, \boldsymbol{\phi}, s\|^*$ are given by

$$\|\Theta^i - \tilde{\Theta}^i; X, \boldsymbol{\phi}, s\|^* = \min_{f_i \in F} \|\Theta^i - \tilde{\Theta}^i; X, \boldsymbol{\phi}, f_i, s\|^*, \tag{5.10}$$

where $\|\Theta^i - \tilde{\Theta}^i; X, \boldsymbol{\phi}, f_i, s\|^*$ are the optimal solutions to Problem 5.9.

Finally, the corresponding overall Euclidean distance for the given values of X and $\boldsymbol{\phi}$ is obtained according to

$$\|\Theta - \tilde{\Theta}; X, \boldsymbol{\phi}, s\|^* = \sqrt{\sum_{i=1}^{m_X} \left(\|\Theta^i - \tilde{\Theta}^i; X, \boldsymbol{\phi}, s\|^* \right)^2}. \tag{5.11}$$

After calculating the optimal solutions for all combinations of X and $\boldsymbol{\phi}$, the optimal solution $\|\Theta - \tilde{\Theta}; s\|^*$ is given by

$$\|\Theta - \tilde{\Theta}; s\|^* = \min_{X \in \{X_1, \ldots, X_n\}, \boldsymbol{\phi} \in \Omega} \|\Theta - \tilde{\Theta}; X, \boldsymbol{\phi}, s\|^*. \tag{5.12}$$

In practice, Problems 5.7 and 5.9 are solved with MATLAB utilizing the implementation A of RNM presented in Section 5.1. The optimization in MATLAB is performed with the routine *patternsearch* designed for non-smooth optimization problems. To show whether the problems 5.7 and 5.9 have unique solutions or not is a non-trivial problem itself and beyond the scope of this thesis. Because of the lack of knowledge concerning the existence of unique global minima, *fmincon* is set to find local minima of the problems from three alternative initial points of the iteration $(\boldsymbol{w}_{0,i}, \sigma_{0,i}^2)_{i=1}^3$. By using different initial points, the chance of finding the smallest local minimum increases. When solving Problems 5.7 and 5.9, the final results are obtained by taking the smallest of the local minima that the iterations with different initial points produce. When the bounds in Equation 5.6 are used, the initial points of the iteration are

$$\begin{cases} (\boldsymbol{w}_{0,1}, \sigma_{0,1}^2) & = (1, ..., 1, 10^{-6}), \\ (\boldsymbol{w}_{0,2}, \sigma_{0,2}^2) & = (50, ..., 50, 0.25), \\ (\boldsymbol{w}_{0,3}, \sigma_{0,3}^2) & = (1000, ..., 1000, 1). \end{cases} \tag{5.13}$$

With the bounds in Equation 5.5, the initial points of the iteration are

$$\begin{cases} (\boldsymbol{w}_{0,1}, \sigma_{0,1}^2) & = (1, ..., 1, 5 * 10^{-4}), \\ (\boldsymbol{w}_{0,2}, \sigma_{0,2}^2) & = (2.5, ..., 2.5, 0.25), \\ (\boldsymbol{w}_{0,3}, \sigma_{0,3}^2) & = (5, ..., 5, 0.5). \end{cases} \tag{5.14}$$

As discussed in Sections 4.2.1 and 4.2.4, only the relative sizes of the weights matter in WMEAN and MIXMINMAX. Thus, in the case of these two weight expressions, the optimal weights related to different initial points of iteration are normalized to the scale $[0, 1]$ in the end. This enables that when analyzing the uniqueness of the optimal solutions, one readily detects those optimal solutions that are seemingly different but actually correspond to the same relative weights.

Measures of Goodness of Fit

The Euclidean distance defined in Equation 5.1 is a sensible measure to describe the similarity of two CPTs of given size. However, the measure is not adequate to present and compare the accuracy of the optimal approximate CPTs when the CPTs are of different sizes. This is because the value of the Euclidean distance grows as the size of the CPT grows, even if the magnitude of the errors in elements would remain the same. Thus, instead of examining the optimal values $||\Theta - \tilde{\Theta}; s||^*$ in Equations 5.8 and 5.12, two other measures are used in the analysis of the optimization results. First one is the mean absolute error (MAE) defined by

$$||\Theta - \tilde{\Theta}^*||_{MAE} = \frac{1}{N} \sum_{j=1}^{\prod_{k=1}^{n} m_k} \sum_{i=1}^{m_{n+1}} |\Theta_{i,j} - \tilde{\Theta}_{i,j}|, \tag{5.15}$$

where $N = m_{n+1} \prod_{k=1}^{n} m_k$ and $\tilde{\Theta}^*$ is the approximate CPT related to the optimal solution of the given minimization problem. MAE is the average of the absolute errors in the elements of $\tilde{\Theta}^*$ and describes the average goodness of the fit of the approximation. The other measure used to analyze the optimization results is the maximum absolute error (MAX) defined by

$$\|\Theta - \tilde{\Theta}^*\|_{MAX} = \max_{\substack{j=1,\ldots,\prod_{k=1}^{n} m_k \\ i=1,\ldots,m_{n+1}}} |\Theta_{i,j} - \tilde{\Theta}_{i,j}|. \tag{5.16}$$

MAX is the maximum absolute error in the elements of $\tilde{\Theta}^*$ and describes the worst fit for a single element in the approximation.

The MAE and MAX measures provide means to describe and compare the goodness of the fit of the approximate CPTs independent of the size of the CPTs. As these measures are used in the analysis of the results also in [58], the modeling accuracy of RNM can be compared to the modeling accuracy of noisy-MAX.

Bayesian Networks Investigated

Three benchmark BNs are used in the experiment: *Alarm* [4], *Hailfinder* [1], and *Insurance* [6]. *Alarm* models patients in intensive medical care. *Hailfinder* is developed to forecast severe weather in North-Eastern Colorado, US. *Insurance* is a model for estimating the expected claim costs for a car insurance policyholder. The CPTs in *Alarm* and *Hailfinder* are known to be assessed by human experts [57]. The background of the CPTs in *Insurance* is not specified in [6]. *Alarm* and *Hailfinder* are also used in the earlier study concerning noisy-MAX [58]. The third BN studied in [58] is left out from this experiment as it does not contain a single group of parent nodes and a child node where the number of parent nodes is at least two and each of the nodes in the group can be considered to be a ranked node. Instead, *Insurance* is used as it contains such groups.

5.2.2 Results

In total, 22 suitable groups of parent nodes and a child node are in the BNs used. Table 5.2.2 summarizes numerical characteristics of the related CPTs. The results obtained to Problems 5.3 and 5.4 are examined in separate sections below.

Results of Problem 5.3

The analysis of the results of Problem 5.3 begins by examining the MAE values of the optimal approximations $\tilde{\Theta}^*$ obtained using the more restricted bounds in Equation 5.5. Figure 5.4 displays the MAE values of the optimal approximations

Table 5.1: Characteristics of the CPTs approximated in the experimental study. m_i refers to the number of states of the parent nodes and the child node. On each row, the rightmost m_i is the number of states of the child node. CPT size refers to the number of elements in the CPT.

Child node	BN	m_1	m_2	m_3	m_4	m_5	CPT size
SAO2	Alarm	2	3	3	-	-	18
CO	Alarm	3	3	3	-	-	27
BP	Alarm	3	3	3	-	-	27
CombMoisture	Hailfinder	4	4	4	-	-	64
AreaMoDryAir	Hailfinder	4	4	4	-	-	64
CombClouds	Hailfinder	3	3	3	-	-	27
CldShadeOth	Hailfinder	4	3	3	4	-	144
InsInMt	Hailfinder	3	3	3	-	-	27
InsChange	Hailfinder	4	3	3	-	-	36
CapInScen	Hailfinder	3	3	3	-	-	27
InsSclInScen	Hailfinder	3	3	3	-	-	27
CurPropConv	Hailfinder	4	4	4	-	-	64
CombVerMo	Hailfinder	4	4	4	4	-	256
CompPlFcst	Hailfinder	3	3	4	3	3	324
RiskAversion	Insurance	3	4	4	-	-	48
HomeBase	Insurance	4	4	4	-	-	64
DrivQuality	Insurance	3	4	3	-	-	36
VehicleYear	Insurance	4	4	2	-	-	32
MedCost	Insurance	4	3	4	4	-	192
OtherCarCost	Insurance	4	3	4	-	-	48
ThisCarDam	Insurance	4	3	4	-	-	48
PropCost	Insurance	4	4	4	-	-	64

obtained with sample point sizes $s = 1, ..., 5$. Table 5.2 presents the averages and medians of the MAE values for the different values of s.

Figure 5.4 implies that the MAE values obtained with different values of s are generally similar for a given node. This is also reflected by the similarity of the averages and medians of the MAE values related to different sample point sizes, see Table 5.2. However, in Figure 5.4, there are especially two cases, *PropCost* and *SAO2*, where there are distinctive differences in the MAE values related to different values of s. In both of these cases, the MAE value related to $s = 1$ is distinctively smaller than the values related to other sample point sizes. The common trend for all the nodes in Figure 5.4 is that the MAE values related to

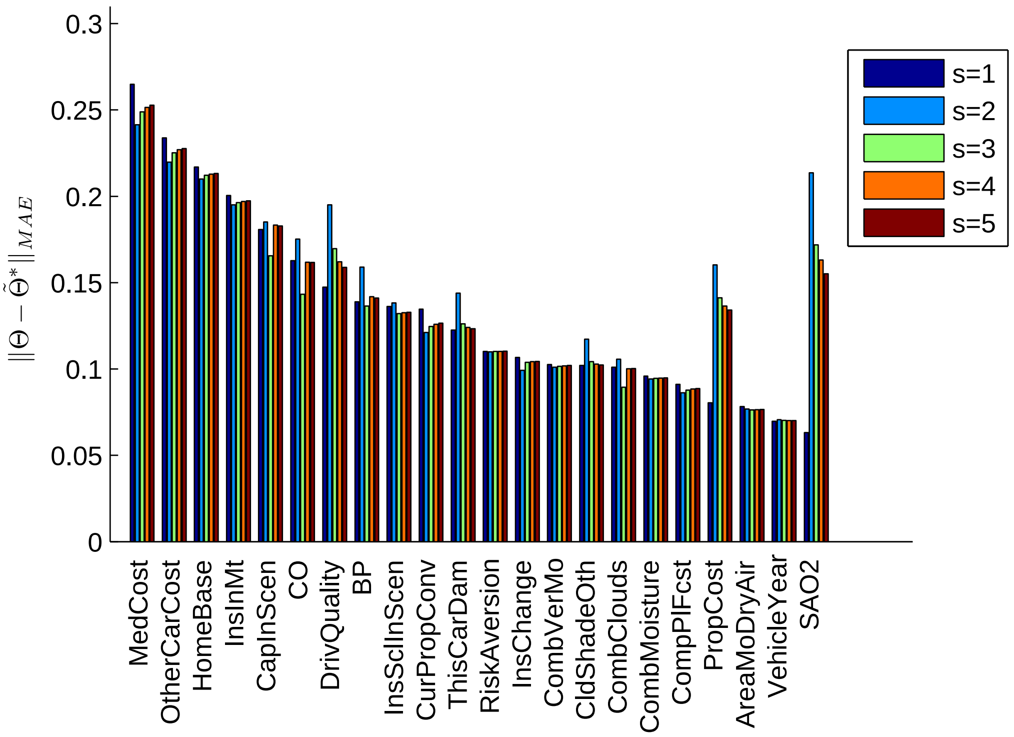


Figure 5.4: MAE values of the solutions to Problem 5.3 with the bounds in Equation 5.5.

Table 5.2: Averages, $Av(\|\Theta - \tilde{\Theta}^*\|_{MAE})$, and medians, $Md(\|\Theta - \tilde{\Theta}^*\|_{MAE})$, of the MAE values related to the optimal solutions of Problem 5.3 obtained using the bounds in Equation 5.5 and the sample point size s.

	s				
	1	2	3	4	5
$Av(\|\Theta - \tilde{\Theta}^*\|_{MAE})$	0.134	0.146	0.138	0.140	0.139
$Md(\|\Theta - \tilde{\Theta}^*\|_{MAE})$	0.116	0.141	0.129	0.129	0.130

$s = 1$ and $s = 2$ tend to differ the most from the values related to other values of s. In particular, the values related to $s = 4$ and $s = 5$ are always the closest to each other. These observations are in line with the prediction made concerning the convergence of the optimal solutions $(\boldsymbol{\phi}^*, f^*, \boldsymbol{w}^*, \sigma^{2*}; s)$ with increasing s. That is, the MAE values related to $s = 1$ and $s = 2$ tend to be the most distinctive because the approximations $\tilde{\Theta}^*$ obtained with given parameters have not yet converged, i.e., the elements have not yet "settled" to their final values. From this perspective, the nodes for which the MAE values are similar for different values of s correspond to cases of rapid convergence. On the other hand, for nodes such as *PropCost* and *SAO2*, the convergence of the optimal values oscillates more.

The study of the optimal solutions verifies them to be terms of converging sequences. For example, in Figure 5.4, *RiskAversion* is one of the nodes for which the optimal solutions are expected to be terms of a rapidly converging sequence. Table 5.3 presents the optimal solutions $(\boldsymbol{\phi}^*, f^*, \boldsymbol{w}^*, \sigma^{2*}; s)$ and the related MAE values obtained for the node with $s = 1, ..., 8$. The weight expression f^*, the mapping vector $\boldsymbol{\phi}^*$, and the weight of the first parent node w_1 are fixed from $s = 1$ onwards. The other parameters also converge as s increases.

Table 5.3: Optimal solutions and MAE values related to *RiskAversion* for Problem 5.3 using bounds in Equation 5.5.

s	f^*	$\boldsymbol{\phi}^*$	$\boldsymbol{w}^*$	σ^{2*}	$\|\Theta - \tilde{\Theta}^*\|$	$\|\Theta - \tilde{\Theta}^*\|_{MAE}$
1	WMIN	(0, 1)	(5.0, 1.33)	0.0724	1.00	0.110
2	WMIN	(0, 1)	(5.0, 1.51)	0.0632	0.986	0.110
3	WMIN	(0, 1)	(5.0, 1.45)	0.0663	0.990	0.110
4	WMIN	(0, 1)	(5.0, 1.43)	0.0673	0.992	0.110
5	WMIN	(0, 1)	(5.0, 1.42)	0.0678	0.993	0.110
6	WMIN	(0, 1)	(5.0, 1.42)	0.0681	0.993	0.110
7	WMIN	(0, 1)	(5.0, 1.41)	0.0683	0.993	0.110
8	WMIN	(0, 1)	(5.0, 1.41)	0.0684	0.993	0.110

Table 5.4 presents the optimal solutions of *SAO2* for $s = 1, ..., 8$. The optimal solutions verify the trend observable in Figure 5.4 — the optimal values of *SAO2* oscillate at first but then they clearly converge with increasing s.

The results of *SAO2* are also an example of the possible non-uniqueness of the optimal solutions. For $s = 1$ and $s = 3, ..., 8$, the same optimal value $\|\Theta - \tilde{\Theta}^*\|$ is obtained with two different weight expressions, WMAX and MIXMINMAX. These cases are embodiments of the phenomenon that with suitable weights, the functional forms of WMAX and MIXINMAX become identical. The two weight expressions become functionally identical when there are two parent nodes, which receive weights $\boldsymbol{w}^{WMAX} = (w, w)$ in WMAX and $\boldsymbol{w}^{MIX} = (\frac{1}{w+1}, \frac{w}{w+1})$ in MIXMINMAX. Using these weights in the functions of the weight expressions in Equations

Table 5.4: Optimal solutions related to *SAO2* for Problem 5.3 using bounds in Equation 5.5.

s	f^*	$\boldsymbol{\phi}^*$	$\boldsymbol{w}^*$	σ^{2*}	$\lVert\Theta - \tilde{\Theta}^*\rVert$	$\lVert\Theta - \tilde{\Theta}^*\rVert_{MAE}$
1	WMAX	$(1, 0)$	$(5.0, 5.0)$	0.0018	0.507	0.0631
	MIXMINMAX	$(1, 0)$	$(0.167, 0.833)$	0.0018	0.507	0.0631
2	WMAX	$(1, 0)$	$(3.19, 5.0)$	0.0106	1.119	0.214
3	WMAX	$(1, 0)$	$(5.0, 5.0)$	$5.0 * 10^{-4}$	0.938	0.172
	MIXMINMAX	$(1, 0)$	$(0.167, 0.833)$	$5.0 * 10^{-4}$	0.938	0.172
4	WMAX	$(1, 0)$	$(5.0, 5.0)$	$9.1 * 10^{-4}$	0.922	0.163
	MIXMINMAX	$(1, 0)$	$(0.167, 0.833)$	$9.1 * 10^{-4}$	0.922	0.163
5	WMAX	$(1, 0)$	$(5.0, 5.0)$	$5.0 * 10^{-4}$	0.888	0.155
	MIXMINMAX	$(1, 0)$	$(0.167, 0.833)$	$5.0 * 10^{-4}$	0.888	0.155
6	WMAX	$(1, 0)$	$(5.0, 5.0)$	$5.7 * 10^{-4}$	0.876	0.151
	MIXMINMAX	$(1, 0)$	$(0.167, 0.833)$	$5.7 * 10^{-4}$	0.876	0.151
7	WMAX	$(1, 0)$	$(5.0, 5.0)$	$5.0 * 10^{-4}$	0.865	0.149
	MIXMINMAX	$(1, 0)$	$(0.167, 0.833)$	$5.0 * 10^{-4}$	0.865	0.149
8	WMAX	$(1, 0)$	$(5.0, 5.0)$	$5.0 * 10^{-4}$	0.856	0.147
	MIXMINMAX	$(1, 0)$	$(0.167, 0.833)$	$5.0 * 10^{-4}$	0.856	0.147

3.3 and 3.4, respectively, one obtains for a pair of sample points (a, b)

$$\begin{aligned} WMAX(a, b, w, w) &= \max\left\{\frac{w * a + b}{w + 1}, \frac{w * b + a}{w + 1}\right\} \\ &= \frac{w}{w+1}\max\{a, b\} + \frac{1}{w+1}\min\{a, b\} \qquad (5.17) \\ &\equiv MIXMINMAX\left(a, b, \frac{1}{w+1}, \frac{w}{w+1}\right). \end{aligned}$$

Whereas the optimal solutions of *SAO2* have WMAX and MIXMINMAX coinciding, the same can occur with WMIN and MIXMINMAX as well. By referring to the definitions of these functions in Equations 3.2 and 3.4, for a pair of sample points (a, b) one obtains

$$\begin{aligned} WMIN(a, b, w, w) &= \min\left\{\frac{w * a + b}{w + 1}, \frac{w * b + a}{w + 1}\right\} \\ &= \frac{w}{w+1}\min\{a, b\} + \frac{1}{w+1}\max\{a, b\} \qquad (5.18) \\ &\equiv MIXMINMAX\left(a, b, \frac{w}{w+1}, \frac{1}{w+1}\right). \end{aligned}$$

In total, six of the 22 nodes investigated are discovered to have non-unique optimal solutions in the sense that either WMAX or WMIN coincide with MIXMINMAX. The connections between the weight expressions depicted by Equations 5.17 and 5.18 indicate that in some cases, the probabilistic relationship between ranked nodes can be portrayed or understood in two different ways. The implications of this observation are discussed more in Section 5.2.3.

There are 16 nodes for which WMAX or WMIN do not coincide with MIXMINMAX in the optimal solutions. In 13 of these cases, a unique optimal solution is found for each value of s, similarly to *RiskAversion*. That is, in these 13 cases, the optimal weight expression f^* and the optimal mapping vector $\boldsymbol{\phi}^*$ are the same for a given node with each value of s. In addition, the optimal weights $\boldsymbol{w}^*$ and the variance parameter σ^{2*} converge with increasing s. Behaviour different from this common trend is observed for three nodes, *InsChange*, *CombVerMo*, and *VehicleYear*. These cases are next discussed separately.

In the case of *InsChange*, the optimal weight expression varies depending on the value of s, see Table 5.5. For $s = 1, 4, 5$, the optimal weight expression is WMEAN but for $s = 2, 3$, it is WMIN.

Table 5.5: Optimal solutions related to *InsChange* for Problem 5.3 using bounds in Equation 5.5.

s	f^*	$\boldsymbol{\phi}^*$	$\boldsymbol{w}^*$	σ^{2*}	$\|\Theta - \tilde{\Theta}^*\|$	$\|\Theta - \tilde{\Theta}^*\|_{MAE}$
1	WMEAN	(1, 1)	(0.828, 0.173)	0.033	0.802	0.107
2	WMIN	(1, 1)	(5.0, 1.0)	0.023	0.757	0.099
3	WMIN	(1, 1)	(5.0, 1.0)	0.026	0.779	0.102
4	WMEAN	(1, 1)	(0.825, 0.175)	0.028	0.783	0.104
5	WMEAN	(1, 1)	(0.825, 0.175)	0.029	0.785	0.104

Table 5.6 presents the optimization results obtained with WMIN for $s = 1, 4, 5$ and with WMEAN for $s = 2, 3$. Comparing these results to those in Table 5.5 reveals that for a given value of s, the optimal Euclidean distance and the MAE values obtained with WMEAN and WMIN are close to each other. Thus, the results related to *InsChange* are another example of how different ways of considering the probabilistic relationship of ranked nodes may lead to similar CPTs.

In the case of *CombVerMo*, the optimal weights $\boldsymbol{w}^*$ are not necessarily unique for a given value of s, see Table 5.7. Now, for $s = 1, 3, 5$, the different optimal weights generate different approximate CPTs that have an equal Euclidean distance to the target CPT. This represents the idea that different paramterizations of RNM can generate CPTs that cannot be said to be unambiguously better or worse repre-

Table 5.6: Optimal solutions related to *InsChange* for Problem 5.3 using bounds in Equation 5.5 and a given weight expression.

s	f	$\boldsymbol{\phi}^*\|f$	$\boldsymbol{w}^*\|f$	$\sigma^{2*}\|f$	$\|\|\Theta - \tilde{\Theta}^*\|\|\|f$	$\|\|\Theta - \tilde{\Theta}^*\|\|_{MAE}\|f$
1	WMIN	(1, 1)	(5.0, 1.0)	0.032	0.818	0.108
2	WMEAN	(1, 1)	(0.823, 0.177)	0.024	0.771	0.103
3	WMEAN	(1, 1)	(0.824, 0.176)	0.027	0.780	0.104
4	WMIN	(1, 1)	(5.0, 1.0)	0.027	0.787	0.104
5	WMIN	(1, 1)	(5.0, 1.0)	0.028	0.790	0.104

sentations of the probabilistic relationship of the parent nodes and the child node when compared to each other.

Table 5.7: Optimal solutions related to *CombVerMo* for Problem 5.3 using bounds in Equation 5.5.

s	f^*	$\boldsymbol{\phi}^*$	$\boldsymbol{w}^*$	σ^{2*}	$\|\|\Theta - \tilde{\Theta}^*\|\|$	$\|\|\Theta - \tilde{\Theta}^*\|\|_{MAE}$
1	WMIN	(0, 0, 0)	(1.329, 1.127, 1.229)	0.022	2.374	0.103
	WMIN	(0, 0, 0)	(1.329, 1.229, 1.127)	0.022	2.374	0.103
2	WMIN	(0, 0, 0)	(1.192, 1.278, 1.241)	0.017	2.339	0.101
3	WMIN	(0, 0, 0)	(1.237, 1.265, 1.205)	0.019	2.351	0.102
	WMIN	(0, 0, 0)	(1.205, 1.237, 1.265)	0.019	2.351	0.102
4	WMIN	(0, 0, 0)	(1.212, 1.236, 1.258)	0.019	2.355	0.102
5	WMIN	(0, 0, 0)	(1.294, 1.089, 1.276)	0.020	2.357	0.102
	WMIN	(0, 0, 0)	(1.294, 1.276, 1.089)	0.020	2.357	0.102

In the case of *VehicleYear*, there are always two optimal solutions for a given value of s, see Table 5.8. The optimal solutions are identical except for the mapping vector $\boldsymbol{\phi}^*$. The way the states of the second parent node are mapped to the normalized scale $[0, 1]$ do not make any difference to the results. The reason for this is discovered by examining the CPT of *VehicleYear*. It turns out that the other parent node does not affect the child node in any way. Hence, the states of this parent node can be mapped to $[0, 1]$ in either way which explains the non-unique optimal solutions.

Figure 5.5 presents the MAX values related to the optimal solutions of Problem 5.3 obtained using the bounds in Equation 5.5 and sample point sizes $s = 1, ..., 5$. Table 5.9 presents the averages and medians of the MAX values for the values of s.

The MAX values obtained with alternative s presented in Figure 5.5 are similar

Table 5.8: Optimal solutions related to *VehicleYear* for Problem 5.3 using bounds in Equation 5.5.

s	f^*	$\boldsymbol{\phi}^*$	$\boldsymbol{w}^*$	σ^{2*}	$\|\Theta - \tilde{\Theta}^*\|$	$\|\Theta - \tilde{\Theta}^*\|_{MAE}$
1	WMEAN	$(1, 0)$	$(0.833, 0.167)$	0.044	0.508	0.070
	WMEAN	$(1, 1)$	$(0.833, 0.167)$	0.044	0.508	0.070
2	WMEAN	$(1, 0)$	$(0.833, 0.167)$	0.036	0.517	0.071
	WMEAN	$(1, 1)$	$(0.833, 0.167)$	0.036	0.517	0.071
3	WMEAN	$(1, 0)$	$(0.833, 0.167)$	0.039	0.512	0.070
	WMEAN	$(1, 1)$	$(0.833, 0.167)$	0.039	0.512	0.070
4	WMEAN	$(1, 0)$	$(0.833, 0.167)$	0.040	0.511	0.070
	WMEAN	$(1, 1)$	$(0.833, 0.167)$	0.040	0.511	0.070
5	WMEAN	$(1, 0)$	$(0.833, 0.167)$	0.040	0.511	0.070
	WMEAN	$(1, 1)$	$(0.833, 0.167)$	0.040	0.511	0.070

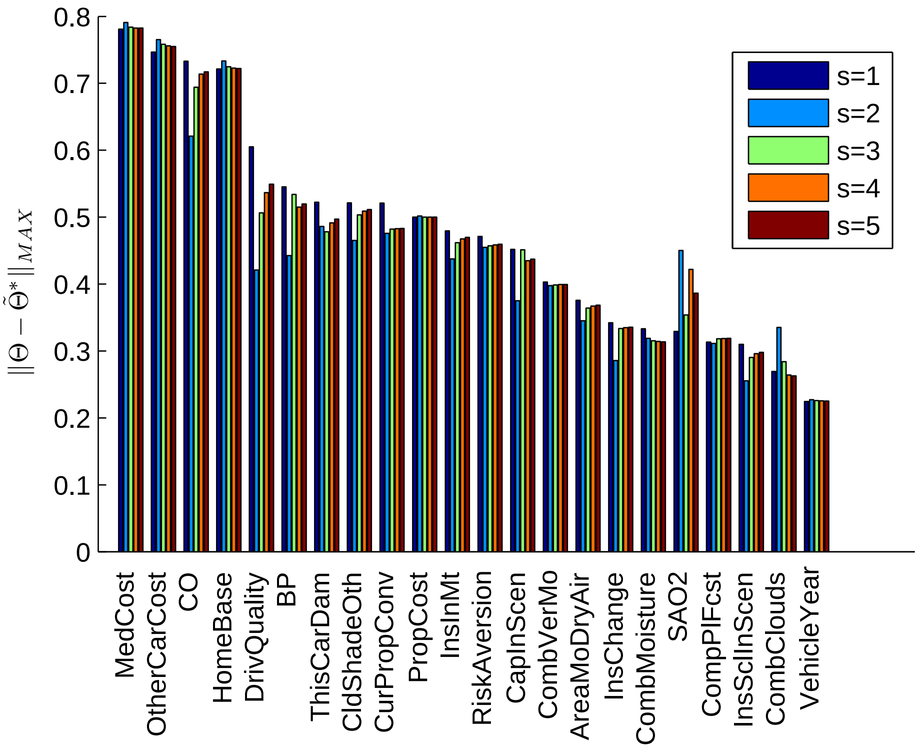


Figure 5.5: MAX values of the solutions to Problem 5.3 with the bounds in Equation 5.5.

Table 5.9: Averages, $Av(\|\Theta - \tilde{\Theta}^*\|_{MAX})$, and medians, $Md(\|\Theta - \tilde{\Theta}^*\|_{MAX})$, of the MAX values related to the optimal solutions of Problem 5.3 obtained using the bounds in Equation 5.5 and the sample point size s.

	s				
$Av(\|\Theta - \tilde{\Theta}^*\|_{MAX})$	0.477	0.450	0.465	0.469	0.469
$Md(\|\Theta - \tilde{\Theta}^*\|_{MAX})$	0.475	0.440	0.459	0.463	0.465

for a given node. Especially the MAX values related to $s = 4$ and $s = 5$ are always close to each other. Thus, the convergence of the optimal solutions can be observed from Figure 5.5 similarly to Figure 5.4. The likeness of the MAX values is also reflected by the similarity of the averages and medians presented in Table 5.9.

The MAX values in Figure 5.5 are considerably larger than the MAE values in Figure 5.4 for each node. This becomes apparent also by comparing the averages and medians of the MAE and MAX values presented in Tables 5.2 and 5.9, respectively. Whereas the averages and medians of the MAE values vary roughly between 0.11–0.14, the corresponding range for the averages and medians of the MAX values is roughly 0.44–0.48. Thus, the accuracy of the optimal approximate CPT $\tilde{\Theta}^*$ may be worse in some elements than it is on average. This indicates that in many cases the probabilistic relationship between the nodes does not purely correspond to the form of a single weight expression.

To get an idea of how "good" the modeling accuracy of RNM is, the results can be compared to those obtained for noisy-MAX in a similar study reported in [58]. For noisy-MAX, the medians of the MAE values of the 17 nodes from *Alarm* and 19 nodes from *Hailfinder* are less than 0.01 in both cases. In turn, the medians of the MAX values related to *Alarm* and *Hailfinder* in [58] are less than 0.1 and equal to roughly 0.2, respectively. As noted above, the medians of the MAE and MAX values now obtained vary in ranges 0.11–0.14 and 0.44–0.48, respectively. In [58], the names of the nodes are not presented. Therefore, a full comparison of the results between the experiments is not possible. Despite of this, the comparison of the median values for the MAE and MAX measures suggests that the modeling accuracy of RNM is poorer than that of noisy-MAX. The difference in the results indicates that a majority of the CPTs approximated represent probabilistic relationships that correspond better to the basic idea of noisy-MAX rather than of RNM, see the discussion in Section 4.4. Thus, the results indicate that the probabilistic relationships that RNM is designed to represent are not as common in applications of BNs as the ones corresponding to noisy-MAX.

From the technical point of view, the difference in the results reflects the amount of decision variables in the optimization problems. As discussed in Section 4.4, the number of parameters to be elicited in noisy-MAX is larger than in RNM. Table 5.10 illustrates this by presenting the values of N^{sel}_{RNM}, N^{ass}_{RNM}, $N^{sel}_{noisy-MAX}$, and $\$N^{ass}_{noisy-MAX}$, see Equations 4.31–4.34, for the nodes investigated in this thesis. The values of N^{ass}_{RNM} correspond to the optimal solutions now obtained. The fact that noisy-MAX requires more parameters corresponds to having more degrees of freedom in constructing the approximate CPT $\tilde{\Theta}$ similar to the target CPT Θ. This can also be a reason why the approximations obtained with noisy-MAX are more accurate than the approximations obtained with RNM.

Table 5.10: Amounts of selectable, N^{sel}_{*}, and assignable, N^{ass}_{*}, parameters required by noisy-MAX and RNM to generate approximations of the CPTs of the child nodes.

Child node	$N^{sel}_{noisy-MAX}$	$N^{ass}_{noisy-MAX}$	N^{sel}_{RNM}	N^{ass}_{RNM}
SAO2	3	9	4	3
CO	3	12	4	3
BP	3	12	4	3
CombMoisture	3	24	4	3
AreaMoDryAir	3	24	4	3
CombClouds	3	12	4	3
CldShadeOth	4	24	5	4
InsInMt	3	12	4	3
InsChange	3	15	4	3
CapInScen	3	12	4	3
InsSclInScen	3	12	4	3
CurPropConv	3	24	4	3
CombVerMo	4	36	5	4
CompPlFcst	5	27	6	5
RiskAversion	3	20	4	3
HomeBase	3	24	4	3
DrivQuality	3	15	4	3
VehicleYear	3	12	4	3
MedCost	4	32	5	4
OtherCarCost	3	20	4	3
ThisCarDam	3	20	4	3
PropCost	3	24	4	3

As discussed in Section 5.2.1, Problem 5.3 is solved using two alternative bounds for the weights and the variance parameter in order to investigate the effect of their ranges on the modeling accuracy of RNM. Figures 5.6 and 5.7 present the MAE and MAX values related to the optimal solutions obtained when solving Problem 5.3 with the bounds in Equation 5.6. The averages and medians of the MAE and MAX values are presented in Tables 5.11 and 5.12, respectively.

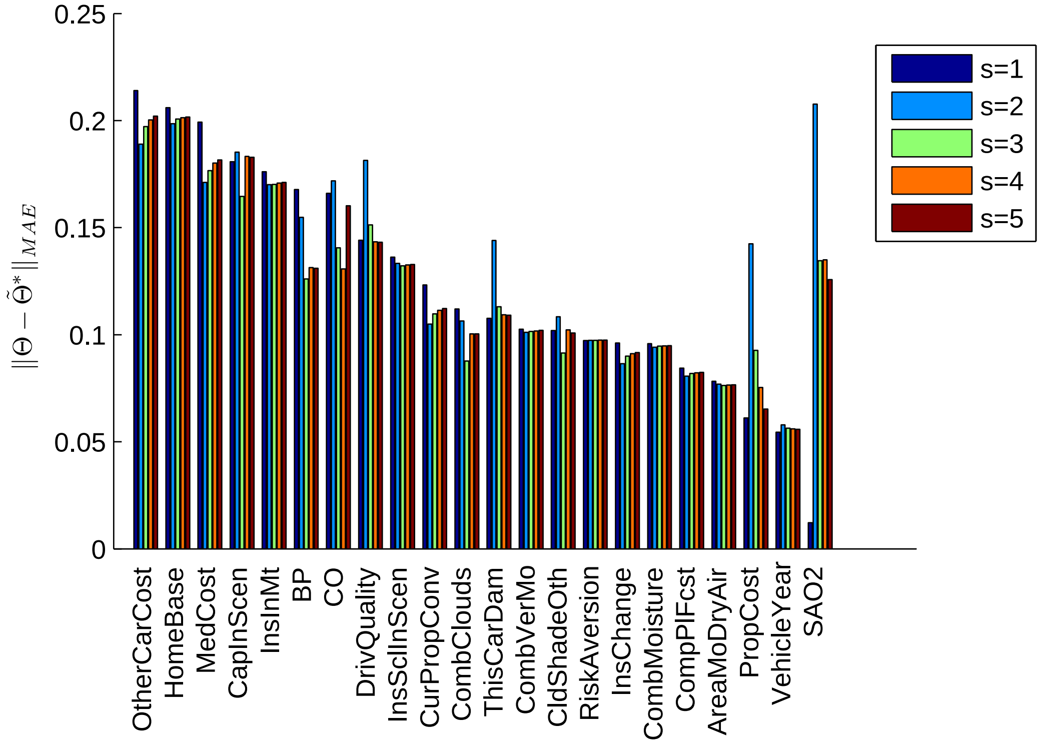


Figure 5.6: MAE values of the solutions to Problem 5.3 with the bounds in Equation 5.6.

Table 5.11: Averages, $Av(\|\Theta - \tilde{\Theta}^{*}\|_{MAE})$, and medians, $Md(\|\Theta - \tilde{\Theta}^{*}\|_{MAE})$, of the MAE values related to the optimal solutions of Problem 5.3 obtained using the bounds in Equation 5.6 and the sample point size s.

	s				
	1	2	3	4	5
$Av(\|\Theta - \tilde{\Theta}^{*}\|_{MAE})$	0.121	0.135	0.122	0.123	0.123
$Md(\|\Theta - \tilde{\Theta}^{*}\|_{MAE})$	0.106	0.140	0.111	0.110	0.111

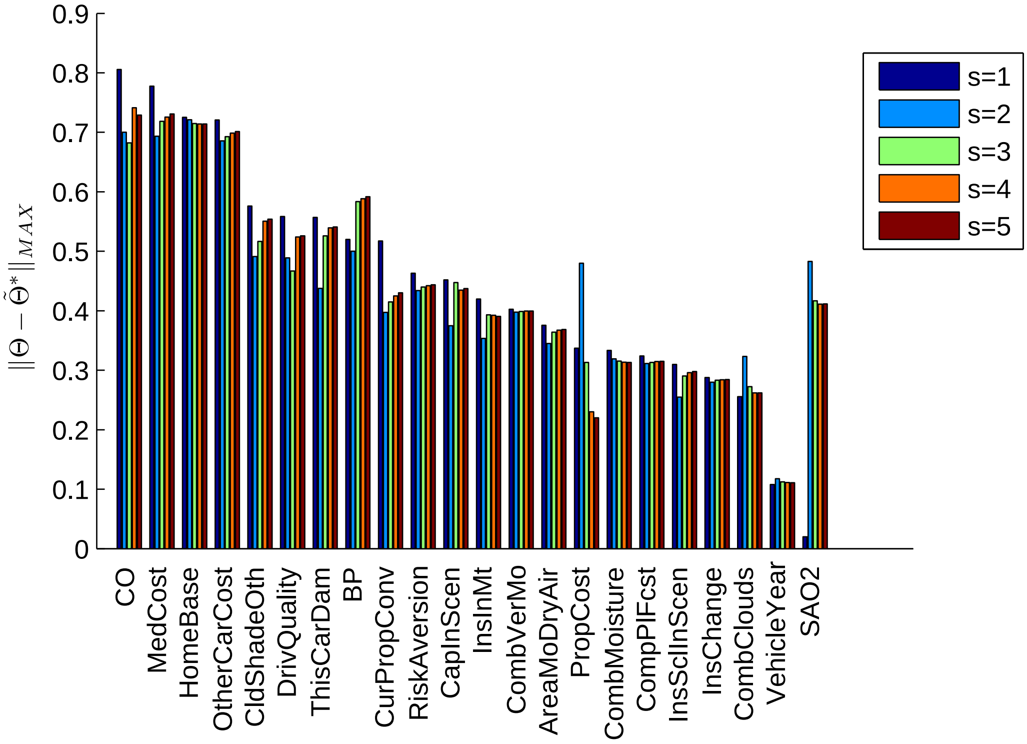


Figure 5.7: MAX values of the solutions to Problem 5.3 with the bounds in Equation 5.6.

Table 5.12: Averages, $Av(\|\Theta - \tilde{\Theta}^{*}\|_{MAX})$, and medians, $Md(\|\Theta - \tilde{\Theta}^{*}\|_{MAX})$, of the MAX values related to the optimal solutions of Problem 5.3 obtained using the bounds in Equation 5.6 and the sample point size s.

	s				
	1	2	3	4	5
$Av(\|\Theta - \tilde{\Theta}^{*}\|_{MAX})$	0.448	0.435	0.440	0.444	0.444
$Md(\|\Theta - \tilde{\Theta}^{*}\|_{MAX})$	0.436	0.416	0.416	0.418	0.421

Figures 5.4 and 5.6 indicate that using the wider ranges for $\boldsymbol{w}$ and σ^2 usually decreases the MAE values but the improvements are not drastic overall. While the improvement is roughly 0.07 at best — e.g., *MedCost* with $s = 5$ — it is only a few hundredths in most of the cases. This is also reflected by the magnitude of the differences of the corresponding elements in Tables 5.2 and 5.11.

Figures 5.5 and 5.7 also point out that the MAX values related to the optimal solutions obtained using the bounds in Equation 5.6 are in extreme cases considerably different from the ones obtained using the bounds in Equation 5.5. For example, for *SAO2* with $s = 1$ and for *PropCost* with $s = 4, 5$, the improvement in the MAX value is roughly 0.3. On the other hand, for *BP* with $s = 5$, the MAX value actually increases with roughly 0.07 when using the wider ranges. The optimal Euclidean distance obtained for *BP* with $s = 5$ using the bounds in Equation 5.5 is 1.058. When using the bounds in Equation 5.6, the corresponding distance is 1.015. Thus, this case illustrates that the Euclidean distance between two CPTs may decrease even though the difference in some of their elements increases. Despite the large change in the MAX values in some extreme cases, Figures 5.5 and 5.7 indicate that widening the ranges of $\boldsymbol{w}$ and σ^2 does not produce significant effect overall. This is also apparent from the similarity of the corresponding averages and medians of the MAX values in Tables 5.9 and 5.12.

Widening the ranges of $\boldsymbol{w}$ and σ^2 points out new cases concerning the non-uniqueness of the optimal solutions. For example, when using the bounds in Equation 5.5 and sample size $s = 2$, the unique optimal solution related to the CPT of *CapInScen* is the one presented in the first row of Table 5.13. However, when the bounds in Equation 5.6 are used, two more optimal solutions producing the same Euclidean distance are discovered. The elements of the approximate CPTs corresponding to the optimal solutions presented in Table 5.13 are the same with an accuracy of 10^{-4}. This example is another illustration of how different ways of viewing the probabilistic relationship between the ranked nodes may produce practically identical CPTs.

Table 5.13: Optimal solutions related to *CapInScen* for Problem 5.3 using bounds in Equations 5.5 and 5.6 as well as the sample point size $s = 2$.

Bounds	s	f^*	$\boldsymbol{\phi}^*$	$\boldsymbol{w}^*$	σ^{2*}	$\|\|\Theta - \tilde{\Theta}^*\|\|$	$\|\|\Theta - \tilde{\Theta}^*\|\|_{MAE}$
5.5	3	WMEAN	(0,0)	$(0.730, 0.270)$	$5 * 10^{-4}$	1.206	0.185
5.6				$(0.923, 0.077)$	10^{-5}	1.206	0.185
5.6				$(0.739, 0.261)$	$1.2 * 10^{-4}$	1.206	0.185

In many cases, the optimal solutions of Problem 5.3 obtained using the bounds in Equation 5.6 have arbitrary large elements in the optimal weights $\boldsymbol{w}^*$. In some of these cases, the large values of the weights make the solutions intractable with respect to the uniqueness. As an example, Table 5.14 presents the optimal solutions related to *InsChange* using the sample point size $s = 3$.

The three optimal solutions in Table 5.14 correspond to the three initial points of

Table 5.14: Optimal solutions related to *InsChange* for Problem 5.3 using bounds in Equation 5.6 and the sample point size $s = 2$.

s	f^*	$\boldsymbol{\phi}^*$	$\boldsymbol{w}^*$	σ^{2*}	$\Vert\Theta - \tilde{\Theta}^*\Vert$	$\Vert\Theta - \tilde{\Theta}^*\Vert_{MAE}$
3	WMIN	$(1, 1)$	(9.007*10^{15}, 1.0)	0.028	0.727	0.090
	WMIN	$(1, 1)$	(5.630*10^{14}, 1.0)	0.028	0.727	0.090
	WMIN	$(1, 1)$	(2.815*10^{14}, 1.0)	0.028	0.727	0.090

iteration presented in Equation 5.13. The elements in the corresponding approximate CPTs are the same with an accuracy of 10^{-15}. Thus, in practice, the optimal solutions in Table 5.14 correspond to the same approximation $\tilde{\Theta}^*$. Yet, the values of the weight of the second parent node vary between the optimal solutions making the solutions intractable from the point of view of uniqueness. Despite this ambiguity, in this particular case, each of the optimal solutions indicates that the Euclidean distance between the approximate and the target CPT is minimized with the following practice concerning sample points $z_{1,k}$ and $z_{2,k}$: If $z_{1,k} \leq z_{2,k}$, then $\mu_k \approx z_{1,k}$. If $z_{1,k} > z_{2,k}$, then $\mu_k = (z_{1,k} + z_{2,k})/2$. This is evident when substituting $z_{1,k}$ and $z_{2,k}$ to Equation 3.2 together with $w_1 \gg w_2$. In other words, when the first parent node is in a lower state than the second, the mode of the child node is determined by the state of the first parent node alone. When the second parent node is in a state lower than the first one, then the mode of the child node is the average of the states of the parent nodes.

The arbitrary large weights are frequently present in the optimal solutions where the use of MIXMINMAX corresponds to using either WMIN or WMAX in the way characterized by Equations 5.17 and 5.18. For example, Table 5.15 presents the optimal solutions related to *OtherCarCost*. Basically, the optimal weights in Table 5.15 are an approximation of the case $w \to \infty$ in Equation 5.18. This corresponds to the situation where the mode of the child node is determined by taking the minimum of the set of sample points $\{z_{i,k}\}_{i=1}^{n}$.

In general, the results obtained using the bounds in Equation 5.6 indicate that in some cases the modeling accuracy of RNM is maximized when the weight of a given parent node can be set arbitrary high. If the weights are elicited indirectly by means described in Section 4.2.5, one does not need to be concerned with their upper bound. However, if the weights are assessed directly with a graphical user interface using scroll bars such as that presented in Figure 5.3a, a finite upper bounds is required. Even though one might have the option to assign a larger value to a parent node than the scroll bar indicates, one might still be anchored to

Table 5.15: Optimal solutions related to *OtherCarCost* for Problem 5.3 using bounds in Equation 5.6 and the sample point size $s = 3$.

s	f^*	ϕ^*	$\boldsymbol{w}^*$	σ^{2*}	$\|\Theta - \tilde{\Theta}^*\|$	$\|\Theta - \tilde{\Theta}^*\|_{MAE}$
3	WMIN	$(0, 0)$	$(1.239*10^{16}, 6.755*10^{15})$	0.035	1.868	0.197
	WMIN	$(0, 0)$	$(1.126*10^{16}, 6.755*10^{15})$	0.035	1.868	0.197
	WMIN	$(0, 0)$	$(3.378*10^{15}, 1.126*10^{15})$	0.035	1.868	0.197
	MIXMINMAX	$(0, 0)$	$(1.902*10^{-14}, 1.000)$	0.035	1.868	0.197

use the upper bound of the scroll bar. This can lead to the underestimation of the strength of influence of some parent nodes. However, the results obtained in the experiment indicate that the inaccuracy that would be resulting, e.g., when using 5 as the upper bound for the weights, is not drastic in general.

The optimization results of Problem 5.3 indicate that there is not a value of s that would maximize the modeling accuracy of RNM in general. For example, in the case of *SAO2*, the MAE and MAX values related to $s = 1$ in Figures 5.4–5.7 are better than the ones obtained with other values of s. On the other hand, in the case of *InsChange*, the lowest MAE and MAX values in Figures 5.4–5.7 are obtained with $s = 2$. In Section 4.1, it is discussed that the use of RNM can be considered to approximate the use of a hierarchical Bayesian model characterized by Equations 4.1–4.5. Though the approximation of the model becomes more precise with increasing s, it does not mean that the model itself would provide means to accurately approximate a given CPT. The hierarchical Bayesian model is just a heuristic approach to generate a CPT representing the probabilistic relationship between the parent nodes and the child node. Similarly, RNM with any value of s can be considered to correspond to some kind of heuristics to generate CPTs. In various cases, different one of these heuristics provides the best means to represent the probabilistic relationship of the parent nodes and the child node. Thus, there is no necessity to use as large sample point size as possible. One may well use RNM with small values of s, like $s = 1$ or $s = 2$, in order to make the computations lighter and faster to carry out.

Results of Problem 5.4

As discussed in Section 5.2.1, the motivation of studying Problem 5.4 is to understand better the importance of partitioned expressions in the effective use of RNM. Hence, the analysis of the optimal solutions of Problem 5.4 is focused on comparing the MAE and MAX values to those describing the optimal solutions of Problem 5.3. The effect of widening the bounds of the weights and variance parameter from those in Equation 5.5 to those in Equation 5.6 is also examined.

Figure 5.8 presents the MAE values of the optimal approximations $\tilde{\Theta}^*$ obtained when solving Problem 5.4 with sample point sizes $s = 1, ..., 5$ and using the bounds in Equation 5.5. The averages and medians of the MAE values are presented in Table 5.16.

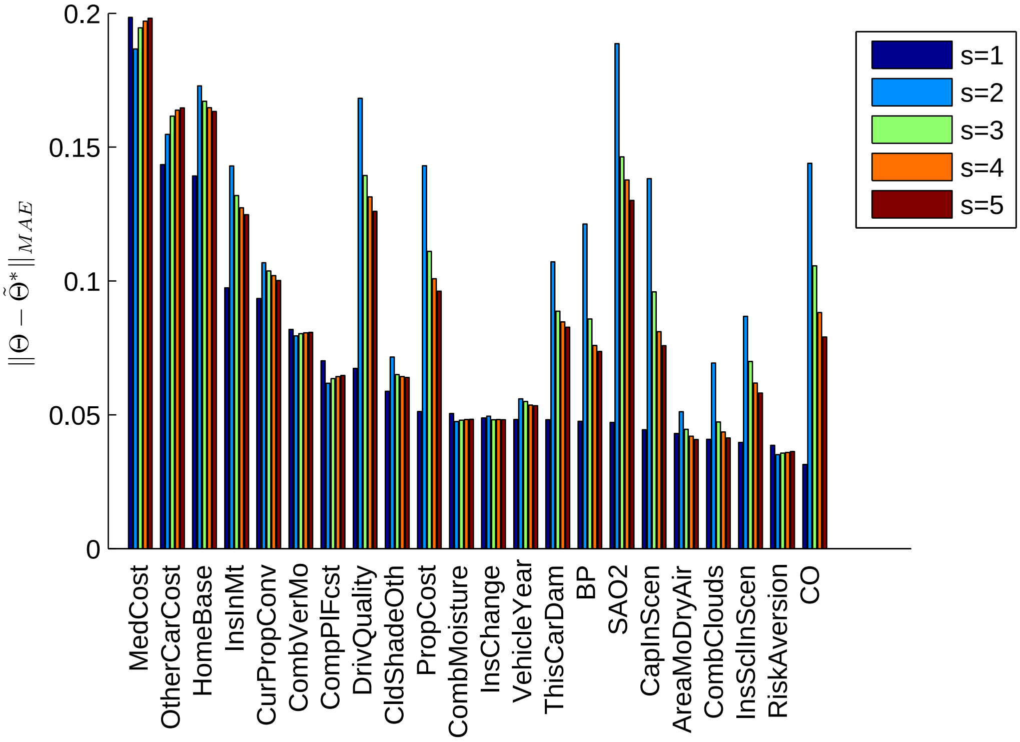


Figure 5.8: MAE values of the solutions to Problem 5.4 with the bounds in Equation 5.5.

Figures 5.4 and 5.8 indicate that in many cases, the use of partitioned expressions clearly improves the accuracy of the approximate CPT. For example, the MAE value related to *CO* with $s = 1$ is reduced from over 0.15 to less than 0.05. Using the numbers in Tables 5.2 and 5.16, it can be calculated that the averages and medians of the MAE values reduce by roughly 0.03–0.06 depending on s. These correspond to relative improvements between 20–50 %.

Figures 5.4 and 5.8 as well as Tables 5.2 and 5.16 imply also that the improvements are usually the largest when $s = 1$ and the smallest when $s = 2$. Especially

Table 5.16: Averages, $Av(\|\Theta - \tilde{\Theta}^*\|_{MAE})$, and medians, $Md(\|\Theta - \tilde{\Theta}^*\|_{MAE})$, of the MAE values related to the optimal solutions of Problem 5.4 obtained using the bounds in Equation 5.5 and the sample point size s.

	s				
	1	2	3	4	5
$Av(\|\Theta - \tilde{\Theta}^*\|_{MAE})$	0.077	0.116	0.102	0.098	0.096
$Md(\|\Theta - \tilde{\Theta}^*\|_{MAE})$	0.056	0.109	0.087	0.081	0.077

in Figure 5.8, the MAE values obtained with $s = 1$ are for many nodes distinctively smaller than the ones obtained with other values of s. Examples include *DrivQuality* and *CO* among others. Common to these cases is also that the worst MAE values are obtained with $s = 2$. The reason for this phenomenon might be related to the following feature of RNM. When $s = 1$, the sample points used are the centre points of the state intervals of the parent nodes. As opposed to this, when $s = 2$, the sample points are the lower and upper bounds of the state intervals. These are the two extreme cases of the use of the sample points — all cases with $s \geq 3$ can be regarded as some kind of compromises between these two. When $s = 1$, only one mean parameter μ_1 is generated for a given combination of the states of the parent nodes $(X_1 = x_1, ..., X_n = x_n)$. The conditional probability distribution of the child node $P(X_{n+1}|X_1 = x_1, ..., X_n = x_n)$ is obtained by integrating a single doubly truncated normal distribution $TNormal(\mu_1, \sigma^2, 0, 1)$ over the state intervals of the child node. When $s = 2$, the different mean parameters μ_k are calculated by using only the most extreme points related to the state intervals of the parent nodes. This means that the resulting values of the mean parameters related to the given combination of the states of the parent nodes $(X_1 = x_1, ..., X_n = x_n)$ are the very extremes that can be obtained. The integration of the doubly truncated normal distributions may produce different conditional probability distributions $P(X_{n+1}|X_1 = x_1, ..., X_n = x_n; \mu_k)$. As the final probability distribution $P(X_{n+1}|X_1 = x_1, ..., X_n = x_n)$ is taken as the average of $P(X_{n+1}|X_1 = x_1, ..., X_n = x_n; \mu_k)$, it may behave differently from the one obtained by integrating a single doubly truncated normal distribution. This could explain why the results obtained with $s = 1$ and $s = 2$ are usually the most different among the ones obtained with different values of s. The reason why the results with $s = 1$ are in many cases so much better than the ones with $s = 2$ would then indicate that in these particular cases, the distributions in the CPTs correspond better to the approximations obtained when only a single doubly truncated normal distributions is used. As stated above, this is just a possible explanation of the phenomenon. The issue should be studied more to explain it better. However, this is not in the focus of the thesis and is therefore now omitted.

Figure 5.9 displays the MAX values related to the optimal solutions of Problem 5.4 when using the bounds in Equation 5.5 and sample point sizes $s = 1, ..., 5$. The averages and medians are presented in Table 5.17. By comparing Figures 5.5 and 5.9, one can observe that the MAX values improve when using partitioned expressions. For example, in the case of *CO*, the improvement is about 0.3–0.5. The general improvement in the MAX values is also pointed out by Tables 5.9 and 5.17. The improvements in the averages and medians of the MAX values vary roughly between 0.1 and 0.17 depending on s. The relative improvements are now varying between roughly 18–35 %. Though the relative improvements are smaller, the use of partitioned expressions decreased the MAX values more than the MAE values. This encourages their use as means to decrease the extreme inaccuracies in the CPT. However, Figures 5.8 and 5.9 indicate that in many cases the MAX values remain larger than the MAE values. If the accuracy is considered too low, one should consider partitioning the CPT even more or possibly correct the critical inaccuracies manually.

Table 5.17: Averages, $Av(\|\Theta - \tilde{\Theta}^*\|_{MAX})$, and medians, $Md(\|\Theta - \tilde{\Theta}^*\|_{MAX})$, of the MAX values related to the optimal solutions of Problem 5.4 obtained using the bounds in Equation 5.5 and the sample point size s.

	s				
	1	2	3	4	5
$Av(\|\Theta - \tilde{\Theta}^*\|_{MAX})$	0.350	0.370	0.357	0.359	0.359
$Md(\|\Theta - \tilde{\Theta}^*\|_{MAX})$	0.309	0.332	0.333	0.333	0.333

Figures 5.10 and 5.11 display the MAE and MAX values related to the optimal solutions obtained when solving Problem 5.4 using the bounds in Equation 5.6 and sample point sizes $s = 1, ..., 5$. The averages and medians of the MAE and MAX values are presented in Tables 5.18 and 5.19, respectively.

Widening the ranges of the weights and the variance parameter has similar effects on the optimal solutions when using partitioned expressions as when a single weight expression is used. For example, Figures 5.8 and 5.10 as well as Tables 5.16 and 5.18 indicate that the MAE values obtained with the wider ranges are in general only a few hundredths smaller than those obtained with the narrower ranges. Compared to the non-partitioned case, the effect of widening the ranges of $\boldsymbol{w}$ and σ^2 is now a bit larger. Figures 5.9 and 5.11 as well as Tables 5.17 and 5.19

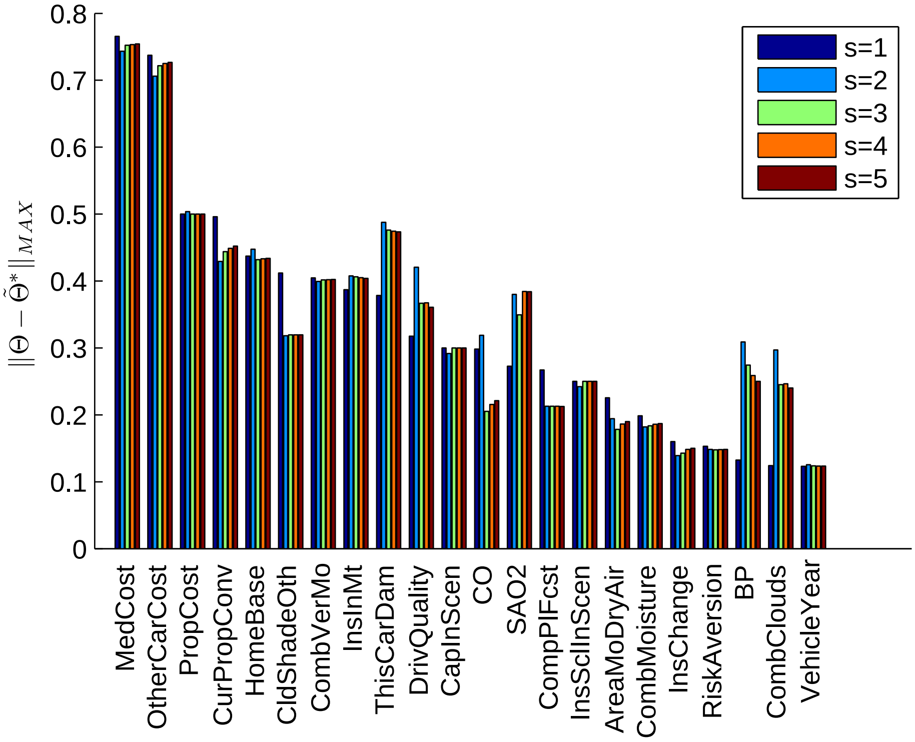


Figure 5.9: MAX values of the solutions to Problem 5.4 with the bounds in Equation 5.5.

Table 5.18: Averages, $Av(\|\Theta - \tilde{\Theta}^*\|_{MAE})$, and medians, $Md(\|\Theta - \tilde{\Theta}^*\|_{MAE})$, of the MAE values related to the optimal solutions of Problem 5.4 obtained using the bounds in Equation 5.6 and the sample point size s.

	s				
	1	2	3	4	5
$Av(\|\Theta - \tilde{\Theta}^*\|_{MAE})$	0.064	0.099	0.083	0.078	0.075
$Md(\|\Theta - \tilde{\Theta}^*\|_{MAE})$	0.045	0.088	0.074	0.069	0.065

imply that the MAX values improve on average by roughly 0.03–0.04.

Overall, the results of the experiments indicate that the use of partitioned expressions can considerably enhance the modeling accuracy of RNM. This is expected because the use of partitioned expressions provides more parameters to modify the approximate CPT to be more similar with the original. Naturally, the downside of the use of partitioned expressions is that it increases the elicitation effort. Table 5.20 presents the values of N^{sel}_{RNM} and N^{ass}_{RNM} related to the optimal solutions

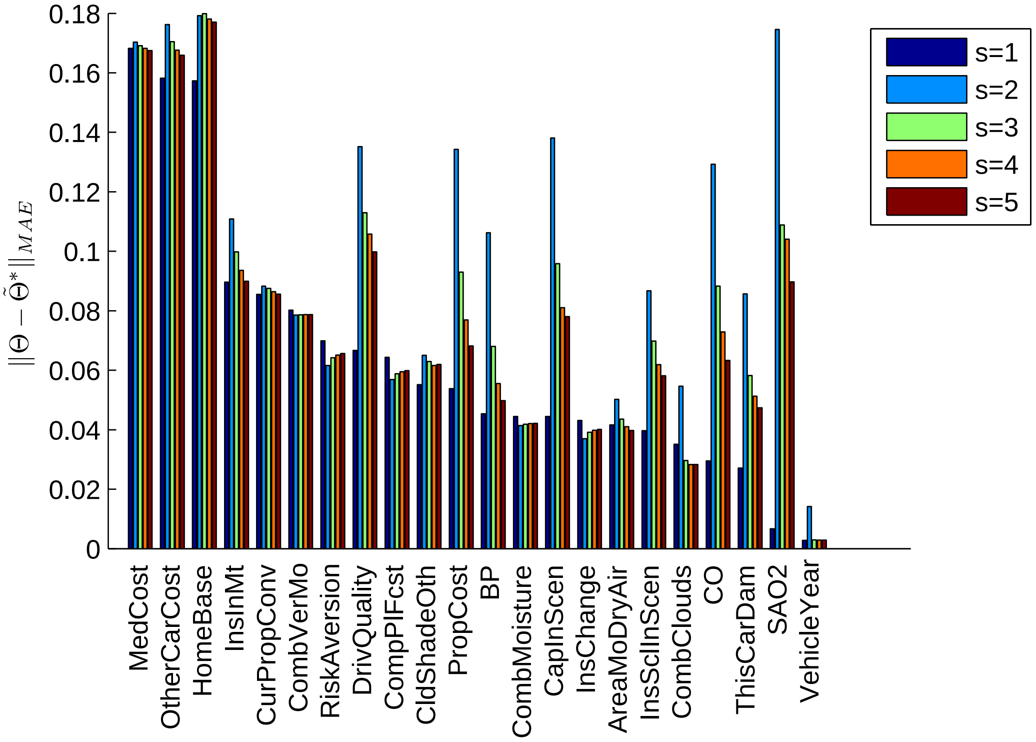


Figure 5.10: MAE values of the solutions to Problem 5.3 with the bounds in Equation 5.6.

Table 5.19: Averages, $Av(\|\Theta - \tilde{\Theta}^{*}\|_{MAX})$, and medians, $Md(\|\Theta - \tilde{\Theta}^{*}\|_{MAX})$, of the MAX values related to the optimal solutions of Problem 5.4 obtained using the bounds in Equation 5.6 and the sample point size s.

	s				
	1	2	3	4	5
$Av(\|\Theta - \tilde{\Theta}^{*}\|_{MAX})$	0.312	0.333	0.306	0.303	0.305
$Md(\|\Theta - \tilde{\Theta}^{*}\|_{MAX})$	0.299	0.321	0.292	0.275	0.275

presented in Figures 5.10 and 5.11 for $s = 1$.

Tables 5.10 and 5.20 indicate that the sum of the selectable and assignable parameters still remains lower for RNM than for noisy-MAX in most of the cases. Thus, the difference between the numbers of parameters may still be used as a rationale why the optimization results for RNM after the use of partitioned expressions are not as good as the ones obtained with noisy-MAX in [58].

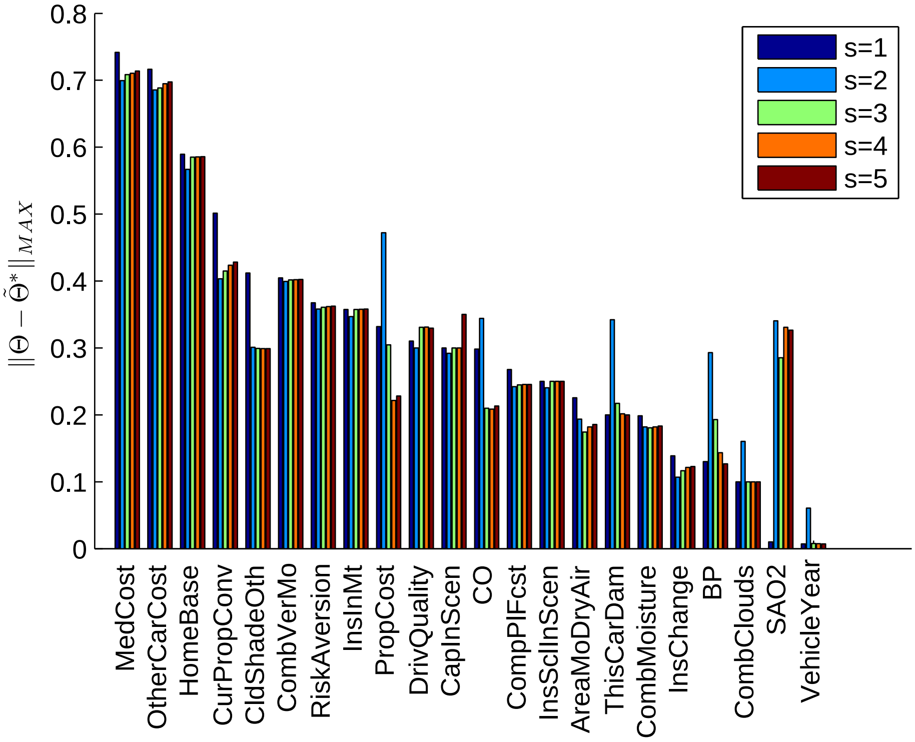


Figure 5.11: MAX values of the solutions to Problem 5.3 with the bounds in Equation 5.6.

5.2.3 Summary

The results obtained for RNM in the experiment are discovered to be generally poorer than those obtained for noisy-MAX in [58]. While the results of these experiments are not fully comparable, they indicate that the probabilistic relationships that RNM is designed to describe are not as common in the applications of BNs as those described by noisy-MAX. The poorer modeling accuracy of RNM may be explained by the fact that it uses smaller amount of parameters in order to calculate the CPT than noisy-MAX.

When using RNM without partitioned expressions, the medians of the MAE and MAX values obtained in the experiment are at best roughly 0.1 and 0.4, respectively. When partitioning the CPTs according to the states of a single parent node, the corresponding medians are roughly 0.05 and 0.3. Some of the CPTs are approximated by RNM considerably better than the median values. This indicates that the method may provide means to construct CPTs of satisfying accuracy.

Table 5.20: Amounts of selectable, N_{RMS}^{sel}, and assignable, N_{RMS}^{ass}, parameters related to the optimal solutions of Problem 5.4 using bounds in Equation 5.6.

Child node	N_{RNM}^{sel}	N_{RNM}^{ass}
SAO2	6	9
CO	6	9
BP	6	9
CombMoisture	7	12
AreaMoDryAir	7	12
CombClouds	6	9
CldShadeOth	7	12
InsInMt	6	9
InsChange	7	12
CapInScen	6	9
InsSclInScen	6	9
CurPropConv	7	12
CombVerMo	8	16
CompPlFcst	9	20
RiskAversion	7	12
HomeBase	7	12
DrivQuality	6	9
VehicleYear	7	12
MedCost	8	12
OtherCarCost	7	12
ThisCarDam	7	12
PropCost	7	12

The modeling accuracy of RNM is discovered to be insensitive to widening the ranges of the weights and the variance parameter from the default ranges used in AgenaRisk software [38]. The use of partitioned expressions improves the modeling accuracy more. However, the use of partitioned expressions also increases the amount of parameters to be elicited. Thus, the effective use of RNM includes making a suitable trade-off between the accuracy of the CPT and the elicitation effort.

To summarize, the small amount of parameters required by RNM indicates that it provides means to construct CPTs with small elicitation effort. On the other hand, the resulting CPTs may not represent the probabilistic relationships of the nodes accurately enough. These features suggest to use RNM as means to quickly construct rough CPTs that are used as the starting points of iterative elicitation process based on, e.g., the sensitivity analysis of the BN which is discussed in Section 3.6. Based on the results of the sensitivity analysis, the CPT could be refined by adjusting the parameters of RNM, using partitioned expressions , or, e.g., altering the critical elements in the CPT directly. Overall, the degree of refinement would be determined by the desired accuracy of the CPT.

Chapter 6

Conclusion

This thesis studied the ranked nodes method (RNM) [25] which is used to construct conditional probability tables (CPTs) to Bayesian networks (BNs) consisting of ranked nodes on the basis of expert elicitation. The thesis presented RNM with a precision that enables the implementation of the method in practise. Based on this presentation, RNM was studied from both modeling and computational aspects in ways that enhance the understanding of the functioning of the method, extend its practical usability, and clarify its possibilities and limitations.

When studying RNM from the modeling point of view, the thesis discussed the interpretation of the normalized scale $[0, 1]$ with respect to the discrete states of the ranked nodes. In the discussion, it was shown that the use of the method approximates the use of a hierarchical Bayesian model of continuous random variables. This finding helps to understand the method and it was also utilized when explaining results of an experimental study concerning the modeling accuracy of RNM. In addition, exact interpretations were derived for weights used in RNM. Based on these interpretations, the weights can be elicited in a more transparent and consistent way compared to their direct assessment. However, the abstract nature of the normalized scale might make the elicitation cognitively challenging in practice. The matter should be studied further empirically to gain more insight into it.

The thesis examined through an illustrative example the application of RNM to ranked nodes whose underlying continuous quantity is expressed as a ratio or interval scale. Through the example, it was discussed how the scales should be discretized compatibly to the functioning of RNM. In addition, the interpretation of the normalized scale $[0, 1]$ was revised. Moreover, the utilization of the interpretations of the weights in their elicitation with explicit and adjustable questions concerning the scales of the nodes was presented. The ideas introduced enable the use of RNM in various practical applications in a more transparent and user friendly way.

The computational complexity of RNM was examined in the thesis by measuring the calculation times of CPTs with a self-made implementation of the method. The results indicate that RNM can be implemented so that in most common settings the CPTs can be calculated in the magnitude of a second. Thus, the practical use of RNM is not hindered by long calculation times of the CPTs.

The modeling accuracy of RNM and its dependence on various computational parameters were examined by approximating CPTs included in existing real-life benchmark BNs. RNM is found to provide accurate approximations in some cases. Moreover, for some of the approximate CPTs, the use of partitioned expressions is discovered to enhance the accuracy of the approximations drastically. However, in general, the results are found to be poorer than the ones obtained in a similar study concerning noisy-MAX [58]. This can be understood to indicate the relative rarity of the probabilistic relationships compatible with the assumptions of RNM in the applications of BNs. The poorer modeling accuracy of RNM can also be due to the smaller amount of parameters it uses. Overall, the results of the empirical studies indicate that RNM offers means to promptly construct rough approximations of CPTs that can be refined based on, e.g., the sensitivity analysis of the BN. This means varying one or more conditional probabilities in the BN and investigating the changes that it introduces to a probability of interest [10]. This type of analysis identifies which conditional probabilities need to be defined more accurately than others.

The results and discoveries of this thesis motivate further research on several topics. The empirical study concerning the modeling accuracy of RNM included 22 CPTs that were approximated. To obtain more reliable results on the commonness of the probabilistic relationships compatible with the assumptions of RNM, a larger sample of CPTs should be investigated. The modeling accuracy of RNM could also be studied empirically from the viewpoint of the elicitation of parameters. For example, similarly to the idea in [59], human test subjects would be allowed to make trials with an artificial model obeying a CPT constructed with RNM. With enough observations acquired with the trials, the test subjects could be considered to have become experts concerning the behaviour of the model. Then, they would be asked to construct a CPT representing their observations with different means, e.g., using direct elicitation and assessing parameters of RNM. The results of the experiment would indicate how intuitive it is to assess the parameters of RNM. The same kind of set-up could also be used to verify whether the elicitation of weights based on the interpretations derived in this thesis is a more useful way to elicit the weights than their direct assessment.

The ideas presented in this thesis for handling ranked nodes with ratio or interval scales should be further studied and systematized. The systematization could refer to guidelines or instructions concerning the discretization of the scales. In addition, it could deal with incompatibilities between the statements of the expert and the implications of the RNM representation. To this end, it would be beneficial to carry out empirical experiments in realistic settings with real experts. The systematization could also be elaborated with similar empirical experiments as described above. For example, human test subjects would be allowed to make trials with an artificial model of continuous random variables that are defined as ranked nodes using piecewise linear functions presented in the thesis. Then, it would be investigated how well the test subjects can discretize the random variables into ranked nodes and assess the weights by answering questions about the mode of the child node in various scenarios.

In the study concerning the modeling accuracy of RNM, it was also discovered that the best approximate CPTs were often obtained using a single point on the normalized scale to represent the state of a parent node. The generality of this phenomenon should be verified by approximating larger amount of CPTs. If the results would turn out to be generalizable, it could lead to establishing an alternative version of RNM where the states of nodes are not identified with subintervals on the normalized scale but as single points. Such a version of RNM might broaden the utilization possibilities of the method to nodes whose discrete states cannot be deemed to represent any underlying continuous quantity.

Bibliography

[1] B. Abramson, J. Brown, W. Edwards, A. Murphy, and R.L. Winkler. Hailfinder: A Bayesian System for Forecasting Severe Weather. *International Journal of Forecasting*, 12(1):57–71, 1996.

[2] Hugin Expert A/S. Hugin Educational Software, Version 6.8. `http://www.hugin.com`, December 2010.

[3] P. Baraldi, M. Conti, M. Librizzi, E. Zio, L. Podofillini, and V.N. Dang. A Bayesian Network Model for Dependence Assessement in Human Reliability Analysis. In *Proceedings of the Annual European Safety and Reliability Conference, ESREL 2009*, pages 223–230. Prague, Czech Republic, September 7–10, 2009.

[4] I. Beinlich, H.J. Suermondt, R. Chavez, G. Cooper, et al. The ALARM Monitoring System: A Case Study With Two Probabilistic Inference Techniques For Belief Networks. In *Proceedings of the 2nd European Conference on Artificial Intelligence in Medicine*, pages 247–256. London, England, August 29–31, 1989.

[5] C. Bielza, M. Gómez, and P.P. Shenoy. Modeling Challenges with Influence Diagrams: Constructing Probability and Utility Models. *Decision Support Systems*, 49(4):354–364, 2010.

[6] J. Binder, D. Koller, S. Russell, and K. Kanazawa. Adaptive Probabilistic Networks with Hidden Variables. *Machine Learning*, 29(2):213–244, 1997.

[7] A. Constantinou, N. E. Fenton, and M. Neil. Pi-Football: A Bayesian Network Model for Forecasting Association Football Match Outcomes. *Knowledge-Based Systems*, 36:322–339, 2012.

[8] G.F. Cooper. The Computational Complexity of Probabilistic Inference Using Bayesian Belief Networks. *Artificial Intelligence*, 42(2-3):393–405, 1990.

[9] Norsys Software Corp. Netica Application, Version 4.16. `http://www.norsys.com/netica.html`, May 2010.

[10] V.M.H. Coupé and L.C. Van Der Gaag. Properties of Sensitivity Analysis of Bayesian Belief Networks. *Annals of Mathematics and Artificial Intelligence*, 36(4):323–356, 2002.

[11] V.M.H. Coupé, L.C. Van Der Gaag, and J.D.F. Habbema. Sensitivity Analysis: An Aid for Belief-Network Quantification. *The Knowledge Engineering Review*, 15(3):215–232, 2000.

[12] A.R. Daneshkhah. Psychological Aspects Influencing Elicitation of Subjective Probability. *Research Report, University of Sheffield*, 2004.

[13] B. Das. Generating Conditional Probabilities for Bayesian Networks: Easing the Knowledge Acquisition Problem. Technical report, Command and Control Division, DSTO, Edinburgh, SA 5111, Australia, 2004.

[14] University of Pittsburgh Decision Systems Laboratory (DSL), School of Information Sciences. GeNIe Software, Version 2.0. `http://genie.sis.pitt.edu`, June 2011.

[15] E. Denney, G. Pai, and I. Habli. Towards Measurement of Confidence in Safety Cases. In *Proceedings of the 5th International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 380–383. Banff, Canada, September 22–23, 2011.

[16] M. Diehl and Y.Y. Haimes. Influence Diagrams with Multiple Objectives and Tradeoff Analysis. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 34(3):293–304, 2004.

[17] F.J. Diez. Parameter Adjustment in Bayes Networks. The Generalized Noisy Or-Gate. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 99–105. Washington, D.C., USA, July 9-11, 1993.

[18] F.J. Dıez and M.J. Druzdzel. Canonical Probabilistic Models for Knowledge Engineering. *Unpublished manuscript, available at http://www.ia.uned.es/ fjdiez/papers/canonical.html*, 2008.

[19] M.J. Druzdel and L.C. van der Gaag. Building Probabilistic Networks:" Where Do the Numbers Come From?". *IEEE Transactions on knowledge and data engineering*, 12(4):481–486, 2000.

[20] H.-S. Eom, G.-Y. Park, S.-C. Jang, H.S. Son, and H.G. Kang. V&V-Based Remaining Fault Estimation Model for Safety–Critical Software of a Nuclear Power Plant. *Annals of Nuclear Energy*, 51:38–49, 2013.

[21] L. Falzon. Using Bayesian Network Analysis to Support Centre of Gravity Analysis in Military Planning. *European Journal of Operational Research*, 170(2):629–643, 2006.

[22] N. Fenton, P. Krause, and M. Neil. Software Measurement: Uncertainty and Causal Modeling. *IEEE Software*, 19(4):116–122, 2002.

[23] N. Fenton, M. Neil, and D. Marquez. Using Bayesian Networks to Predict Software Defects and Reliability. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 222(4):701–712, 2008.

[24] N. Fenton, M. Neil, W. Marsh, P. Hearty, Ł. Radliński, and P. Krause. On the Effectiveness of Early Life Cycle Defect Prediction with Bayesian Nets. *Empirical Software Engineering*, 13(5):499–537, 2008.

[25] N.E. Fenton, M. Neil, and J.G. Caballero. Using Ranked Nodes to Model Qualitative Judgments in Bayesian Networks. *IEEE Transactions on Knowledge and Data Engineering*, 19(10):1420–1432, 2007.

[26] M. Fineman. *Improved Risk Analysis for Large Projects: Bayesian Networks Approach*. PhD thesis, Queen Mary University, London, 2010.

[27] A. Gelman, J.B. Carlin, H.S. Stern, and Rubin D.B. *Bayesian Data Analysis, Second Edition*. Chapman & Hall/CRC, 2004.

[28] P. Hearty, N. Fenton, D. Marquez, and M. Neil. Predicting Project Velocity in XP Using a Learning Dynamic Bayesian Network Model. *IEEE Transactions on Software Engineering*, 35(1):124–137, 2009.

[29] P.S. Hearty. *Modelling Agile Software Processes Using Bayesian Networks*. PhD thesis, Queen Mary University, London, 2008.

[30] D. Heckerman and J. S. Breese. Causal Independence for Probability Assessment and Inference Using Bayesian Networks. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 26(6):826–831, 1996.

[31] D. Heckerman and J.S. Breese. A New Look at Causal Independence. In *Proceedings of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 286–292. Seattle, Washington, USA, July 29–31, 1994.

[32] D. Heckerman, J.S. Breese, and K. Rommelse. Decision-Theoretic Troubleshooting. *Communications of the ACM*, 38(3):49–57, 1995.

[33] D.E. Heckerman, E.J. Horvitz, and B.N. Nathwani. Toward Normative Expert Systems: Part I. The Pathfinder Project. *Methods of Information in Medicine*, 31(2):90–105, 1992.

[34] A.W. Knapp. *Basic Real Analysis*, volume 10. Springer, 2005.

[35] V. Kumar, A. Holzkaemper, B. Surridge, P.I. Rockett, M. Niranjan, and D.N. Lerner. Bayesian Challenges in Integrated Catchment Modelling. In *Proceedings of the iEMSs Fourth Biennial Meeting: International Congress on Environmental Modelling and Software (iEMSs 2008)*, pages 978–984. Barcelona, Catalonia, Spain, July 7–8, 2008.

[36] S.L. Lauritzen and D.J. Spiegelhalter. Local Computations with Probabilities on Graphical Structures and Their Application to Expert Systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 50(2):157–224, 1988.

[37] L. Lee. On the Effectiveness of the Skew Divergence for Statistical Language Analysis. In *Proceedings of the 8th International Workshop on Artificial Intelligence and Statistics (AISTATS)*, pages 65–72. Key West, Florida, USA, January 4–7, 2001.

[38] Agena Ltd. AgenaRisk Software, Version 5.0. `http://www.agenarisk.com`, June 2011.

[39] Inc. MathWorks. Matlab Software, Version R2010b. `http://www.mathworks.se/products/matlab/`, August 2010.

[40] M.G. Morgan and M. Henrion. *Uncertainty: A Guide to Dealing with Uncertainty in Quantitative Risk and Policy Analysis*. Cambridge University Press, 1992.

[41] R.E. Neapolitan. *Learning Bayesian Networks*. Pearson Prentice Hall Upper Saddle River, 2004.

[42] K.G. Olesen, U. Kjaerulff, F. Jensen, F.V. Jensen, B. Falck, S. Andreassen, and S.K. Andersen. A Munin Network for the Median Nerve — A Case Study on Loops. *Applied Artificial Intelligence an International Journal*, 3(2–3):385–403, 1989.

[43] A. Onisko. *Probabilistic Causal Models in Medicine: Applications to Diagnosis of Liver Disorders*. PhD thesis, Institute of Biocybernetics and Biomedical Engineering, Polish Academy of Science, 2003.

[44] J. Pearl. Fusion, Propagation, and Structuring in Belief Networks. *Artificial Intelligence*, 29(3):241–288, 1986.

[45] J. Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[46] D.N. Plüss, A. Groso, and T. Meyer. Expert Judgements in Risk Analysis: a Strategy to Overcome Uncertainties. *Chemical Engineering Transactions*, 31:307–312, 2013.

[47] J. Poropudas and K. Virtanen. Analyzing Air Combat Simulation Results with Dynamic Bayesian Networks. In *Proceedings of the 2007 Winter Simulation Conference*, pages 1370–1377. Washington, D.C., USA, December 9–12, 2007.

[48] C. Preston and A. Colman. Optimal Number of Response Categories in Rating Scales: Reliability, Validity, Discriminating Power, and Respondent Preferences. *Acta Psychologica*, 104(1):1–15, 2000.

[49] S. Renooij. Probability Elicitation for Belief Networks: Issues to Consider. *The Knowledge Engineering Review*, 16(3):255–269, 2001.

[50] W. Rudin. *Principles of Mathematical Analysis*, volume 3. McGraw-Hill New York, 1964.

[51] S.J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2003.

[52] S. Srinivas. A Generalization of the Noisy-Or Model. In *Proceedings of the 9th Conference on Uncertainty in Artificial Intelligence*, pages 208–215. Washington, D.C., USA, July 9–11, 1993.

[53] R.S. Strichartz. *The Way of Analysis*. Jones & Bartlett Learning, 2000.

[54] L.C. Van der Gaag, S. Renooij, C.L.M. Witteman, B.M.P. Aleman, and B.G. Taal. Probabilities for a Probabilistic Network: A Case-Study in Oesophageal Carcinoma. *Artificial Intelligence in Medicine*, 25(2):123–148, 2002.

[55] R.A. Van Engelen. Approximating Bayesian Belief Networks by Arc Removal. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(8):916–920, 1997.

[56] S. Wagner, M. Ruhe, and A.G. Siemens. Using a Bayesian Network in the ProdFLOWTM Approach. In *Proceedings of the 2nd International Workshop on Software Productivity Analysis and Cost Estimation (SPACE 2008)*. Beijing, China, December 1, 2008.

[57] A. Zagorecki. *Local Probability Distributions in Bayesian Networks: Knowledge Elicitation and Interference*. PhD thesis, University of Pittsburgh, 2010.

[58] A. Zagorecki and M. Druzdzel. Knowledge Engineering for Bayesian Networks: How Common Are Noisy-MAX Distributions in Practice? In *Proceedings of the 17th European Conference on Artificial Intelligence*, pages 482–486. Riva del Garda, Italy, August 29–September 1, 2006.

[59] A. Zagorecki and M. Druzdzel. An Empirical Study of Probability Elicitation Under Noisy-OR Assumption. In *Proceedings of the 17th International Florida Artificial Intelligence Research Society Conference (FLAIRS 2004)*, pages 880–885. Miami Beach, Florida, USA, May 17–19, 2004.

Appendices

Appendix A

Distributions of Nodes in Figure 2.1

Table A-1: Skills.

States	High	Medium	Low
Probabilities	0.25	0.60	0.15

Table A-2: Spryness.

States	High	Medium	Low
Probabilities	0.25	0.50	0.25

Table A-3: Disturbance Level.

States	High	Medium	Low
Probabilities	0.167	0.667	0.167

Table A-4: Productivity.

Parents			States and Probabilities		
Skills	Spryness	Disturbance level	High	Medium	Low
High	High	Low	0.909	0.091	0.000
		Medium	0.699	0.301	0.000
		High	0.389	0.606	0.007
	Medium	Low	0.786	0.214	0.000
		Medium	0.493	0.504	0.003
		High	0.203	0.766	0.031
	Low	Low	0.599	0.399	0.001
		Medium	0.288	0.696	0.016
		High	0.084	0.821	0.095
Medium	High	Low	0.506	0.491	0.003
		Medium	0.213	0.759	0.029
		High	0.052	0.805	0.142
	Medium	Low	0.300	0.686	0.01
		Medium	0.089	0.821	0.089
		High	0.014	0.686	0.300
	Low	Low	0.142	0.805	0.052
		Medium	0.029	0.759	0.213
		High	0.003	0.491	0.506
Low	High	Low	0.095	0.821	0.084
		Medium	0.016	0.696	0.288
		High	0.001	0.399	0.599
	Medium	Low	0.031	0.766	0.203
		Medium	0.003	0.504	0.493
		High	0.000	0.214	0.786
	Low	Low	0.007	0.606	0.387
		Medium	0.000	0.301	0.699
		High	0.000	0.091	0.909

Appendix B

Proof of Equation 4.5

The claim of Equation 4.5 is presented below as Theorem 1. The proof of Theorem 1 is based on a series of aiding definitions and lemmas that are presented first. The starting point of the upcoming analysis is the acknowledgement that $\mathbb{R}^n$ and the Euclidean norm $\|\cdot\| : \mathbb{R}^n \to \mathbb{R}$ defined by

$$\|\boldsymbol{u}\| = \|(u_1, ..., u_n)\| = \sqrt{\sum_{i=1}^{n} u_i^2},$$

form a metric space, see [53].

First, two different but equivalent definitions concerning the continuity of functions operating in Cartesian spaces are given. The equivalent definitions in Definition 1 adapt the ones presented in [53].

Definition 1. *(a) The function $F : \mathbb{R}^d \to \mathbb{R}^n$ is continuous if for any point $\boldsymbol{u}_0 \in \mathbb{R}^d$ it applies that*

$$\forall\, \epsilon > 0\, \exists\, \delta > 0\ : \boldsymbol{u} \in \mathbb{R}^d \wedge \|\boldsymbol{u} - \boldsymbol{u}_0\| < \delta \Rightarrow \|F(\boldsymbol{u}) - F(\boldsymbol{u}_0)\| < \epsilon.$$

(b) Equivalently, F is continuous if the following holds for any $\boldsymbol{u}_0 \in \mathbb{R}^d$ and a sequence $\{\boldsymbol{u}_i\}_{i=1}^{\infty} \subset \mathbb{R}^d$:

$$\lim_{i\to\infty} \|\boldsymbol{u}_i - \boldsymbol{u}_0\| = 0 \Rightarrow \lim_{i\to\infty} \|F(\boldsymbol{u}_i) - F(\boldsymbol{u}_0)\| = 0.$$

Now, the following notation for the convergence of a sequence $\{\boldsymbol{y}_i\}$ is adapted:

$$\boldsymbol{y}_i \xrightarrow{i\to\infty} \boldsymbol{y} \Leftrightarrow \lim_{i\to\infty} \|\boldsymbol{y}_i - \boldsymbol{y}\| = 0.$$

Lemma 1. *The weight expressions WMEAN, WMIN, WMAX, and MIXMINMAX — defined in Equations 3.1-3.4, respectively — are continuous with respect to the sample points* $\{z_{i,k}\}$.

Proof. For the brevity of notation, let the combination of sample points $\{z_{1,k}, ..., z_{n,k}\}$ be marked with $\{p_1, ..., p_n\}$. Now, suppose there are sequences $\{\Delta_{1,m}\}_{m=1}^{\infty}, ..., \{\Delta_{n,m}\}_{m=1}^{\infty}$ so that $\Delta_{i,m} \xrightarrow{m\to\infty} 0 \ \forall i = 1, ..., n$. Then, obviously,

$$(p_1 + \Delta_{1,m}, ..., p_n + \Delta_{n,m}) \xrightarrow{m\to\infty} (p_1, ..., p_n).$$

Referring to Definition 1(b), the continuity of the weight expressions is proved by showing that

$$\begin{aligned}&(p_1 + \Delta_{1,m}, ..., p_n + \Delta_{n,m}) \xrightarrow{m\to\infty} (p_1, ..., p_n) \Rightarrow \\ &f(p_1 + \Delta_{1,m}, ..., p_n + \Delta_{n,m}, \boldsymbol{w}) \xrightarrow{m\to\infty} f(p_1, ..., p_n, \boldsymbol{w}),\end{aligned} \tag{B.1}$$

where f is the given weight expression and $\boldsymbol{w}$ refers to the weights. The proof for the different weight expressions goes as follows:

1. $f = WMEAN$

$$WMEAN(p_1 + \Delta_{1,m}, ..., p_n + \Delta_{n,m}, \boldsymbol{w}) = \frac{\sum_{i=1}^n w_i(p_i + \Delta_{i,m})}{\sum_{i=1}^n w_i} =$$
$$\frac{\sum_{i=1}^n w_i p_i}{\sum_{i=1}^n w_i} + \frac{\sum_{i=1}^n w_i \Delta_{i,m}}{\sum_{i=1}^n w_i} \xrightarrow{m\to\infty} \frac{\sum_{i=1}^n w_i p_i}{\sum_{i=1}^n w_i} = WMEAN(p_1, ..., p_n, \boldsymbol{w}).$$

2. $f = WMIN$

$$WMIN(p_1 + \Delta_{1,m}, ..., p_n + \Delta_{n,m}, \boldsymbol{w}) = \min_{i=1,...,n} \left\{ \frac{w_i * (p_i + \Delta_{i,m}) + \sum_{j\neq i}^n (p_j + \Delta_{j,m})}{w_i + n - 1} \right\} =$$
$$\min_{i=1,...,n} \left\{ \frac{w_i * p_i + \sum_{j\neq i}^n p_j}{w_i + n - 1} + \frac{w_i * \Delta_{i,m} + \sum_{j\neq i}^n \Delta_{j,m}}{w_i + n - 1} \right\} \xrightarrow{m\to\infty}$$
$$\min_{i=1,...,n} \left\{ \frac{w_i * p_i + \sum_{j\neq i}^n p_j}{w_i + n - 1} \right\} = WMIN(p_1, ..., p_n, \boldsymbol{w}).$$

3. $f = WMAX$

Replace WMIN and $\min$ by WMAX and $\max$ in the proof for WMIN, respectively.

4. $f = MIXMINMAX$

$$\begin{aligned}&MIXMINMAX(p_1+\Delta_{1,m}, ..., p_n+\Delta_{n,m}, \boldsymbol{w}) =\\ &\frac{w_{MIN} * \min_{i=1,...,n}\{p_i+\Delta_{i,m}\} + w_{MAX} * \max_{i=1,...,n}\{p_i+\Delta_{i,m}\}}{w_{MIN}+w_{MAX}} \xrightarrow{m\to\infty}\\ &\frac{w_{MIN} * \min_{i=1,...,n}\{p_i\} + w_{MAX} * \max_{i=1,...,n}\{p_i\}}{w_{MIN}+w_{MAX}} =\\ &MIXMINMAX(p_1, ..., p_n, \boldsymbol{w}).\end{aligned}$$

□

Lemma 2. *Let $\omega_1 : \mathbb{R}^n \to \mathbb{R}$ and $\omega_2 : \mathbb{R} \to \mathbb{R}$ be continuous functions. Then, the composite function $\omega_2 \circ \omega_1 : \mathbb{R}^n \to \mathbb{R}$ is continuous.*

Proof. The proof refers to Definition 1(a) of continuity. Let $\epsilon > 0$. Then, because ω_2 is continuous, there exists $\delta_1 > 0$ so that

$$\boldsymbol{u}, \boldsymbol{y} \in \mathbb{R}^n \wedge \|\omega_1(\boldsymbol{u}) - \omega_1(\boldsymbol{y})\| < \delta_1 \Rightarrow \|\omega_2(\omega_1(\boldsymbol{u})) - \omega_2(\omega_1(\boldsymbol{y}))\| < \epsilon. \tag{B.2}$$

On the other hand, because ω_1 is continuous, there exists δ_2 so that

$$\|\boldsymbol{u} - \boldsymbol{y}\| < \delta_2 \Rightarrow \|\omega_1(\boldsymbol{u}) - \omega_1(\boldsymbol{y})\| < \delta_1. \tag{B.3}$$

Thus, Equations B.2 and B.3 imply that

$$\forall\, \epsilon > 0\, \exists\, \delta_2 > 0 \ : \ \|\boldsymbol{u} - \boldsymbol{y}\| < \delta_2 \Rightarrow \|\omega_2(\omega_1(\boldsymbol{u})) - \omega_2(\omega_1(\boldsymbol{y}))\| < \epsilon.$$

□

Lemma 3. *$Normpdf(u, \mu, \sigma^2)$ is continuous with respect to μ.*

Proof.

$$g_1(\mu) = -\frac{1}{2\sigma^2}(u-\mu)^2,$$

is continuous because it is a polynomial function [50]. The exponent function

$$g_2(y) = e^y,$$

is continuous as well [50]. By Lemma 2, the composite function

$$g_2 \circ g_1(\mu) = g_2(g_1(\mu)) = e^{-\frac{1}{2\sigma^2}(u-\mu)^2},$$

is continuous. Hence, also

$$Normpdf(u, \mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}(u-\mu)^2},$$

is continuous with respect to μ. □

Lemma 4. *For arbitrary $c_1, c_2 \in \mathbb{R}$, the function $g : \mathbb{R} \to \mathbb{R}$ defined by*

$$g(\mu) = \int_{c_1}^{c_2} Normpdf(u, \mu, \sigma^2) du,$$

is continuous.

Proof. Let there be a function $v : \mathbb{R} \to \mathbb{R}$ defined by

$$v(u) = Normpdf(u, \mu, \sigma^2),$$

a sequence $\{\mu_k\}_{k=1}^{\infty} \in \mathbb{R}$ such that

$$\mu_k \xrightarrow{k\to\infty} \mu,$$

and a set of functions $\{v_k(u)\}_{k=1}^{\infty} : \mathbb{R} \to \mathbb{R}$ defined by

$$v_k(u) = Normpdf(u, \mu_k, \sigma^2).$$

Then, Lemma 3 implies that

$$v_k(u) \xrightarrow{k\to\infty} v(u). \tag{B.4}$$

Moreover, it applies that

$$0 < v_k(u) \leq \frac{1}{\sqrt{2\pi}\sigma} \;\; \forall\, k \geq 1, \forall\, u \in \mathbb{R}. \tag{B.5}$$

Now, based on Equations B.4 and B.5, Lebesgue's dominated convergence theorem [53] implies that

$$\lim_{k\to\infty} \int_{c_1}^{c_2} v_k(u) du = \int_{c_1}^{c_2} v(u) du \;\; \forall\, c_1, c_2 \in \mathbb{R}. \tag{B.6}$$

In turn, Equation B.6 implies that the function

$$q(\mu) = \int_{c_1}^{c_2} Normpdf(u, \mu, \sigma^2) du,$$

is continuous. □

Definition 2. *Function $h : [0,1]^n \to \mathbb{R}$ is defined by*

$$h(p_1, ..., p_n) = \int_{c_1}^{c_2} TNormpdf(u, f(p_1, ..., p_n, \boldsymbol{w}), \sigma^2, 0, 1)du, \tag{B.7}$$

where $f \in \{$WMEAN, WMIN, WMAX, MIXMINMAX$\}$ is any of the weight expressions, see Equations 3.1-3.4, $\boldsymbol{w}$ are the weights, and $TNormpdf$ is the probability density function of a doubly truncated normal distribution, see Equation 3.3.5.

Lemma 5. *$h(p_1, ..., p_n)$ is continuous.*

Proof. By Lemma 4, the functions $g_1, g_2 : \mathbb{R} \to \mathbb{R}$ defined by

$$q_1(\mu) = \int_{c_1}^{c_2} Normpdf(u, \mu, \sigma^2)du, \tag{B.8}$$

and

$$q_2(\mu) = \int_{0}^{1} Normpdf(u, \mu, \sigma^2)du, \tag{B.9}$$

are continuous with respect to μ. Moreover, $q_2(\mu) > 0 \; \forall \; \mu \in \mathbb{R}$. Thus, the function $\tilde{h} : \mathbb{R} \to \mathbb{R}$ defined by

$$\tilde{h}(\mu) = \frac{q_1(\mu)}{q_2(\mu)} = \int_{c_1}^{c_2} TNormpdf(u, \mu, \sigma^2, 0, 1)du, \tag{B.10}$$

is continuous as a rational function [50]. Now, by Lemmas 1 and 2, the function

$$h(p_1, ..., p_n) = \tilde{h} \circ f(p_1, ..., p_n, \boldsymbol{w}) = \tilde{h}(f(p_1, ..., p_n, \boldsymbol{w})), \tag{B.11}$$

is continuous. □

Lemma 6. *$h(p_1, ..., p_n)$ is integrable.*

Proof. For any $\sigma^2 > 0$, h is obviously bounded. This and Lemma 5 imply that h is integrable, see [34]. □

Theorem 1. *For $i = 1, ..., n$, let there be an interval $[a_i, b_i] \subset [0,1]$ and a set $\{\chi_i^{k_i}\}_{k_i=1}^{s}$ defined by*

$$\chi_i^{k_i} = a_i + (k_i - 1)\frac{b_i - a_i}{s - 1}. \tag{B.12}$$

Then, it applies that

$$\frac{1}{s^n}\sum_{k_1=1}^{s}...\sum_{k_n=1}^{s} h(\chi_1^{k_1},...,\chi_n^{k_n}) \xrightarrow{s\to\infty} E(h(\chi_1,...,\chi_n)|\chi_i \sim U(a_i,b_i)\ \ \forall\, i=1,...,n). \tag{B.13}$$

Proof. Let the points $\{b_i^*\}_{i=1}^n$ be defined by

$$b_i^* = b_i + \frac{b_i - a_i}{s-1}, \tag{B.14}$$

and an elementary n-dimensional volume element be defined by

$$\Delta V = \prod_{i=1}^{n} \frac{b_i^* - a_i}{s} = \frac{1}{s^n}\prod_{i=1}^{n}(b_i^* - a_i). \tag{B.15}$$

Equation B.14 implies that

$$\lim_{s\to\infty} b_i^* = b_i\ \ \forall\, i = 1,...,n.$$

On the other hand, $h(p_1,...,p_n)$ is known to be integrable, see Lemma 6. Thus, it applies that

$$\sum_{k_1=1}^{s}...\sum_{k_n=1}^{s} h(\chi_1^{k_1},...,\chi_n^{k_n})\Delta V \xrightarrow{s\to\infty} \int_{a_1}^{b_1}...\int_{a_n}^{b_n} h(\chi_1,...,\chi_n)d\chi_n...d\chi_1. \tag{B.16}$$

Together, Equations B.15 and B.16 imply that

$$\frac{1}{s^n}\sum_{k_1=1}^{s}...\sum_{k_n=1}^{s} h(\chi_1^{k_1},...,\chi_n^{k_n}) \xrightarrow{s\to\infty} \int_{a_1}^{b_1}...\int_{a_n}^{b_n} h(\chi_1,...,\chi_n) * \frac{1}{\prod_{i=1}^{n}(b_i - a_i)}d\chi_n...d\chi_1 =$$
$$E(h(\chi_1,...,\chi_n)|\chi_i \sim U(a_i,b_i)\ \ \forall\, i=1,...,n). \tag{B.17}$$

□

Next, recall properties of RNM presented in Section 3.3.5. For a given combination of states $(x_1,...,x_n)$ of the parent nodes $X_1,...,X_n$, s^n combinations of sample points $\{(z_{1,k},...,z_{n,k})\}_{k=1}^{s^n}$ are formed so that $z_{i,k}$ is always one of the s equidistant sample points taken from the state interval $z_i = [a_i,b_i] \subset [0,1]$ corresponding to x_i. For each of the combinations $\{(z_{1,k},...,z_{n,k})\}_{k=1}^{s^n}$, the mean parameter μ_k is calculated by

$$\mu_k = f(z_{1,k},...,z_{n,k},\boldsymbol{w}), \tag{B.18}$$

where f is the weight expression and $\boldsymbol{w}$ are the weights — see Equations 3.1-3.4.

For each μ_k, conditional probabilities are calculated according to Equation 3.8, i.e.,

$$P(X_{n+1} = x_{n+1} | X_1 = x_1, ..., X_n = x_n; \mu_k) = \int_{a_{n+1}}^{b_{n+1}} TNormpdf(x, \mu_k, \sigma^2, 0, 1)dx, \tag{B.19}$$

where $[a_{n+1}, b_{n+1}] = z_{n+1}$ is the state interval corresponding to the state x_{n+1} of the child node X_{n+1}. In addition, recall from Section 4.1 that the random variable γ is defined by Equations 4.1-4.3 as

$$\begin{cases} \gamma = \int_{a_{n+1}}^{b_{n+1}} TNormpdf(u, \mu, \sigma^2, 0, 1)du, \\ \mu = f(\chi_1, ..., \chi_n, \boldsymbol{w}), \\ \chi_i \sim U(a_i, b_i), \ i = 1, ..., n. \end{cases} \tag{B.20}$$

Corollary 1. *Let $P(X_{n+1} = x_{n+1} | X_1 = x_1, ..., X_n = x_n; \mu_k)$ and γ be defined according to Equations B.19 and B.20, respectively. Then,*

$$\frac{1}{s^n} \sum_{k=1}^{s^n} P(X_{n+1} = x_{n+1} | X_1 = x_1, ..., X_n = x_n; \mu_k) \xrightarrow{s \to \infty} E(\gamma). \tag{B.21}$$

Proof. Equations B.7 and B.20 imply that with the substitution

$$\begin{cases} c_1 = a_{n+1}, \\ c_2 = b_{n+1}, \end{cases} \tag{B.22}$$

it applies that

$$\gamma = h(\chi_1, ..., \chi_n). \tag{B.23}$$

Equation B.23 implies that

$$E(\gamma) = E(h(\chi_1, ..., \chi_n)). \tag{B.24}$$

That is, the right hand sides of Equations B.17 and B.21 are the same. On the other hand, with the substitutions defined by Equation B.22, it follows from Equations B.7, and B.19 that

$$P(X_{n+1} = x_{n+1} | X_1 = x_1, ..., X_n = x_n; \mu_k) = h(z_{1,k}, ..., z_{n,k}). \tag{B.25}$$

On the other hand, for each $i = 1, ..., n$, the set $\{\chi_{i=i}^{k_i}\}_{k_i=1}^{s}$ defined by Equation B.12 corresponds to the equidistant sample points taken from the state interval

z_i. Thus, it applies that

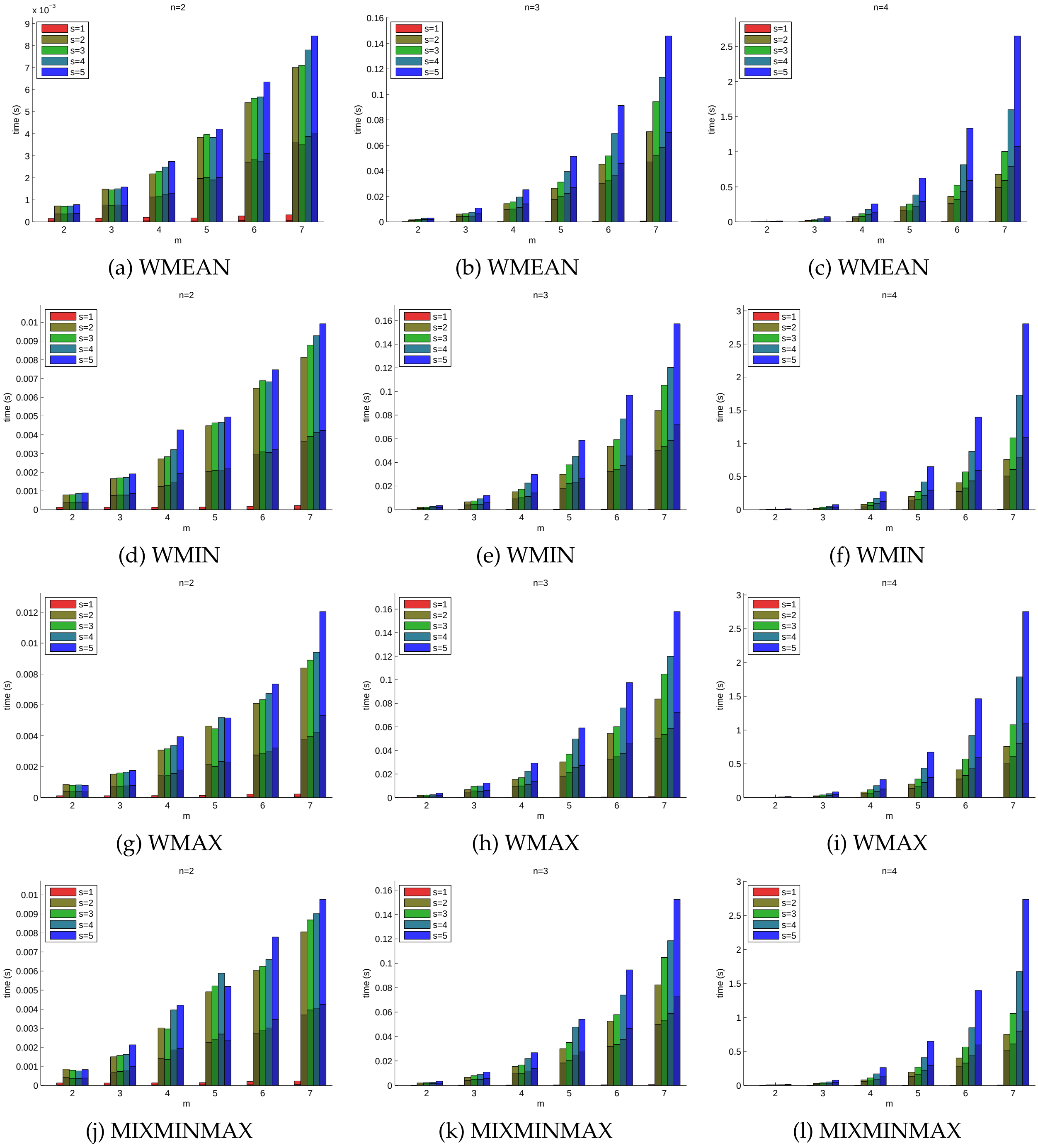
$$\frac{1}{s^n}\sum_{k_1=1}^{s}\dots\sum_{k_n=1}^{s} h(\chi_1^{k_1},...,\chi_n^{k_n}) = \frac{1}{s^n}\sum_{k=1}^{s^n} h(z_{1,k},...,z_{n,k}) =$$
$$\frac{1}{s^n}\sum_{k=1}^{s^n} P(X_{n+1}=x_{n+1}|X_1=x_1,...,X_n=x_n;\mu_k), \tag{B.26}$$

where the last equality is due to Equation B.25. Equation B.26 shows that the left hand sides of Equations B.17 and B.21 are the same. Hence, by Equations B.24 and B.26 as well as Theorem 1, it applies that the claim presented in Equations 4.5 and B.21 is true.

□

Appendix C

Results of Experiment in Section 5.1



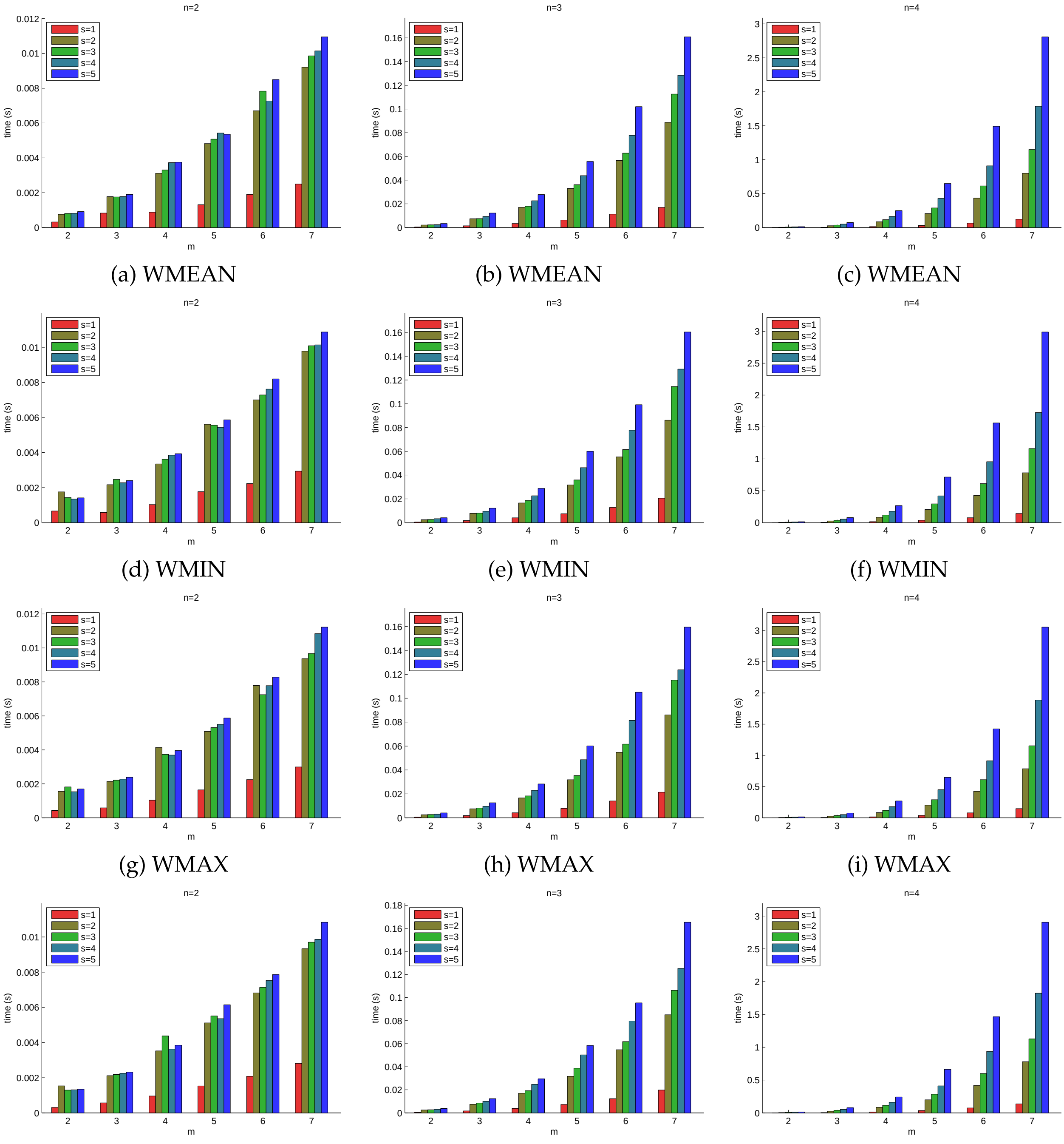
(a) WMEAN (b) WMEAN (c) WMEAN

(d) WMIN (e) WMIN (f) WMIN

(g) WMAX (h) WMAX (i) WMAX

(j) MIXMINMAX (k) MIXMINMAX (l) MIXMINMAX

Figure C-1: Average calculation times of a CPT with implementation A and with different weight expressions for varying number of parent nodes n, number of states of the nodes m, and number of sample points s.



(a) WMEAN

(b) WMEAN

(c) WMEAN

(d) WMIN

(e) WMIN

(f) WMIN

(g) WMAX

(h) WMAX

(i) WMAX

(j) MIXMINMAX

(k) MIXMINMAX

(l) MIXMINMAX

Figure C-2: Average calculation times of a CPT with implementation B and with different weight expressions for varying number of parent nodes n, number of states of the nodes m, and number of sample points s.