Copyright 2014 Yerkin Garyshuly Abdildin

MULTIATTRIBUTE UTILITY FUNCTIONS FOR THE DEEP BOREHOLE FILTER RESTORATION PROBLEM

BY

YERKIN GARYSHULY ABDILDIN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Industrial Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2014

Urbana, Illinois

Doctoral Committee:

Associate Professor Ali E. Abbas, Chair, Director of Research
Professor Rizwan Uddin, Co-Director of Research
Associate Professor Ramavaparu S. Sreenivas
Assistant Professor James T. Allison

ABSTRACT

The energy problem is one of the biggest challenges facing the World in the 21st century. It is related to the issues of natural resource extraction, resource depletion, power generation, environmental degradation, and atmospheric change such as global warming. Since more than 80% of the world's primary energy is generated from fossil fuels (coal, oil, gas), emissions of carbon dioxide (CO_2) from all fossil-fuel burnings are the largest cause of climate change. Global climate disruption, in turn, impacts on human health, flora, and fauna. The global energy demand is expected to double by 2050 and that is inevitably due to global population growth, global economic growth, and continued urbanization. To meet the increasing demand for energy and to avoid catastrophic climate change, increases in energy efficiency and increases in the fraction of low carbon energy sources are required.

Uranium is a good energy source, because it has high energy density and nuclear power does not contribute to carbon dioxide emissions. However, difficulties in uranium mining cause large worldwide shortages of uranium for power generation. Decision makers in uranium mining are often challenged by various uncertainties in their decision problems (financial, technological, geological) and multiple objectives (increase profits, decrease radiation hazards, improve safety of operations, preserve environments). This dissertation studies multiattribute utility functions for modeling such challenging decision problems using the example of the deep borehole filter restoration problem from the uranium extraction industry. In this problem, the filter of the production borehole (or well) is periodically contaminated or clogged, causing significant uranium output reduction. The efficient modeling of this decision-making problem is of paramount importance for uranium mining worldwide and requires normative decision analysis.

Motivated by the complexity of multiattribute decision problems under uncertainty and multiple objectives, this dissertation considers a set of open research problems related to the number of attributes and their degree of 'interdependence under uncertainty,' formally, utility dependence and independence. This dissertation characterizes the special functional forms of multiattribute utility functions (MUFs) under the partial utility independence (PUI) condition, verifies their applicability to the deep borehole filter restoration problem, evaluates the alternatives of the decision problem by three different approaches, and introduces novel methods for excluding redundant utility assessments.

In Part I of this work we present our study: (1) what are the objectives and the corresponding attributes (i.e. factors or criteria of the decision-making process) of the deep borehole filter restoration problem from the well-field manager's point of view, (2) does the ultimate decision maker find these attributes good or not, (3) does utility independence (UI) among the attributes exist in this decision problem, (4) whether or not the canonical functional forms from the theory of decision analysis are applicable to this decision problem, (5) are the decision analysis tests easy for the experts in the uranium extraction industry to use? For this, we create new tests for assessing interdependence among the attributes of the decision problem. In the first experiment with Test 1, 105 professionals in uranium mining were requested to provide their preferences among the four most important attributes from the well-field manager's point of view. In the second experiment with the more formal Test 2, 40 experts were asked to provide their preferences from among three of these four attributes.

Based on the results of the experimental study, with 95% confidence we can conclude that the proportion of the population (thousands) of experts who assert utility independence of

the attributes is at most 0.23. The results imply that the conventional approach, the assumption of utility independence, may not be valid for our decision problem.

In Part II, we evaluate five alternatives of the deep borehole filter restoration problem by constructing a multiattribute utility function (MUF) utilizing two different approaches: (i) under the assumption of utility independence of attributes, and (ii) with the existing (assessed from the decision maker) partial utility independence of the attributes. The four most important attributes were selected by the ultimate decision maker, the Deputy Director General of a transnational corporation. We first assume utility independence of the attributes and utilize the corresponding multilinear form of the MUF. We then utilize the general decomposition approach for constructing the MUF under the assessed partial utility independence conditions. Finally, we compare the results of these two approaches and the results of the profit analysis.

By direct assessments, (i) we find that utility independence among the attributes of the deep borehole filter restoration problem is not a valid assumption, (ii) we verify that the decision maker's preferences assert partial utility independence, (iii) we illustrate the assessments required under partial utility independence assertions, (iv) we compare the decisions made using the assumption of utility independence and the existing partial utility independence conditions, and (v) determine that the assumption of utility independence yields recommendations, which are different from the true preferences of the decision maker. Our results also demonstrate that the assessments required for the construction of the MUF by the utility independence approach is easier for the decision maker (DM), but the DM is more comfortable with the partial utility independence conditions. To our knowledge, this is the first practical study on the comparison of profit analysis, the utility independence approach, and the partial utility independence approach in a complex real life multiattribute decision problem.

In Part III, we present algorithms for excluding redundant assessments from the set of assessments required for construction of the multiattribute utility function. In complex decision problems, the number of utility assessments, and, therefore, the number of questions for the decision maker, increases dramatically. It is thus very important to check the consistency of the assessments and eliminate all redundant utility assessments. With the efficient algorithms introduced in this dissertation, the elimination of the redundant utility assessments is considerably simplified.

The results of this dissertation were applied to an important and complex decision problem in the uranium extraction industry, the deep borehole filter restoration problem. The decision modeling proposed in this dissertation should also help decision makers in addressing the worldwide 14% shortage of uranium needed for nuclear power generation.

ACKNOWLEDGMENTS

I would like to thank everyone who supported, motivated, and helped me during my years of study at the University of Illinois at Urbana-Champaign.

I am very grateful to my academic advisor, Professor Ali E. Abbas, for his guidance, encouragement, and patience throughout this long journey. His invaluable knowledge, outstanding research experience, and in-depth charisma have always been motivating to me. I am also grateful to my co-advisor, Professor Rizwan Uddin, for his support and guidance, for his meticulous reviews and valuable comments on my dissertation. I owe thanks to the members of my Doctoral Committee. I thank Professor Ramavarapu S. Sreenivas for his constant support, fruitful discussions, and career advice. I also thank Professor James T. Allison for his time and helpful suggestions.

I am grateful to the Government of Kazakhstan for awarding me the Bolashak Presidential Scholarship, which was a vital support in my initial years in the US. I am also grateful to President Nursultan Nazarbayev for initiating this scholarship and to the Center of International Programs for administrating it in an innovative way. In particular, I am thankful to Alzhan Braliev, Dina Akhmetova, Sayasat Nurbek, Asset Abdualiyev, Gani Nygymetov, Galiya Dosmukhambetova, Indira Mamasheva, and Aliya Baltabayeva. I am also thankful to the Department of Industrial and Enterprise Systems Engineering for providing me with teaching assistantships, which were both a financial support and the opportunity to perfect my teaching skills.

Special thanks to James Leake, Director of Engineering Graphics, for giving me my first chance to immerse myself into the classroom atmosphere as a teaching assistant. I also thank

Professor Steven M. Lavalle, Professor Brian P. Bailey, Dr. Cinda Heeren, and Dr. Lawrence Angrave from the Department of Computer Science, who made their time and support available for me at any time.

I thank the proficient staff members in the Department of Industrial and Enterprise Systems Engineering: Donna Eiskamp, Holly Kizer, Debi Hilligoss, Lynnell Lacy, Amy Summers, Barbara Bohlen, Heidi Craddock, Peggy Regan, and Randy Elkins, who helped me with numerous administrative tasks. I also thank Mary Beth A. Kelley from the Department of Computer Science. I remember late Professor Emerita Geneva Belford for her kindness and support in my first years at the University of Illinois at Urbana-Champaign (UIUC). I was never shy about stopping by her office and discussing any questions related to my academic life.

I also thank many professionals in the uranium extraction industry, who helped me with the experimental studies. In particular, I thank Zhakiya Arystanbayev, Berdi Bekmagambetov, Baurzhan Duisebayev, Sabyrzhan Duisebayev, Marat Oralbekov, Kuanysh Kembayev, Sakypnazar Aitenov, Serik Nurkenov, and Jeremy Licht, whose help was extremely important.

It has been a pleasure to study at the University of Illinois at Urbana-Champaign, meet phenomenal individuals and genius students from around the world, and find great friends.

I thank my labmates Mahmood Al Kindi, Sara Behdad, Xi Chen, Ryan Hafiz, Andrea Hupman, Aaron Joseph, Nihan Görmez Karahan, Vijitashwa Pandey, Heather Ritchey, James Schreiner, Zhengwei Sun, Muhammed Sutcu, Aida Williams, Lijuan Wu, and Yuan Zhao for innumerable meetings, productive discussions, and the fun we have had in the lab.

I thank Shahzad Bhatti, Joe Bradly, Fawad Hassan, Douglas King, Rustem Khafizov, Sherzod Madrahimov, Farzad Usefian, Dmitry Yershov, Carol Ehrhardt, Sara Behdad, Andrea Hupman, Aaron Joseph, Heather Ritchey, James Schreiner, Zhengwei Sun, and Muhammed

Sutcu for their time spent on my practice presentations and invaluable suggestions in mastering my presentation skills.

I thank Naeem Sheikh for his help countless times and in multiple forms. He is a man with a big heart, who would help me and others in dealing with any problems: from day-to-day issues to various mathematical problems. I also thank Azeem Sarwar, a true leader and motivator, who was continuously inspiring me and others to be persistent in achieving our goals while he was in Urbana-Champaign. I thank Zahid Anwar and Mueen Nawaz for sharing their knowledge and experience about interacting in the academic community. I thank Numair Ahmed for providing me with study material for the qualifying exam in Control Systems. I am very thankful to my friend Jayash Koshal for sharing good times at UIUC and being so supportive and helpful when I had family issues. I also thank my friends Alexander Kotov and Fawad Hassan for unforgettable experiences we have had during our years in Urbana-Champaign.

I am thankful for Jaouad El Hamrini, Haris Anwaar Chaudhry, Malek Chaabane, Si Ilyes, Mohamed Yousef Mohamed, Umit Deniz Tursun, Banafsheh Behzad, Anand Deshmukh, Arash Khatibi, Elham Zafarani, Alok Tiwari, Rasoul Etesami, and others for their companionship.

I also thank many awesome people in or from Kazakhstan, who helped me with various activities and also made my life joyful while I was studying in the US.

I thank Agipa Monobayeva, Aibek Yessenalin, Aliya Azhibekova, Aliya Kuzhabekova, Aslan Nazaraliyev, Assel Akzhalova, Baurzhan Konurbayev, Baurzhan Kozhabayev, Bolatbek Barkenov, Daniyar Kassymzhanov, Daniyar Nurtazin, Kairat Nitkaliyev, Meruert Aliyeva, Nurlan Igembaev, Olzhas Akylbekov, Ramesh Kini, Sagat Makhanov, Serik Aliaskarov, Timur Aubakir, Timur Umarov, Turlan Yermukhanov, Yerlan Meirambayev, Yerzhan Tutkushev, and Zhazira Mubarak for their friendship and support.

I thank all members of the Kazakh Student Association at Urbana-Champaign for making my campus life more enjoyable. In particular, I thank Nursultan Alibaev, Olzhas Ayazbayev, Galikhan Baitikov, Olzhas Karim, Tungyshbek Kassymzhanov, Azamat Makhsudov, Aibar Nurmukhanov, Sergey Popov, Yerzhan Suleimenov, Azamat Uskenbayev, and Madi Zholbaryssov for helping me with various things while I was at UIUC.

I am grateful to hundreds of extraordinary, open-minded, hard-working, humble, unique, and so different students at UIUC, whom I taught during past years, for their passion and love. I really enjoyed this time.

Many brilliant people motivated, encouraged, supported, and helped me in various ways in preparing this dissertation, and my apologies go to those whose names I could have missed.

Finally, I thank my family and relatives, who are my biggest supporters. Their emotional support, tremendous enthusiasm, and unconditional love motivate me all the time. I will always remember the strong-willed personality of my father, who could not wait to share this happy achievement. He was a real man, true warrior, who never showed any sign of complaint or indecision.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS xv

CHAPTER 1: THE DEEP BOREHOLE FILTER RESTORATION PROBLEM 1

1.1 THE WORLD'S BIG CHALLENGE 1

1.1.1 The energy problem 1

1.1.2 World energy markets by fuel type 2

1.1.3 World carbon dioxide emissions 3

1.1.4 Uranium as an energy source 4

1.2 BACKGROUND ON DECISIONS IN URANIUM MINING 5

1.2.1 Introduction to uranium mining 6

1.2.2 In situ leach mining and the deep borehole filter restoration problem 7

1.2.3 A pyramid of decision makers in uranium mining 10

1.3 DEFINING THE DEEP BOREHOLE FILTER RESTORATION PROBLEM 11

1.3.1 Description of the problem 12

1.3.2 Alternatives 13

1.3.3 Objectives and attributes of the decision problem by the well-field manager 15

1.3.4 Objectives and attributes of the decision problem by the ultimate decision maker 16

1.3.5 Uncertainties 18

1.3.6 Laymen ranking of the alternatives 22

1.3.7 The influence diagram 22

1.3.8 The decision tree 23

1.4 SUMMARY 25

CHAPTER 2: RELATED WORK, NOTATIONS AND DEFINITIONS 26

2.1 MULTIATTRIBUTE DECISION PROBLEMS: BACKGROUND 26

2.2 DECISION ANALYSIS IN INDUSTRIAL PROJECTS 28

2.3 REVIEW OF THE BASIC CONCEPTS AND DEFINITIONS 29

2.3.1 Review of the Utility Dependence Matrix 32

2.3.2 Review of the Basic Expansion Theorem 33

2.3.3 Review of the Iterative Decomposition 35
2.3.4 Review of Utility Independence 36
2.3.5 Review of the Canonical Form 37
2.4 DISSERTATION CONTRIBUTION AND OVERVIEW 38

CHAPTER 3: PROFIT ANALYSIS OF THE ALTERNATIVES OF THE DEEP BOREHOLE FILTER RESTORATION PROBLEM 41
3.1 DETERMINISTIC ANALYSIS 41
3.1.1 A value model 41
3.1.2 Sensitivity analysis 42
3.1.3 Cumulative Distribution Functions of each alternative 43
3.2 UTILITY ANALYSIS BASED ON ATTRIBUTE PROFIT 44
3.2.1 Assessing utility function of Profit 44
3.2.2 Evaluating the alternatives of the borehole recovery problem based on Profit 46

CHAPTER 4: TESTING INTERDEPENDENCE AMONG THE ATTRIBUTES OF THE BOREHOLE RECOVERY PROBLEM 48
4.1 OVERVIEW 48
4.2 NEW TESTS AND EXPERIMENTS ON PARTIAL UTILITY INDEPENDENCE 48
4.2.1 Test 1: Test of partial utility independence conditions in Experiment 1 48
4.2.2 Test 2: Test of partial utility independence conditions in Experiment 2 50
4.3 ANALYSIS OF THE INTERDEPENDENCE OF THE ATTRIBUTES 52
4.3.1 Formulating a statistical test on utility and partial utility independence conditions among the attributes of the borehole recovery problem 52
4.3.2 The results of Test 1 in Experiment 1 based on a 95% confidence interval 53
4.3.3 The results of Test 2 in Experiment 2 based on a 95% confidence interval 54
4.4 DISCUSSION OF THE RESULTS OF THE EXPERIMENTS 54
4.4.1 Distributions of the assessors 54
4.4.2 Discussion of Experiment 1 with Test 1: novelty, importance, and results 55
4.4.3 Discussion of Experiment 2 with Test 2: novelty, importance, and results 56
4.5 SUMMARY 67

CHAPTER 5: EVALUATING ALTERNATIVES OF THE DEEP BOREHOLE FILTER RESTORATION PROBLEM 69

5.1 MULTIATTRIBUTE ANALYSIS UNDER THE UTILITY INDEPENDENCE ASSUMPTION 69

5.1.1 Multilinear form of the MUF for the deep borehole filter restoration problem 69

5.1.2 Assessments of the multilinear form of the MUF 70

5.1.3 Ranking of the alternatives under the utility independence assumption 75

5.2 MULTIATTRIBUTE ANALYSIS UNDER PARTIAL UTILITY INDEPENDENCE 76

5.2.1 Decomposition of the MUF for the deep borehole filter restoration problem 76

5.2.2 Assessments of the MUF for the deep borehole filter restoration problem 77

5.2.3 Ranking of the alternatives with assessed partial utility independence conditions . 80

5.3 DISCUSSION OF THE RESULTS 81

5.4 SUMMARY 82

CHAPTER 6: AN ALGORITHM FOR EXCLUDING REDUNDANT ASSESSMENTS .. 84

6.1 TERNARY MATRIX REPRESENTATION OF UTILITY ASSESSMENTS 85

6.2 EXCLUSION ALGORITHM 89

6.2.1 Brute-force approach 89

6.2.2 Twos-Complement method 91

6.2.3 The detailed pseudocodes of both approaches 95

6.3 COMPARISON OF THE RUNNING TIMES: SIMULATION RESULTS 96

6.4 SUMMARY 99

CHAPTER 7: TERNARY-DECIMAL EXCLUSION ALGORITHM 100

7.1 INTRODUCTION 100

7.1.1 Review of a conversion from the ternary number system to the decimal 100

7.1.2 Review of a conversion from a decimal number to a ternary number 101

7.2 THE MAIN IDEA 101

7.3 PSEUDOCODES 103

7.3.1 High-level pseudocode of the Ternary-Decimal Exclusion algorithm 103

7.3.2 Step 1: Converting a ternary matrix to a decimal vector 104

7.3.3 Step 2: Sorting the decimal vector and eliminating duplicates 105

7.3.4 Step 3: Recursive divide and conquer approach for excluding redundancies 106

7.3.5 Step 4: Converting the decimal vector to a ternary matrix 107
7.3.6 Step 5: Eliminating specific redundancies from the ternary matrix 107
7.3.7 An illustrative example 108
7.4 SIMULATION RESULTS 111
7.5 APPLICATION TO THE DEEP BOREHOLE FILTER RESTORATION PROBLEM 113
7.5.1 Three-attribute case: Utility Dependence Matrix #35 from Experiment 2 113
7.5.2 Four-attribute case: Utility Dependence Matrix #4 from Experiment 1 116
7.5.3 Five-attribute case: randomly generated Utility Dependence Matrix 121
7.6 APPLICATION TO THE 10-ATTRIBUTE NUCLEAR POWER SITING PROBLEM 122
7.7 SUMMARY AND FUTURE WORK 124

CHAPTER 8: CONCLUSION 125

APPENDIX A (TEST 1): Experimental assessments of utility dependence matrices for the deep borehole filter restoration problem in uranium mining. 128

APPENDIX B (TEST 2): An experimental assessment of interdependence of attributes 130

APPENDIX C (RESULTS OF TEST 1): 105 4x4 utility dependence matrices assessed for the deep borehole filter restoration problem during Experiment 1. 138

APPENDIX D (RESULTS OF TEST 2): Actual responses of 40 experts in Experiment 2 142

APPENDIX E (RESULTS OF EXPERIMENT 2): 40 UDMs converted to a binary form 143

APPENDIX F: DETERMINING THE NUMBER AND CONVERGENCE OF SPECIAL FUNCTIONAL FORMS FOR PARTIAL UTILITY INDEPENDENCE 144

F1: New definitions 144
F2: Determining the number of the canonical forms 144
F3: Determining the convergence of the canonical forms 146

F4: Determining the number and convergence of proper partial forms 148

APPENDIX G: Elicitation of the utility function of attribute X_3 (safety of operations) in uranium extraction industry (in number of days without accident) 151

APPENDIX H: Assessment of scaling constants for the multilinear form of the multiattribute utility function 152

APPENDIX I: Assessments of the value function $V(x_2, x_3)$ when $(x_1 = x_1^*, x_4 = x_4^*)$. (The 1st and 2nd pages only). 157

APPENDIX J: Elicitation of the utility function over the value function $U_V(V(x_2, x_3))$ when $(x_1 = x_1^*, x_4 = x_4^*)$. 159

APPENDIX K: C++ code of the exclusion algorithms 160

APPENDIX L: MATLAB code of the Ternary-Decimal Exclusion algorithm 163

REFERENCES 167

LIST OF ABBREVIATIONS

BET	Basic Expansion Theorem
Btu	British thermal unit
DA	Decision Analysis
DM	Decision Maker
GWe	Gigawatt-electric
IEO	International Energy Outlook
kWh	kilowatt-hour
MUF	Multiattribute Utility Function
PUI	Partial Utility Independence
UI	Utility Independence
UDM	Utility Dependence Matrix
WEC	World Energy Council
WNA	World Nuclear Association

CHAPTER 1: THE DEEP BOREHOLE FILTER RESTORATION PROBLEM

1.1 THE WORLD'S BIG CHALLENGE

1.1.1 The energy problem

The energy problem is the world's big challenge. Human civilization uses a great deal of energy in the 21st century. We use energy in the following sectors (U.S. Energy Information Administration, 2011): residential (18%), commercial (12%), industrial (50%), and transportation (20%). According to the U.S. Energy Information Administration, the total world energy use is expected to rise by 56% from 524 quadrillion British thermal units (Btu) in 2010 to 820 quadrillion Btu in 2040.

With sufficient energy supply we can more effectively address the issues of electricity, food, water, poverty, and even climate change. According to the World Energy Council (WorldEnergy.org, 2013), 1.3 billion people worldwide do not have access to electricity. The global electricity demand is expected to increase by 93% from 20.2 trillion kilowatt-hours (kWh) in 2010 to 39.0 trillion kWh in 2040 (www.eia.gov). On the one hand, this increase is aimed to lift billions out of poverty in the developing countries and keep up with the population growth. On the other hand, more than 80% of world's primary energy is generated from fossil fuels (coal, oil, gas), which indirectly "contribute" to the global climate disruption. To meet the increasing demand in energy and avoid catastrophic climate change, increases in clean energy sources are required.

1.1.2 World energy markets by fuel type

It is expected that world energy consumption will increase in the long term (IEO2013). Figure 1.1 shows the projections of world energy consumption by fuel type until 2040. Fossil fuels are predicted to continue being the major source of energy in the long term.

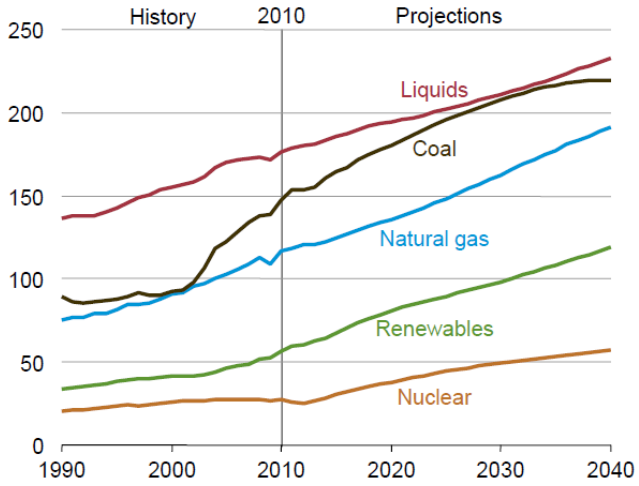


Figure 1.1 (adapted from www.eia.gov): World energy consumption by fuel type, 1990‑2040 (quadrillion Btu).

Table 1.1: Fuel type share of world marketed energy consumption, in 2010 and 2040 (%).

Fuel type	**Share of world marketed energy consumption, % in 2010**	**in 2040**
Liquids (petroleum-based)	34%	28%
Coal	28%	26.8%
Natural gas	22%	23.2%
Renewables	11%	15%
Nuclear	5%	7%

Table 1.1 illustrates the expected changes in shares of world marketed energy consumption by fuel type from 2010 to 2040. The share of the petroleum-based liquids is expected to decrease from 34% (87 million barrels per day) in 2010 to 28% (115 million barrels per day) in 2040. The share of coal is also projected to decrease from 28% (147 quadrillion Btu) to 26.8% (220 quadrillion Btu), while the shares of renewables and nuclear energy are expected to increase. The

use of natural gas is expected to increase from 113 trillion cubic feet in 2010 to 185 trillion cubic feet in 2040.

1.1.3 World carbon dioxide emissions

The increasing energy consumption has a negative impact on environment; carbon dioxide (CO_2) emissions are increasing at an alarming rate. According to the International Energy Outlook report (IEO2013), worldwide energy-related carbon dioxide emissions are expected to increase by 46% from about 31.2 billion metric tons in 2010 to 45.5 billion metric tons in 2040. Emissions of CO_2 from all fossil-fuel burnings are the largest driver of climate change (www.eia.gov). As shown in Figure 1.2, coal is the "leader," followed by liquids and natural gas.

The major actions suggested for reducing the carbon dioxide emissions are well known:

- Improve energy efficiency
- Increase the use of clean energy sources such as nuclear and renewables
- Develop methods for reducing the impact of fossil fuels
- Develop new clean energy sources.

Uranium is a good source of energy since it does not contribute to Greenhouse gas.

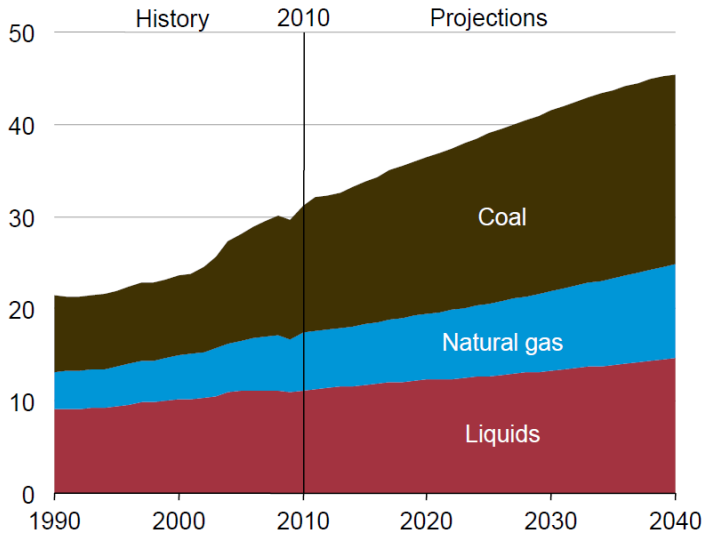


Figure 1.2 (adapted from www.eia.gov): World energy related carbon dioxide emissions by fuel type, 1990‑2040 (billion metric tons).

1.1.4 Uranium as an energy source

The International Energy Outlook report (IEO2013) states that nuclear power is one of the world's fastest-growing energy sources, which increases by 2.5% per year. Uranium is also a good energy source due to its high energy density. According to "Uraniumletter International," one pound of yellowcake (U_3O_8) is energy equivalent to 31 barrels of fuel oil or 10 tons of coal. It is also illustrative to see the energy density of uranium in comparison with some other energy sources per kilowatt-hour of electricity produced as shown in Table 1.2. In a related work, Keeney and Sicherman (1983) compare coal/nuclear choice for generating electricity.

Table 1.2 (adapted from http://uraniumletterint.com): Electricity produced from 1kg of some energy sources.

Energy source	Electricity produced
1 kg of firewood	1 kWh (kilowatt-hour)
1 kg of coal	3 kWh
1 kg of oil	4 kWh
1 kg of uranium	50000 kWh

However, the output of uranium meets only 86% of global demand for nuclear power generation (World-Nuclear.org, 2013). The significant shortage of uranium worldwide is mainly due to the complexity of the decision problems in uranium mining. Decision makers in the uranium extraction industry are challenged by various uncertainties (geological, technological, financial) and multiple objectives (profit, safety, environment). To overcome these difficulties and address the lack of uranium production, the decision makers need normative methods for more accurate solutions to complex decision problems.

1.2 BACKGROUND ON DECISIONS IN URANIUM MINING

The decision problems in nuclear energy systems are complex due to multiple objectives, interdependence of attributes, and various uncertainties. In addition to the economics, proper level of safety of operations, and environmental protection, companies should often maintain, as an example, non-proliferation, sustainability, and social impact. These attributes or factors of the decision problem may have interdependencies, which should be properly incorporated into the model. As the number of incommensurable attributes of the decision problem under uncertainty grows, the interdependence of the attributes makes the trade-off among the multiple objectives difficult.

If *utility independence* exists, i.e. preferences for uncertain outcome of one attribute do not change as the fixed levels of another attribute changes, the decision maker can solve the problem using a combination of one-attribute utility functions. Utility independence does not always hold (Farquhar, 1975; Kirkwood, 1976; Fishburn, 1977; Bell, 1979a; Keeney, 1981; Tamura and Nakamura, 1983; Harvey, 1993; Abbas and Howard, 2005), because: 1) it is a very strong condition, 2) preferences of the decision makers' often vary depending on values of the attributes

of the decision problem (see some experimental studies in Baucells and Rata, 2006; Bodily and Pfeifer, 2010; Anderson and Clemen, 2013). When utility independence (UI) does not exist, the decision makers (DM) often make the assumption of UI of attributes, which may limit the applicability of the results obtained. For example, the assumption that the DM's preferences for the attribute *safety of operations* will not change with the level of attribute *impact on personnel* may not reflect the true preferences of the DM, and the errors may lead to significant consequences that may place people at an unnecessary risk. More often, the attributes of a decision problem have partial utility independence (PUI) conditions.

The partial utility independence among the attributes of the decision problem can be incorporated by the different functional forms of multiattribute utility functions (MUFs). Thus, exploring the functional forms of MUFs is necessary and important to study the effect of incorporation of the partial utility independence. The experimental assessments of functional forms in this work are aimed to verify their practical applicability for problems in uranium mining. This will help the decision-making practitioners to make informed decisions.

1.2.1 Introduction to uranium mining

Uranium mining is the first and important step in the nuclear energy process chain. In 2011, 63,085 tons of uranium oxide U_3O_8 was produced worldwide. The market price was $130,000/ton U. According to the World Nuclear Association (WNA, 2013), there is a 14% shortage of uranium for nuclear power generation. As the global demand for nuclear energy grows, the decisions in uranium mining are becoming a problem of international concern.

About half of world uranium is extracted by in situ leach mining (ISL). Compared to the conventional mines and open-pits, ISL is more ecologically friendly and cost-effective. In this

method, the leaching process takes place underground and the uranium-bearing solution is recovered by production boreholes and pumped to the surface. The uranium-bearing solution is then transferred by header pipeline to the processing plant, or the ion exchange system, where uranium is removed from the solution, concentrated, and dried into yellowcake.

The key factor of the decisions in the uranium extraction process is the output of uranium. The uranium output from a production borehole or extraction well often slows down due to pipeline damage, deep borehole filter contamination or clogging, and the exhaustion of uranium in the ore body. The pipeline damages and filter contamination may result in spills, which may raise the radiation hazard to the local population, flora, and fauna. Damage control methods vary in multiple dimensions such as: effectiveness, additional expenses, restoration time, safety of operations, number of employees involved in the process, radiation hazard, and contaminated area. These factors, or attributes, may have interdependencies, which make the decision problems complex. Since the decisions that the companies make are ad hoc or based on economic models, they may be limited to addressing the decision maker's preference according to only one objective (e.g. maximize profit), but not necessarily in multiple objectives (for example, maximize output, minimize job hazard and contaminated area). A normative analysis can help to determine the best decision in a systematic way. In this work, we will analyze the borehole recovery problem in in situ leach mining.

1.2.2 In situ leach mining and the deep borehole filter restoration problem

According to in situ leach (ISL) mining, the uranium ore field is drilled over by technological boreholes into many hexagons. There are also some other schemes or patterns such as linear well pattern or a rectangular system, but our focus in this work is on hexagonal layouts.

A typical hexagon in uranium technological fields, schematically shown in Figure 1.3, has six injection boreholes on its vertices and one production borehole in its center. The distance between the two adjacent injection boreholes of the hexagon is around 50 meters. Depending on the geological structure of the uranium ore deposit, the depth of the boreholes may vary from 50 to 1,000 meters. A *lixiviant*, leaching, diluted acidic solution, is introduced through the injection boreholes into the underground productive horizon, where uranium transfers into solution in about three months. The concentration of uranium is measured regularly, and the production starts when the concentration reaches a certain level.

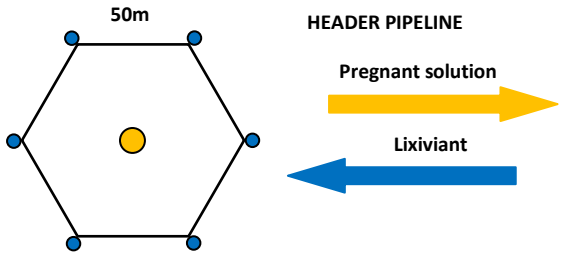


Figure 1.3: Hexagon in uranium mining.

During the production phase, the lixiviant that comes from the technological distribution unit is pumped through the injection boreholes to the underground productive horizon (see Figure 1.4). Then, the so called *pregnant solution* (with uranium) is pumped out from the production borehole and sent by the header pipeline to the processing plant, which is usually located relatively close to the technological field. Uranium is extracted from the pregnant solution at the processing plant, and then the *mother solution*, enriched by sulfuric acid, is sent back to the field at five or six atmospheres. Then, the whole process is repeated until all uranium is extracted, which makes the process cyclic.

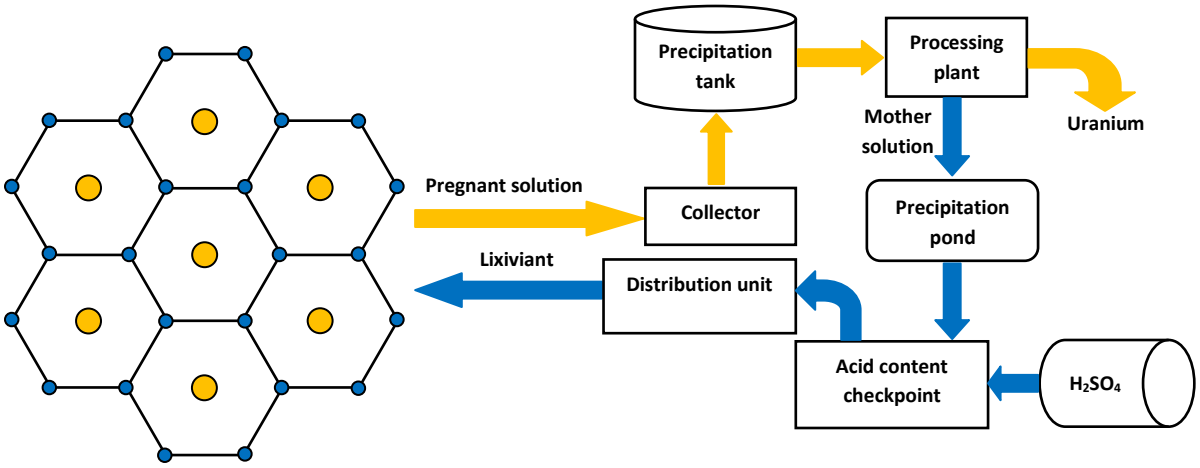


Figure 1.4: Borehole in situ leach mining layout (based on data at www.kazatomprom.kz).

It is very important to maintain the average concentration of uranium in the header pipeline coming to the processing unit above some threshold (e.g. 20mg/liter), so that the ore deposit produces the predetermined number of tons of uranium per year. For example, if a company has a 15-year license to develop an ore deposit with an assured 15,000 tons of uranium, then the company should extract 1,000 tons of uranium per year, on average. The concentration of uranium is usually controlled at the processing plant. When the concentration lessens, the reason for its reduction should be determined and the appropriate decision should be made. The reduction can be caused by any of the following:

a) the pipeline damage (9%),

b) the deep borehole filter contamination or clogging (90%),

c) the exhaustion of uranium (1%).

Often, the decisions are made solely at each ore deposit by their managers in an ad hoc manner. The main goal of the ore deposit manager is to maintain the concentration of uranium in the central pipeline (uranium output) as maximal as possible 24 hours a day, 7 days a week, while minimizing additional expenses, the negative environmental impact, and job hazard.

1.2.3 A pyramid of decision makers in uranium mining

If we construct a pyramid of decision makers in the uranium extraction industry as shown in Figure 1.5, then the deep borehole filter restoration problem would be at its processes level. The well-field manager has a primary responsibility for processes in the well-field such as which damage control method to choose for a particular borehole. However, the upper level decision makers in the pyramid can set up a corporate policy. Our intention is to study the decision frame or prospective of the decision makers at the processes level and at the corporate level to close any possible gap. This may ultimately help to supply the 14% shortage of uranium for nuclear power generation, which will hopefully improve the lives of every 5th person on this planet who lives today without access to electricity.

The ultimate decision maker of a transnational corporation determines the corporation's mission, vision, and values. The decisions are typically related to the portfolio of assets. The corporation's portfolio of assets in uranium mining includes geological survey expeditions (GSE), industrial complexes, and uranium ore deposits. The decision makers in GSE, for example, deal with various business units like processing plants, storage facilities, sets of drilling machines, and other producing assets. As an example, the business unit for extracting uranium is responsible for the budget and workforce allocation, time frame, safety of operations, and environmental protection. The decisions regarding the processes in the uranium extraction industry include the choice of the best borehole recovery method, among airflow, chemical, and other methods. Finally, processes include various operations. For example, the borehole construction operations include the drilling of a borehole, its calibration, ablution, and pumping. So, the successful construction of a borehole depends on correct and timely decisions about each operation.

There are many uncertainties (e.g. geological, environmental) even at the processes level and, therefore, the decisions are often difficult. In this work, we will study the decisions related to the choice of the damage control methods for the deep borehole filter restoration problem. Since, in 90% of the cases the uranium output reduction is caused by deep borehole filter contamination, its restoration is central. We define the deep borehole filter restoration problem in the next sub-section.

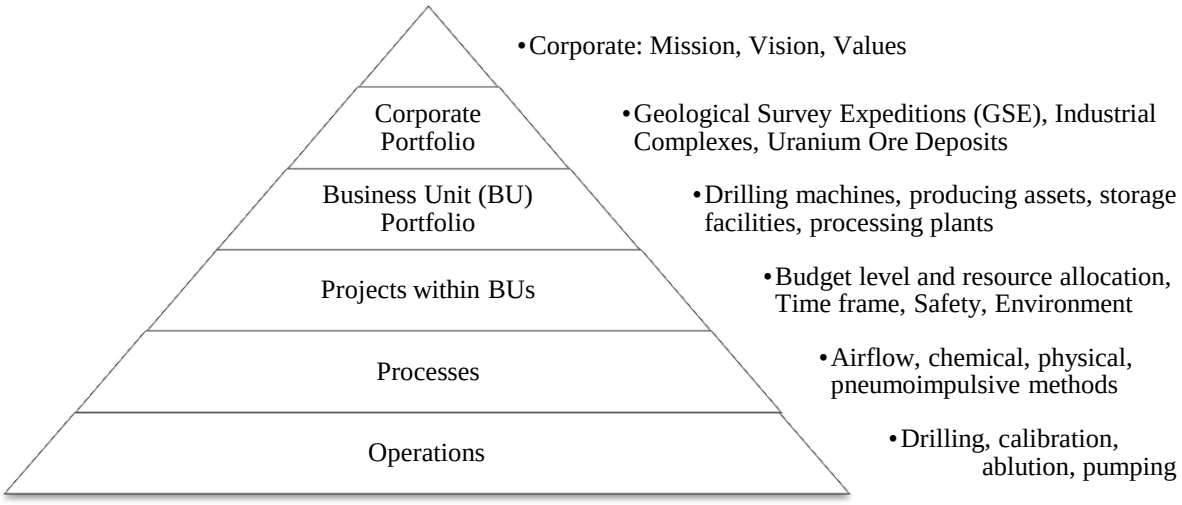


Figure 1.5: The pyramid of decision makers in the uranium extraction industry.

1.3 DEFINING THE DEEP BOREHOLE FILTER RESTORATION PROBLEM

In this section we will define and structure the deep borehole filter restoration problem in terms of objectives, attributes, alternatives, uncertainties, influence diagrams, and decision trees. In a related work, Keeney, Renn, and von Winterfeldt (1987) structure the former West Germany's energy supply decision problem by combining the objectives of multiple stakeholders together into "a value tree."

1.3.1 Description of the problem

The borehole recovery problem is fundamental in the uranium extraction industry. Failure or malfunctioning of even one borehole negatively affects whole process of uranium extraction by in situ leach (ISL) mining and requires active realignment of production routines. The deep borehole filter contamination or clogging can be caused by various factors. The bottleneck of the ISL method is due to "*colmatage du filtre*" (filter clogging from French). During the production phase, the debit of the borehole reduces over time due to chemical, mechanical, or gas formations in the filter zone. Sandy plugs, mud cakes, various mechanical and chemical sediments, suspended solid concentrations, casing and/or filter integrity damages, all are associated with one term, colmatage du filtre.

The well-field manager can choose one of four methods to clean the filter: airflow, chemical, physical, and pneumoimpulsive. However, none of these methods have clear dominance over the others. Sometimes, the manager may decide to re-drill the borehole, because they have already cleaned the filter recently. The DM may think that this spot has some unfortunate geological structure.

The interrepair period for damage control of operating characteristics could be initially from three to six months. This period is going to decrease with production time (i.e. age of the field under the development). Recent innovative technologies tremendously improved the performance of the production boreholes. For example, it has been experimentally proven that due to the reduction of the mechanical sediments in the productive solution from 100 to 50 milligrams/liter the service life of the submerged filters increased 1.5 times (Sushko, Begun, and Abdildin, 2008). However, the problem is still not solved. If the expected time of production at a particular ore deposit is five years, and the interrepair period is six months, then over time this

period will decrease. At some point it is not optimal anymore to do damage control. If the uranium output cannot be restored, then the borehole can be redrilled within this hexagon.

1.3.2 Alternatives

During my fieldwork with companies in the uranium extraction industry, the deep borehole filter restoration problem was discerned in consultation with the decision makers. As mentioned earlier, there are four damage control methods, but none of them have clear dominance over any other. There is an option of redrilling the borehole with guaranteed success, but at higher cost and longer time.

So, there are five *alternatives* that are used in practice:

1. Airflow method: air compressor is used to clean the deep borehole filter.

2. Chemical method: cleaning up the filter using (i) 10-20% solutions of hydrochloric acid, HCL, or (ii) 50-98% solutions of sulfuric acid, H_2SO_4.

3. Pneumoimpulsive method: cleaning up the filter using a pneumatic compressor up to 60 atm.

4. Physical method: washing up the filter zone with technical water.

5. Redrill: the borehole can be redrilled at a new location within the hexagon.

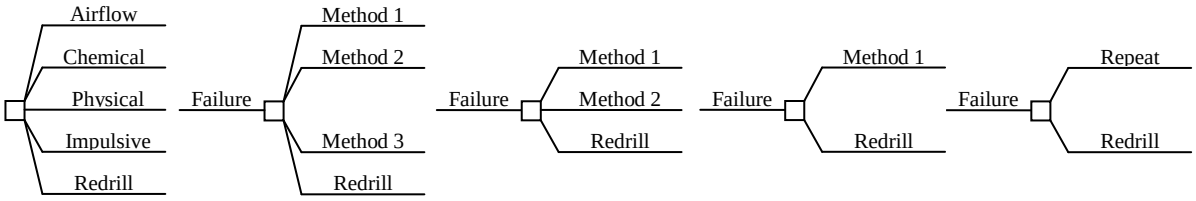


Figure 1.6: The partial decision tree for the deep borehole filter restoration problem

The DM can either apply some of the damage control methods sequentially, or redrill the

borehole at any step as shown in Figure 1.6. For example, if the manager chooses one of the methods and it does not work, then he can utilize one of the three untried methods or still redrill (as shown by a rectangular decision node of the partial decision tree). In other words, the DM encounters alternatives sequentially.

Some alternatives were excluded by the DM due to the technical difficulties of the methods. For example, one alternative is to change the filter. However, this method is too complex since the deep borehole filter is located at the very bottom of the casing strings, up to several hundred meters below the surface. Some other alternatives are rarely used and were not considered for comparison.

Each of the five alternatives above has a *probability of success,* which is the percentage of boreholes successfully recovered after utilizing the corresponding method. *Success* of the method in the partial decision tree, illustrated in Figure 1.7, implies non-zero uranium output that is above the predetermined level. The probabilities of success of the methods are defined as the following: airflow $(p_a = 0.56)$, chemical $(p_c = 0.65)$, pneumoimpulsive $(p_i = 0.62)$, and physical $(p_f = 0.59)$. Redrilling of the borehole guarantees success. In discussion with the DM, we found that the methods are technologically different from each other, so technically they are independent. That is, the failure of one method does not change the probability of success of any of the other methods.

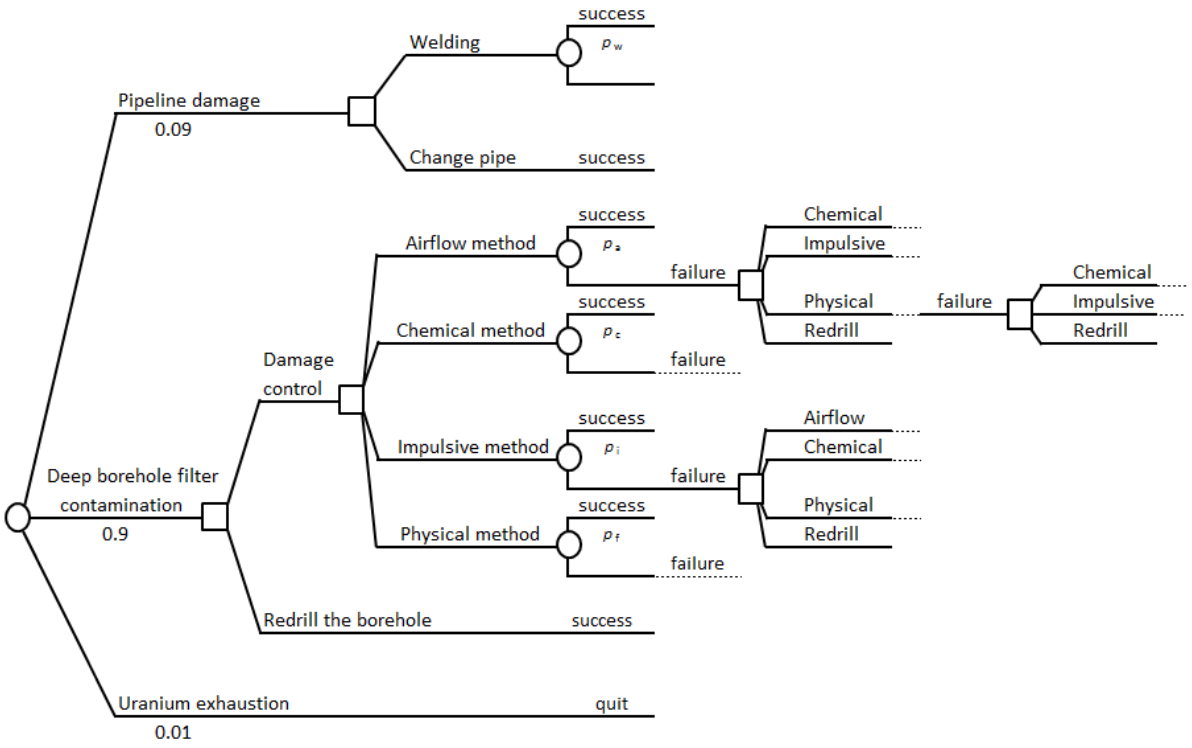


Figure 1.7: The partial decision tree constructed for the borehole restoration problem. The method is considered "successful" if uranium output is restored.

With what method should we start? If it does not work, which one should be used next? And so on. Should we first use the most efficient method, i.e. with the highest probability of success? Or the cheapest one? Or the fastest one? How about the safest method or the most environmentally friendly one? The best method can be found by utilizing normative methods. After the alternatives are ranked by the professional decision-making approach for one borehole, the company can make a new policy that would be applied to thousands of boreholes.

1.3.3 Objectives and attributes of the decision problem by the well-field manager

Clear objectives are vital in contemplating a choice of available alternatives, and the appropriate attributes can serve as measures of the objectives. The fundamental objective of the borehole restoration problem from the well-field manager's point of view is to restore the rated

output of uranium from a production borehole with minimal additional expenses and time, maximally avoiding adverse health effects on the personnel. This overall objective consists of the following sub-objectives:

- Increase *uranium output* from a production borehole
- Decrease *duration* of the restoration process
- Decrease *additional expenses* for cleaning the filter or redrilling the borehole
- Decrease the adverse *health effects* on the personnel.

The efficiency of damage control depends on many factors. The most important ones are presented in Table 1.3, where the ranges of attributes can vary within their most preferable (best) and least preferable (worst) values. For example, the attribute *Health effects of the process* (W) can vary between:

- $w^* = 1,$ *best value*, no adverse health effects on the personnel, and
- $w^0 = 5,$ *worst value*, moderate intoxication.

Table 1.3: Attributes for the borehole restoration problem, their measures and ranges.

Attribute	**Measure**	**Range**	
		best	worst
X - Uranium output from a borehole	U kg/hour	$x^* = 3.2$	$x^0 = 0.016$
Y - Restoration time (idle time)	hour	$y^* = 6$	$y^0 = 200$
Z - Additional expenses (for repair or redrilling)	$	$z^* = 73$	$z^0 = 23.3\text{K}$
W - Health effects of the process (job hazard)	1 – 5 scale	$w^* = 1$	$w^0 = 5$

1.3.4 Objectives and attributes of the decision problem by the ultimate decision maker

In consultation with the *ultimate decision maker*, Deputy Director General, we found that the the overall objective of the Company consists of the following main sub-objectives:

- Maximize *profit*
- Maximize *labor protection*
- Maximize *safety of operations*
- Maximize *preserved area.*

From the Deputy Director General's point of view, the four most important attributes for this decision problem are: *Profit*, *Labor protection*, *Safety of operations*, and *Preserved area*. We will discuss these attributes in detail later, for now we provide a brief overview.

Profit (X_1): This attribute refers to the annual net income from one production borehole. For the overall profit of the company, it must be multiplied by the hundreds of boreholes in operation possessed by the company.

Labor protection (X_2): Number of laborers unused and not exposed to radiation hazard for a damage control method compared with the redrill alternative. If redrilling a new borehole requires 12 employees (full crew), cleaning the filter by the airflow method will require only three employees (base level). Hence, utilizing the airflow method we could protect and not expose to radiation hazard nine employees. This is addressed by attribute X_2.

Safety of operations (X_3): The alternatives also differ in the number of consecutive days without an accident (technical, or technological). This is represented by attribute X_3.

Preserved area (X_4): The preserved area is calculated as the difference between the total area of the hexagon and the possible polluted area (with uranium/chemical spills) during the restoration process. The total area of the hexagon with a side length, $l = 50$ meters, is equal to:

$$Area = 6\frac{l^2\sqrt{3}}{4} \approx 6495 \text{ m}^2 \tag{1}$$

The ranges of each attribute are compactly represented in Table 1.4.

Table 1.4: Attributes for the borehole restoration problem, their measures and ranges (per production borehole or one hexagon).

Attribute	**Measure**	**Range**	
		best	worst
X_1 - Profit (annual) from one borehole	$	161,856	-70,866
X_2 - Labor protection	employees	10	0
X_3 - Safety of operations	days	1825	365
X_4 - Preserved area	m^2	6495	4000

1.3.5 Uncertainties

The annual profit of the company depends on several uncertainties as shown in Table 1.5.

Table 1.5: Attributes by the ultimate decision maker, related uncertainties, and measures.

Attribute	**Related uncertainties**	**Measure**	**Variable**
Profit (annual)		$	*P*
	Uranium output from a borehole	kg/hour	*Q*
	Concentration of uranium	mg/liter	*H*
	Flow rate of the borehole	m^3/hour	*F*
	Efficiency of the method	%	*E*
	Number of processes per year	integer	*N*
	Time of the process	hours	*T*
	Cost of the process	$	*C*
	Net income from 1kg of uranium	$	*I*
Labor protection	Number of laborers unused (and not exposed to hazard) for each method compared with the redrilling of a new borehole	employees	*L*
Safety of operations	Fulfillment of safety measures in the number of consecutive days without an accident (in 5-year period, i.e. in 1825 days)	days	*S*
Preserved area	Uncontaminated land of a hexagon area in square meters (out of 6495 m^2)	m^2	*A*

First of all, the uranium output from a production borehole depends on the concentration of uranium, volume of the pregnant solution from a borehole, and the "cleanliness" of the filter, which depends on the utilized method:

$$Q = H \times F \times E \tag{2}$$

A typical borehole with a clean filter has a flow rate of 10 m^3/hour that is around 10,000 liters/hour. The lowest acceptable value of the concentration of uranium is 20 mg/liter. That means that the lowest value of uranium output we are interested in is 0.2 kg/hour.

We assessed the Low, Base, High values (0.10, 0.50. 0.90 fractiles, respectively) of uranium output from the DM for each of the alternatives. These fractiles can be weighted in the decision tree by 25%, 50%, and 25%, respectively (Abbas, 2002b).

The decision maker was asked:

> "*If the applied method was successful, what do you believe would be the uranium output (kg/hour) at which there is a probability of 10% that the outcome will be below it?*" This is the Low value.

Next, we ask:

> "*If the applied method was successful, what do you believe would be the uranium output at which there is a probability of 90% that the outcome will be below it?*" This is the High value.

Finally:

> "*If the applied method was successful, what do you believe would be the uranium output at which there is a probability of 50% that the outcome will be below or above it?*" This is the Base value.

Then the annual profit can be found as:

$$P = (24 \times 365 - N \times T) \times Q \times I - N \times C \tag{3}$$

The equation in (3) shows that the annual profit is the difference between the income, $24 \times 365 \times Q \times I$, and the losses due to idle times, $N \times T \times Q \times I$, and the restoration processes,

$N \times C$.

Another important factor here is the net income (profit) per kilogram of uranium, *I*. It is also a random variable that varies from seven to 18 dollars per kilogram of yellowcake. Utilizing (2) and (3) we can model the annual profit of the company from the extraction borehole (i.e. one hexagon) to determine which method is economically more preferable. For this, we first assess from the decision maker the ranges (see Tables 1.6-1.10) and the Low, Base, High values of each of the uncertain variables related to each of the methods.

Table 1.6: Fractiles of uncertainty of the attributes for the Airflow method

# - Variable	**Measure**	**Values**				
		Min. Value	Low	Base	High	Max. Value
Q - Uranium output	kg/hour	0.016	0.2	0.6	1.2	1.4
N - Processes per year	integer	2	4	8	14	24
T - Process time	hours	6	8	16	48	72
C - Process cost	$	100	120	150	190	213
I - Profit per 1kg	$/kg	7	9	10	15	18
L - Labor protection	employees	8	8	9	10	10
S - Safety of operations	days	365	500	1000	1750	1825
A - Preserved area	m^2	4100	4350	5300	6450	6495

Table 1.7: Fractiles of uncertainty of the attributes for the Chemical method

# - Variable	**Measure**	**Values**				
		Min. Value	Low	Base	High	Max. Value
Q - Uranium output	kg/hour	0.016	0.2	0.6	1.2	1.8
N - Processes per year	integer	2	4	8	14	24
T - Process time	hours	12	48	72	84	96
C - Process cost	$	224	250	320	400	448
I - Profit per 1kg	$/kg	7	9	10	15	18
L - Labor protection	employees	6	6	7	8	8
S - Safety of operations	days	365	400	900	1500	1825
A - Preserved area	m^2	4000	4200	5200	6350	6495

Table 1.8: Fractiles of uncertainty of the attributes for the Pneumoimpulsive method

# - Variable	Measure	Values				
		Min. Value	Low	Base	High	Max. Value
Q - Uranium output	kg/hour	0.016	0.2	0.6	1.2	1.7
N - Processes per year	integer	2	4	8	14	24
T - Process time	hours	6	10	16	20	24
C - Process cost	$	233	260	300	345	373
I - Profit per 1kg	$/kg	7	9	10	15	18
L - Labor protection	employees	6	6	7	8	8
S - Safety of operations	days	365	400	1000	1600	1825
A - Preserved area	m^2	4000	4200	5300	6400	6495

Table 1.9: Fractiles of uncertainty of the attributes for the Physical method

# - Variable	Measure	Values				
		Min. Value	Low	Base	High	Max. Value
Q - Uranium output	kg/hour	0.016	0.2	0.6	1.2	1.6
N - Processes per year	integer	2	4	8	14	24
T - Process time	hours	6	8	10	11	12
C - Process cost	$	73	100	200	300	467
I - Profit per 1kg	$/kg	7	9	10	15	18
L - Labor protection	employees	8	8	9	10	10
S - Safety of operations	days	365	700	1300	1800	1825
A - Preserved area	m^2	4000	4200	5100	6100	6495

Table 1.10: Fractiles of uncertainty of the attributes for the Redrill alternative

# - Variable	Measure	Values				
		Min. Value	Low	Base	High	Max. Value
Q - Uranium output	kg/hour	0.5	0.6	1.1	1.8	3.2
N - Processes per year	integer	2	3	4	5	6
T - Process time	hours	120	144	156	180	200
C - Process cost	$	20000	21000	21666	22666	23333
I - Profit per 1kg	$/kg	7	9	10	15	18
L - Labor protection	employees	0	0	0	0	0
S - Safety of operations	days	365	600	1200	1650	1825
A - Preserved area	m^2	4000	4300	5300	6450	6495

1.3.6 Laymen ranking of the alternatives

As we can see from the data presented in Tables 1.6-1.10, it is not clear which method is the best one. If the manager in the well-field rank orders the methods on cost, time, and efficiency (probability of success), then he would probably rank them as shown in Table 1.11. *This laymen ranking shows no deterministic dominance of one alternative over any other*. For example, even though Redrill is the most efficient method, it is also the most expensive and time consuming alternative. Therefore, we need to utilize some normative method to choose the best alternative.

Table 1.11: Laymen ranking of the methods on cost, time, and efficiency

Method	**Rank**		
	Cheapest	Fastest	Efficient
Airflow	#1	#3	#5
Chemical	#3-4	#4	#2
Impulsive	#3-4	#2	#3
Physical	#2	#1	#4
Redrill	#5	#5	#1

1.3.7 The influence diagram

Decision trees and *influence diagrams* (Howard and Matheson, 1981; Howard and Matheson, 2005a) have been proven to simplify qualitative description and quantitative specification of the decision problems. See more discussions on influence diagrams in (Burns and Clemen, 1993; Boutilier, 2005; Buede, 2005; Detwarasiti and Shachter, 2005; Howard and Matheson, 2005b; Pauker and Wong, 2005) and the *knowledge maps* in (Howard, 1989). The multiattribute utility trees (Abbas, 2011b) help to decompose the problem into simple binary gambles.

An undirected graph may be "a perfect representation" for conditional additive independence (Bacchus and Grove, 1995). Directed (acyclic) graph representations are also popular. Boutilier et al. (1999) introduced *CP networks* to represent conditional preferential independence.

Boutilier, Bacchus, and Brafman (2001) extended CP-nets to *UCP networks* to address the generalized additive independence. Engel and Wellman (2008) proposed *CUI networks* to represent conditional UI and derive the functional form of a MUF. The *bidirectional utility diagrams* were presented (Abbas, 2009b; Abbas, 2010a) to simplify the graphical representation of partial utility independence conditions among the attributes of the decision problem.

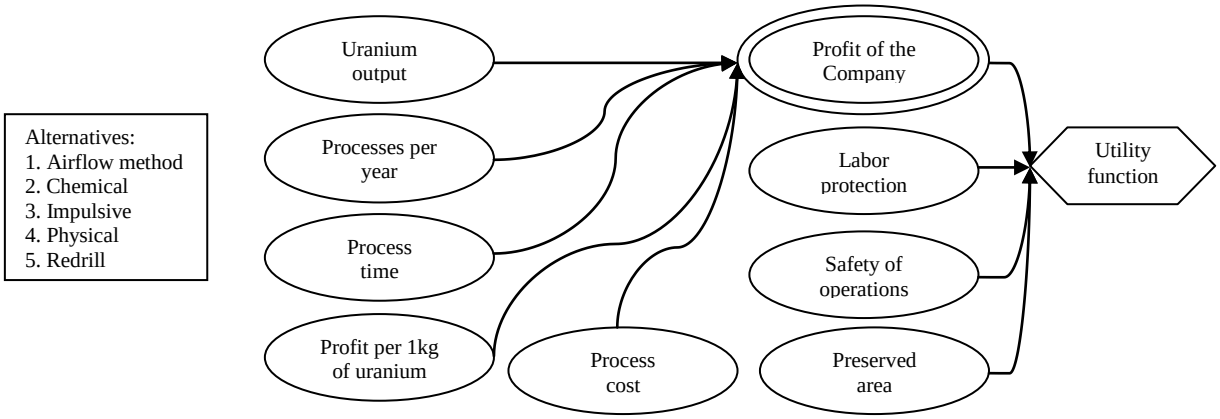


Figure 1.8: The influence diagram for the deep borehole filter restoration problem.

The influence diagram for the deep borehole filter restoration problem is illustrated in Figure 1.8. The influence diagram shows the alternatives in a rectangle, necessary calculations in a double oval node, the utility function in a hexagonal node, and influences with arrows. The decision influences all uncertainties, which in turn influence the outcome of the multiattribute utility function. There are five uncertainties influencing the Profit of the Company. We can calculate which of them contribute significantly (i.e. more than 90%) to the variance in the Profit, and then set others to their base level for easy of calculations.

1.3.8 The decision tree

The decision tree for constructing the multiattribute utility function is shown in Figure 1.9.

There are $729 = 9(3^4)$ outcomes in this decision tree. Our overall plan is to determine the best alternative by comparing the expected utility of each alternative. For this, we need to (i) assess the interdepdence among the four attributes, (ii) determine the functional form of the MUF based on those interdependencies, (iii) assess and construct the MUF, and (iv) choose the alternative with the highest expected utility.

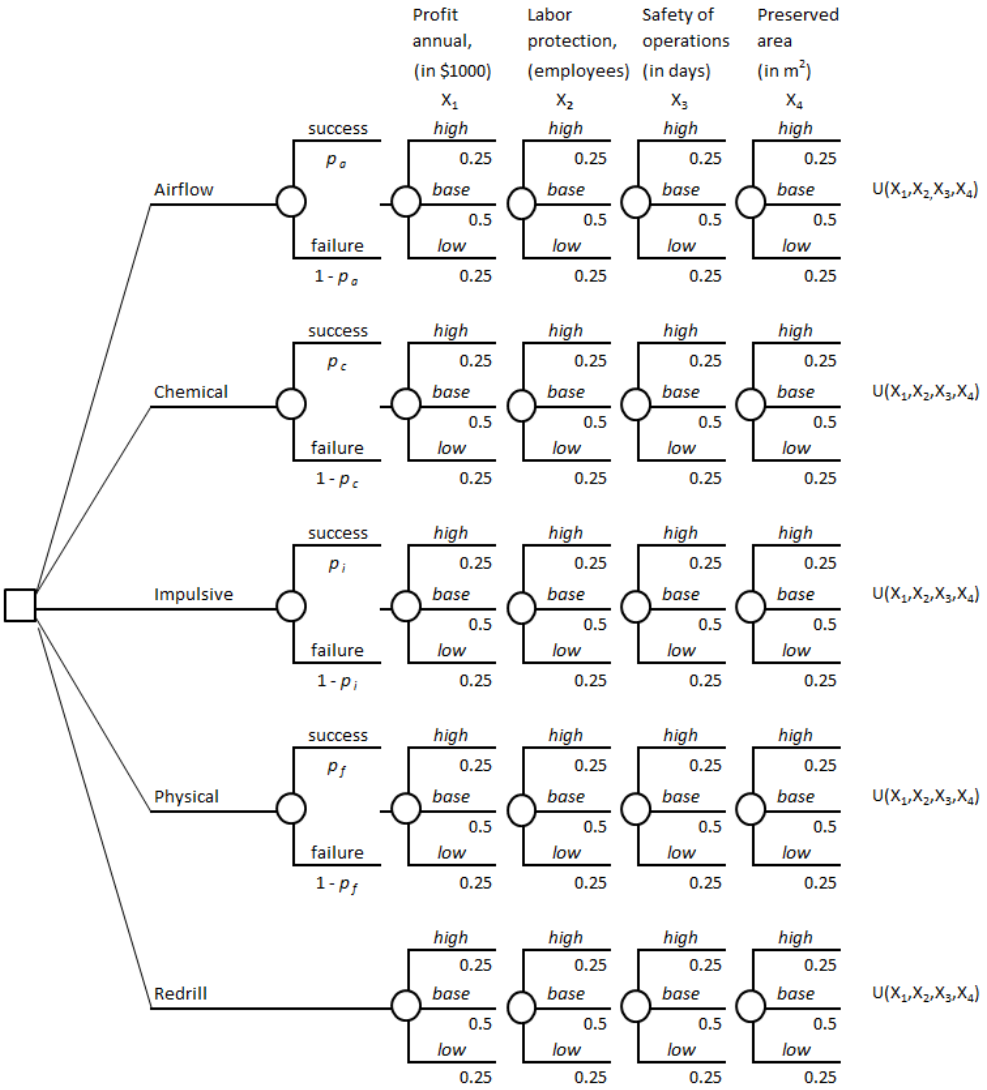


Figure 1.9: The partial decision tree constructed for the deep borehole filter restoration problem.

1.4 SUMMARY

In this Chapter we stated that the energy problem is one of the biggest challenges facing the world in the 21st century. With sufficient energy we can more easily address problems of food, water, poverty, and even climate change. Uranium is a good energy source since it does not contribute to carbon dioxide emission and has high energy density. We then discussed a decision problem that requires faster and more accurate solutions to address the 14% lack of uranium production for nuclear power generation. In particular, we showed in detail that the deep borehole filter restoration problem is a difficult sequential decision-making problem with multiple objectives and various uncertainties. We defined and structured the deep borehole filter restoration problem in terms of objectives, attributes, alternatives, uncertainties, influence diagrams, and decision trees. We discussed that complete independence of attributes may not exist in such complex decision problems and some partial independence or dependence may be present. To address such complex industrial decision problems, the analyst should properly capture the decision maker's preferences. This requires easy and accurate tests for assessing the decision maker's preferences and normative models for evaluating the existing alternatives.

CHAPTER 2: RELATED WORK, NOTATIONS AND DEFINITIONS

2.1 MULTIATTRIBUTE DECISION PROBLEMS: BACKGROUND

There are many problems in life in which people make decisions under various uncertainties and multiple objectives. Often, the complexity of such problems makes them difficult tasks to perform for decision makers (DM). Decision analysis (Howard, 1966) delivers normative methods (see, for example, Howard and Matheson, 1983; Howard, 1992; Abbas and Matheson, 2010; Abbas and Aczél, 2010) for formalizing decision problems. In decisions with uncertainties and multiple objectives, a multiattribute utility function, $U(x_1, x_2, ..., x_n)$, is required to represent trade-offs over the incommensurable attributes. Examples of attributes in an energy related decision problem could be profit of the company, safety of operations, and environmental impact.

However, the assessment of the multiattribute utility function (MUF) can be a challenging task, because the number of questions for the decision maker (DM) quickly grows with the number of attributes, *n.* The construction of a MUF is simplified if the attributes exhibit some forms of *utility independence* conditions (Keeney and Raiffa, 1976).

To illustrate this concept, consider a decision problem with two attributes, X and Y, where the DM is choosing between two risky projects A and B, while gambling on an attribute, X. Attribute X is utility independent (UI) of Y, if the DM's preference between the two projects does not change as the fixed value of Y varies. If X *UI* Y and Y *UI* X, then the attributes are *mutually UI* (Keeney and Raiffa, 1976), and the MUF is a combination of single-attribute utility

functions (Keeney and Raiffa, 1976). The UI approach is popular among practitioners for its simplicity (Dyer et al., 1998; Feng and Keller, 2006; Merrick and McLay, 2010).

What if mutual UI conditions do not hold? In some cases, some weaker *partial utility independence (PUI)* conditions are present, and so a *general decomposition* (Abbas, 2010a) allows us to decompose the MUF into lower-order terms, and construct it as a sum of products of (conditional) utility functions. A class of independence conditions that provides a simple decomposition with partial utility independence is referred to as the *canonical form* and was introduced in (Abbas, 2010a).

There are many methods for constructing a multiattribute utility function for a various degree of utility independence. For example, the *value-based* approach (Matheson and Abbas, 2005) does not require any utility independence assumptions, while the *multilinear* approach (Keeney and Raiffa, 1976) assumes that every attribute is utility independent of its complement. The *multiattribute utility copula* (Abbas, 2009c) is based on various utility independence conditions. The *double-sided utility copula* (Abbas, 2013) simplifies the construction of the multiattribute utility functions utilizing boundary assessments. Some methods are based on *attribute dominance conditions* (Abbas and Howard, 2005), *multiattribute transformation invariance* (Abbas, 2009a; Abbas, Aczél, and Chudziak, 2009; Abbas, 2007), *fractional hypercubes* (Farquhar, 1975), and *interpolation independence* (Bell, 1979a, 1979b). The *multiattribute utility tree* (Abbas, 2011b) was also presented to derive the functional form by decomposition into binary gambles. A review of these and other methods, which are based on the expected utility of von Neumann and Morgenstern (1947), can be found in (Abbas, 2010b).

The main purpose of this dissertation is a comprehensive analysis of functional forms of MUFs with PUI, verification of their applicability for complex multiattribute decision problems,

and development of novel methods for assessment of interdependencies among the attributes and determination of the utility assessments that are required for MUFs with PUI. This Chapter presents some work published in (Abdildin and Abbas, 2012; Abdildin and Abbas, 2013).

2.2 DECISION ANALYSIS IN INDUSTRIAL PROJECTS

The multiattribute utility theory is an essential part of decision making in industrial projects. Vice Chairman of Chevron George Kirkland says that the Company uses decision analysis for the last 20 years and "decision analysis helps them to manage risks with the right level of analysis" (Kirkland, 2010). Webster (2008) also found that decision analysis is "an appropriate choice of analytical method" for decisions in global climate-change policy. In fact, decision analysis (DA) has been applied in a broad number of industrial problems such as, for example, nuclear power plant siting (Gros, 1975), determining a value of life under nuclear risk (Owen, Matheson, and Howard, 1978), nuclear waste repository siting (Merkhofer and Keeney, 1987; Miklas et al., 1995), resource allocation in the oil and gas industry (Keefer, 1991), disposition of surplus weapons-grade plutonium (Dyer at al., 1998), potassium iodide distribution in nuclear incidents (Feng and Keller, 2006), evaluation of countermeasures to secure cargo at US ports of entry (Bakır, 2008), screening cargo containers for smuggled nuclear threats (Merrick and McLay, 2010), an analysis of nuclear counter proliferation (Caswell, Howard, and Paté-Cornell, 2011), and many other decision problems (Howard and Matheson, 1983; Keeney and Raiffa, 1976).

The general trend is that practitioners apply the *additive form* (§2.3.5) of a utility independence approach (Keeney and Raiffa, 1976), where 'every subset of the attributes is assumed to be UI of its complement.' However, this strong assumption may have dramatic

consequences if not carefully verified, because not all decision problems may comply with utility independence assumptions (Farquhar, 1975; Kirkwood, 1976; Fishburn, 1977; Bell, 1979a; Keeney, 1981; Tamura and Nakamura, 1983; Harvey, 1993; Abbas and Howard, 2005; Abdildin and Abbas, 2013). Recall, the Deepwater Horizon explosion in the Gulf of Mexico in 2010 caused catastrophic ecological damage. Although we cannot relate this event directly to some faults in decision processes, we should analyze the decisions in the energy sector to prevent future explosions. For example, if we consider three attributes for an evaluation of two exploration projects as *profit* (X_1), *safety of operations and personnel* (X_2), and *environmental protection* (X_3), then attribute X_2 is not utility independent from X_1 if the DM's preferences for uncertain deals on X_2, given the fixed value of X_3, changes with the value of X_1. In this case, the DM may assert the partial utility independence. Therefore, the assumption of UI may result in an imprecise MUF if utility independence conditions are not assessed carefully. Many important decisions (Keeney, 2004) in industrial projects are complex by nature and require normative analysis. Simple target-based approaches when organizations set some thresholds on various parameters of interest (e.g. tons of uranium produced, quarterly earnings per share) may lead to suboptimal decisions (Abbas, Matheson, and Bordley, 2009).

2.3 REVIEW OF THE BASIC CONCEPTS AND DEFINITIONS

Throughout this dissertation, capital letters denote the attributes as X_i, $i = 1,...,n$, and small letters indicate their instantiations, x_i. Also, x_i^0 and x_i^* will represent the *least* and *most* preferred values of attribute X_i, respectively. $\bar{X}_i$ will mean the complement of X_i, the set of remaining attributes. The multiattribute utility function $U(x_1,...,x_n)$ will be normalized between

$U(x_1^0,...,x_n^0)=0$ and $U(x_1^*,...,x_n^*)=1$. Finally, $U(x_i \mid \bar{x}_i)$ will designate a *normalized conditional utility function*

$$U(x_i \mid \bar{x}_i)=\frac{U(x_i,\bar{x}_i)-U(x_i^0,\bar{x}_i)}{U(x_i^*,\bar{x}_i)-U(x_i^0,\bar{x}_i)}, \tag{4}$$

and $\bar{U}(x_i \mid \bar{x}_i)=1-U(x_i \mid \bar{x}_i)$, the normalized conditional *disutility* function.

The order of utility assessments needed for the MUF increases with the number of attributes, *n.* In particular, if the attributes of a decision problem do not provide utility independence conditions, then the order of utility assessments is equal to *n.* If there exist "mutual utility independence" (Keeney and Raiffa, 1976) or even "partial utility independence" conditions (Abbas, 2010a), then the order of the required utility assessments can be dramatically reduced.

Definition 1: Utility Independence (Abbas, 2010a). *A set of attributes X_K is utility independent (UI) of another set X_I given X_D, written $(X_K \ UI \ X_I \mid X_D)$, with $X_I \cup X_D = \bar{X}_K$, if preferences for joint lotteries over X_K do not depend on the instantiations of X_I.*

As an example, think about a decision problem with three attributes, X,Y,Z. Then, $(X \ UI \ Y \mid Z)$ if the DM's preference over gambling on X does not change as the fixed value of Y varies, given that the values of Z in those gambles are identical. Thus, we can substitute the instantiation of the attribute Y, denoted y, with its least preferred, y^0, (or its most preferred, y^*) value in a normalized conditional utility function, $U(x \mid y,z)=U(x \mid y^0,z)$, decreasing the order (size) of the assessment from three to two. Here, $U(x \mid y^0,z)$ requires two-dimensional assessments of (X,Z) when the attribute Y is fixed at instantiation y^0.

There are different forms of PUI. Two additional forms will be utilized in this work: "boundary independence" (Abbas, 2011b) and "one-switch independence" (Abbas and Bell,

2012; Abbas and Bell, 2011; Bell and Fishburn, 2001; Bell, 1988; Chudziak, 2012; Abbas and Chudziak, 2013).

Definition 2: Boundary Independence (Abbas, 2011b). *Attribute X is boundary independent of another attribute Y, written X BI Y, if $U(x \mid y^0) = U(x \mid y^*)$.*

As an example of boundary independence, consider the case when the DM's preference to gamble on X changes when the fixed value of Y is different from its boundary values.

Definition 3: One-Switch Independence (Abbas and Bell, 2011). *Attribute X exhibits one-switch independence from attribute Y, written X 1S Y, if preference between any pair of gambles on X can switch at most once as the level of Y increases.*

One-switch independence may occur if the DM's preference may switch at most once at some threshold of attribute Y. We will show examples of these partial UI conditions in the following Chapters.

The partial utility independence condition implies that the normalized conditional utility function $U(x_K \mid x_I, x_D) = U(x_K \mid x_I^0, x_D)$, and it also corresponds to the following functional form

$$U(x_K, x_I, x_D) = U(x_K^0, x_I, x_D) + [U(x_K^*, x_I, x_D) - U(x_K^0, x_I, x_D)]U(x_K \mid x_I^0, x_D), \qquad (5)$$

where $U(x_K \mid \bar{x}_K)$ is a normalized conditional utility function for a set of attributes X_K; x_K^0 and x_K^* are the least and most preferred values of all the attributes in X_K (respectively), and x_K represents an instantiation of X_K.

To illustrate, consider a situation with four attributes. The condition $(X_3\ UI\ X_4 \mid X_1, X_2)$ implies that $U(x_3 \mid x_1, x_2, x_4) = U(x_3 \mid x_1, x_2, x_4^0)$, and so

$$U(x_1, x_2, x_3, x_4) = U(x_1, x_2, x_3^0, x_4) + [U(x_1, x_2, x_3^*, x_4) - U(x_1, x_2, x_3^0, x_4)]U(x_3 \mid x_1, x_2, x_4^0). \quad (6)$$

This functional form requires three different utility assessments: $U(x_1, x_2, x_3^0, x_4)$, $U(x_1, x_2, x_3^*, x_4)$, and $U(x_3 \mid x_1, x_2, x_4^0)$. For example, $U(x_1, x_2, x_3^0, x_4)$ implies assessments of the attributes X_1, X_2, and X_4 on their entire domains when the attribute X_3 is fixed at instantiation x_3^0. Similarly, $U(x_1, x_2, x_3^*, x_4)$ requires utility assessments of size three when X_3 is fixed at instantiation x_3^*. The normalized conditional utility function, $U(x_3 \mid x_1, x_2, x_4^0)$, needs assessments of X_1, X_2, and X_3 when the attribute X_4 is set arbitrarily to the instantiation, x_4^0. As we can see from this example, when partial UI conditions exist, we can reduce the order of utility assessments of size four, $U(x_1, x_2, x_3, x_4)$, to three assessments of size three. Notice, there are two identical terms $U(x_1, x_2, x_3^0, x_4)$; hence, one of them is a *duplicate* and should be excluded from the list of the required utility assessments.

The problem of determination of the minimal set of utility assessments required for a construction of a MUF with partial utility independence is relatively new and has not been addressed by decision analysis society. Therefore, another goal of this dissertation is to develop novel algorithms for excluding duplicate and redundant assessments in a multiattribute decision problem with partial utility independence.

2.3.1 Review of the Utility Dependence Matrix

Utility interdependence conditions among the attributes can be represented in the form of a "*utility dependence matrix*" (Abbas, 2010a). The "utility dependence matrix is an $n \times n$ incidence matrix," which represents pairwise utility dependence assertions among n attributes. A vacant cell in row i and column j of utility dependence matrix (UDM), X_{ij}, asserts the utility independence relation of X_i from X_j, while a non-vacant cell does not assert any independence

relation between X_i and X_j (Abbas, 2010a). Consider the UDM displayed in Figure 2.1. The matrix asserts that attributes X_1 and X_3 both are utility independent from the attribute X_2. The representation of utility independence assertions among the attributes of a decision problem in the form of utility dependence matrix stimulates a further development of compact matrix-based algorithms for multiattribute decision problems.

	X_1	X_2	X_3
X_1	○		○
X_2	○	○	○
X_3	○		○

Figure 2.1: Utility dependence matrix for Example 1.

2.3.2 Review of the Basic Expansion Theorem

The condition of UI enables the reduction of the order of utility assessments required for the construction of the MUF. Such reduction would significantly minimize the order of assessments needed from the decision maker. The iterative decomposition algorithm (Abbas, 2010a) addresses this problem by expanding attributes, which assert partial UI conditions, in the form of a tree. As shown in (Abbas, 2010a), the expansion of the MUF through one attribute, say x, leads to a functional form

$$U(x,\bar{x}) = U(x^*,\bar{x})U(x\,|\,\bar{x}) + U(x^0,\bar{x})\bar{U}(x\,|\,\bar{x}), \tag{7}$$

where $\bar{U}(x_K\,|\,\bar{x}_K) = 1 - U(x_K\,|\,\bar{x}_K)$ is a normalized conditional disutility function (Abbas, 2010a). A further expansion through the second attribute, y, results in

$$U(x,y,\overline{xy}) = U(x^*,y^*,\overline{xy})U(x\mid\bar{x})U(y\mid x^*,\overline{xy}) + U(x^*,y^0,\overline{xy})U(x\mid\bar{x})\bar{U}(y\mid x^*,\overline{xy}) + U(x^0,y^*,\overline{xy})\bar{U}(x\mid\bar{x})U(y\mid x^0,\overline{xy}) + U(x^0,y^0,\overline{xy})\bar{U}(x\mid\bar{x})\bar{U}(y\mid x^0,\overline{xy}). \quad (8)$$

By induction, additional expansions of the MUF generate the *basic expansion Theorem* (Abbas, 2010a):

$$U(x_1,x_2,...,x_n) = \sum_{x_K^{*0}\in X_K^{*0}} U(x_K^{*0},\bar{x}_K)\prod_{i\in K} g(x_i\mid x_{iP}^{*0},x_{iF}), \quad (9)$$

where X_K is the set of attributes that are expanded, and $g(\cdot)$ is either a conditional utility or disutility function depending on the instantiations of x_i in the terms $U(x_K^{*0},\bar{x}_K)$; X_{iP} symbolizes the set of attributes expanded prior to X_i and, X_{iF}, represents the set of attributes to be expanded after X_i, such that $X_{iP}\cup X_{iF} = \bar{X}_i$.

The basic expansion Theorem (BET) has a significant contribution to the field of decision analysis since it proves that any utility function that is (i) continuous, (ii) bounded between zero and one, and (iii) non-decreasing with each of its arguments can be expanded in terms of normalized conditional utility assessments as in (9).

For the UDM in Figure 2.1, the BET results in the following multiattribute utility function

$$\begin{aligned} U(x_1,x_2,x_3) = {} & U(x_1^*,x_2,x_3^*)U(x_1\mid x_2^0,x_3)U(x_3\mid x_1^*,x_2^0) \\ & + U(x_1^0,x_2,x_3^*)\bar{U}(x_1\mid x_2^0,x_3)U(x_3\mid x_1^*,x_2^0) \\ & + U(x_1^*,x_2,x_3^0)U(x_1\mid x_2^0,x_3)\bar{U}(x_3\mid x_1^0,x_2^0) \\ & + U(x_1^0,x_2,x_3^0)\bar{U}(x_1\mid x_2^0,x_3)\bar{U}(x_3\mid x_1^0,x_2^0). \end{aligned} \quad (10)$$

This multiattribute utility function requires assessments of four one-dimensional utility assessments $U(x_1^*,x_2,x_3^*), U(x_1^0,x_2,x_3^*), U(x_1^*,x_2,x_3^0),\ U(x_1^0,x_2,x_3^0)$, and two conditional utility assessments, $U(x_1\mid x_2^0,x_3), U(x_3\mid x_1,x_2^0)$, since $\bar{U}(\cdot) = 1-U(\cdot)$.

2.3.3 Review of the Iterative Decomposition

Abbas (2010a) presented the "*iterative decomposition algorithm"* for a multiattribute utility function. This algorithm simplifies the determination of the requisite utility assessments. The iterative decomposition of the multiattribute utility function for the decision problem of Example 1 is illustrated in Figure 2.2.

The expansion of the iterative decomposition algorithm in Figure 2.2 is done in the order $X_1, X_3,$ and the required utility assessments are shown in the bold rectangles. Recall, these are the assessments which are necessary for the construction of a multiattribute utility function of Example 1 as shown in equation (10).

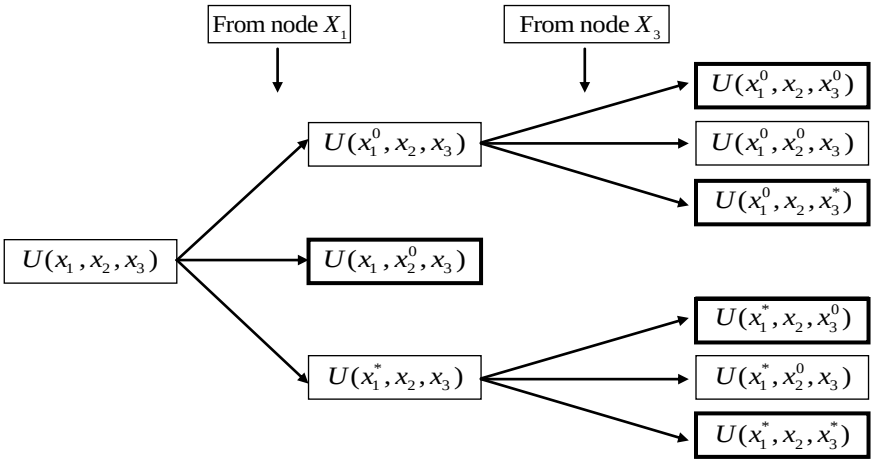


Figure 2.2: The iterative decomposition of the MUF for Example 1.

The two conditional utility assessments, $U(x_1 \mid x_2^0, x_3), U(x_3 \mid x_1, x_2^0),$ correspond to $U(x_1, x_2^0, x_3)$. The middle term in the first branch in the tree, $U(x_1, x_2^0, x_3)$ incorporates UI conditions asserted by the expanded attribute, X_1. The other two terms, $U(x_1^0, x_2, x_3)$ and $U(x_1^*, x_2, x_3),$ were expanded from $U(x_1, x_2, x_3)$ by replacing the expanding attribute X_1 into its

least and most preferable instantiations, respectively. By induction, the further expansion of these three terms incorporates the UI conditions provided by the attribute X_3.

It is necessary to have an algorithm that can automate the detection of all the required utility assessments discussed above. Such algorithms will be presented in Chapters 6 and 7.

2.3.4 Review of Utility Independence

This sub-section reviews widely used forms of multiattribute utility functions corresponding to the utility independence approach (Keeney and Raiffa, 1976).

Definition 4: Multilinear Form (Keeney and Raiffa, 1976).

If every attribute, $X_i,\ i=1,...,n,\ (n\geq 2)$ *is UI of* $\bar{X}_i$, *then the MUF takes the multilinear form:*

$$U(x_1,...,x_n)=\sum_{i=1}^{n}k_iU_i(x_i)+\sum_{i=1}^{n}\sum_{j>i}k_{ij}U_i(x_i)U_j(x_j) + \sum_{i=1}^{n}\sum_{j>i}\sum_{l>j}k_{ijl}U_i(x_i)U_j(x_j)U_l(x_l)+...+k_{123...n}U_1(x_1)U_2(x_2)\cdots U_n(x_n), \tag{11}$$

where

1. $U_i(x_i)$ *is a conditional utility function on* X_i *normalized by* $U_i(x_i^0)=0$ *and* $U_i(x_i^*)=1$,
2. *the scaling constants*

$$k_i=U(x_i^*,\bar{x}_i^0),$$
$$k_{ij}=U(x_i^*,x_j^*,\bar{x}_{ij}^0)-k_i-k_j,$$
$$k_{ijl}=U(x_i^*,x_j^*,x_l^*,\bar{x}_{ijl}^0)-k_{ij}-k_{il}-k_{jl}-k_i-k_j-k_l,$$
$$k_{123...n}=1-\sum_i k_{1...(i-1)(i+1)...n}-\cdots-\sum_{i,j>i}k_{ij}-\sum_i k_i,\ i<j<h,\ i=1,...,n.$$

To illustrate, consider a decision problem with three attributes, $X, Y,$ and Z. If each of these attributes are UI of their respective complements, then

$$U(x,y,z)=k_xU_x(x)+k_yU_y(y)+k_zU_z(z)+k_{xy}U_x(x)U_y(y)+k_{xz}U_x(x)U_z(z)$$

$$+k_{yz}U_y(y)U_z(z)+k_{xyz}U_x(x)U_y(y)U_z(z),$$

where the constants k_x, k_y and k_{xy}, for instance, are the following: $k_x=U(x^*,y^0,z^0), k_y=U(x^0,y^*,z^0),$ and $k_{xy}=U(x^*,y^*,z^0)-k_x-k_y$. The constant k_{xyz} can be

found by subtracting the other six constants from one. The multilinear form is applicable for a UDM in in the form of an identity matrix. The UDM corresponding to (11) is an identity matrix.

This multilinear form of single-attribute functions can have even simpler special cases when the attributes are mutually utility independent.

Definition 5: Mutual Utility Independence (Keeney and Raiffa, 1976).

If every subset of the attributes is UI of its complement, then the attributes are mutually utility independent.

In this case, the MUF can take either the *multiplicative form* (Keeney and Raiffa, 1976),

$$1+kU(x_1,...,x_n)=\prod_{i=1}^{n}[1+kk_iU_i(x_i)], \tag{12}$$

or, the *additive form* (Keeney and Raiffa, 1976),

$$U(x_1,...,x_n)=\sum_{i=1}^{n}k_iU_i(x_i), \quad \sum_{i=1}^{n}k_i=1 \tag{13}$$

2.3.5 Review of the Canonical Form

Due to the diversity of decision problems, the assumption of utility independence may not always hold; the attributes may have partial utility independence. One special case of partial utility independence is called the "canonical form" (Abbas, 2010a).

Let r be the number of *incomplete rows* of UDM, the rows with at least one vacant cell. In other words, r is the cardinality of the set of attributes that provide at least one independence assertion, X_K. Let i and j be the indices of cells of UDM.

Definition 6: Symmetric pair (Abdildin and Abbas, 2013). *A pair of cells* (i, j) *and* (j,i) *of the UDM constitutes a symmetric pair*.

Definition 7: Canonical UDM (Abbas, 2010a). *A utility dependence matrix is said to be in a canonical form if one of these conditions holds:*

(1) r = 0 or 1

(2) if $r \geq 2$, all symmetric pairs in the incomplete rows of the UDM are vacant.

For the UDM in Fig. 2.4(a), $X_K = \{X_2, X_3\}$, $r = 2$, so we check only symmetric cells $(2,3)$ and $(3,2)$, which are vacant; hence, the UDM is canonical. The UDM in Fig. 2.4(b) is non-canonical, because $X_K = \{X_1, X_2, X_3\}$, $r = 3$, and the first symmetric pair $(1,2)$ and $(2,1)$ is non-vacant. The multiattribute utility function, corresponding to the canonical UDM, is said to be a *canonical multiattribute utility function*.

	X_1	X_2	X_3
X_1	○	○	○
X_2		○	
X_3			○

	X_1	X_2	X_3
X_1	○	○	
X_2		○	○
X_3		○	○

Figure 2.3. Examples of the utility dependence matrix for three attributes: (a) canonical UDM, and (b) non-canonical UDM.

2.4 DISSERTATION CONTRIBUTION AND OVERVIEW

Motivated by the complexity of multiattribute decision problems under uncertainty and multiple objectives, this dissertation considers a set of open research problems related to the number of attributes and their degree of utility dependence and independence. This research determines the number and convergence of the canonical functional forms of multiattribute utility functions, introduces proper partial utility independence conditions, verifies their

applicability to the deep borehole filter restoration problem, evaluates the alternatives of the decision problem by three different approaches, and introduces novel methods for excluding duplicate and redundant assessments from the assessments required for the construction of MUFs with PUI.

The structure of the dissertation is as follows.

Chapter 1 defines the deep borehole filter restoration problem, a fundamental problem in the uranium extraction industry, in terms of: objectives, attributes, alternatives, uncertainties, and decision trees. It also discusses the role of uranium as an energy source in solving the world's big challenge, the Energy problem.

Chapter 2 reviews the related work and defines the notations and definitions.

The profit analysis of the deep borehole filter restoration problem is provided in Chapter 3. If only profit is considered, then the alternatives can be easily rank-ordered by the utility function of attribute profit assessed from the decision maker. But what if more than one attribute is of interest for the decision maker?

Chapter 4 describes new tests for assessing the decision maker's preferences among several attributes and the experimental study of interdependence of the attributes of the deep borehole filter restoration problem. Conducting two experiments with large numbers of professionals in uranium mining, we determine with 95% confidence that the proportion of professionals in the population who assert (i) partial utility independence of the attributes is at least 0.91, and (ii) utility independence of the attributes is at most 0.23. The results demonstrate that the assumption of utility independence may not be valid for this decision problem.

Chapter 5 provides the multiattribute analysis of the deep borehole filter restoration problem among the four most important attributes from the ultimate decision maker's, Deputy Director

General's, point of view. The multiattribute analysis of this decision problem is also conducted under the utility independence (UI) assumption to study how well UI approach can approximate the multiattribute analysis with the partial utility independence (PUI) conditions. The results demonstrate that the decisions may be inconsistent in 40% of the times if only profit is considered or if utility independence is assumed compared with the true preferences of the decision maker.

In complex decision problems some utility assessments in the functional form may be redundant or duplicate. Redundant utility assessments imply redundant questions to the decision maker and must be avoided. The elimination of these utility assessments can be a tall order. Chapter 6 presents a ternary matrix representation of utility assessments and the algorithm for excluding redundant assessments. Using the ternary matrix of utility assessments as an input, the Twos-Complement Exclusion algorithm efficiently eliminates all duplicate and redundant utility assessments. It can also serve for a consistency check.

Chapter 7 introduces the Ternary-Decimal Exclusion algorithm that outperforms the Twos-Complement Exclusion algorithm.

The results of this dissertation were applied to an important and complex decision problem in the uranium extraction industry and may help to address the worldwide 14% shortage of uranium for nuclear power generation.

CHAPTER 3: PROFIT ANALYSIS OF THE ALTERNATIVES OF THE DEEP BOREHOLE FILTER RESTORATION PROBLEM

In this Chapter the alternatives of the deep borehole filter restoration problem are analyzed based on one attribute, profit of the company. First, the deterministic analysis is conducted, then the one-attribute utility analysis is provided and the alternatives are rank-ordered.

3.1 DETERMINISTIC ANALYSIS

3.1.1 A value model

We first conduct deterministic analysis with sensitivity to ranges of variables in (2) and (3). We develop a value model in order to calculate net income for each of the alternatives for Low, Base, High cases from Tables 1.6-1.10 and plot a bar chart as shown in Figure 3.1.

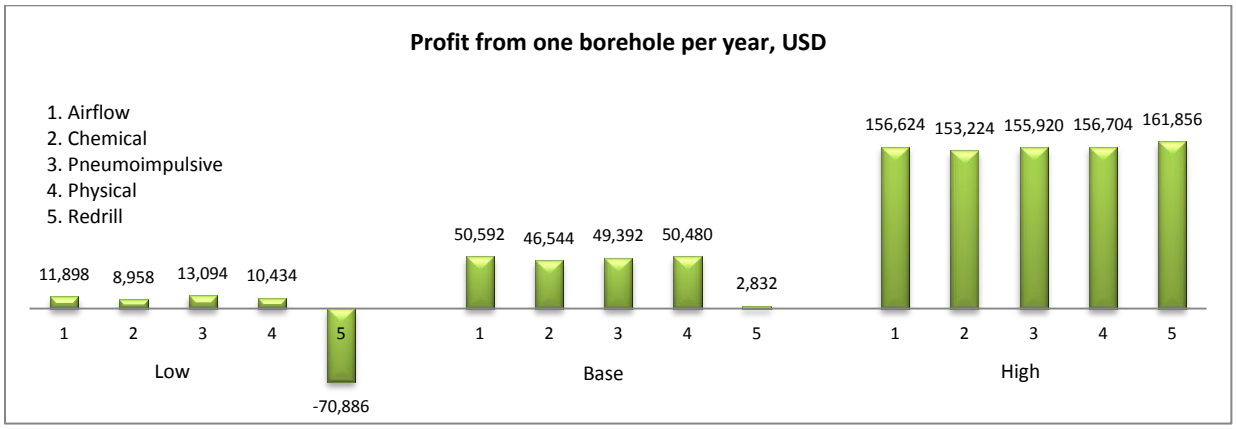


Figure 3.1: Profit for Low-Base-High cases for each alternative.

For example, the five bars on the right hand side (RHS) of the bar chart (High values) were

calculated by setting all five variables (uncertainties) to their High values, namely, *Uranium output* (kg/hour) and *Profit per 1 kg of uranium* ($/kg) to their High values, and the other three variables to their Low values since their Low values correspond to High values of profit and vice versa. It is clear from Figure 3.1 that the 5th alternative, Redrill, is worse than the rest when all variables are at low levels (Low case) and when all variables are at base levels (Base case), but when all variables are high (High case), it is the most preferable. The four other alternatives are relatively comparable in all three cases, so the spread sheet analysis does not provide us the "full picture;" in other words, there is no deterministic dominance of one alternative over others.

3.1.2 Sensitivity analysis

We now conduct sensitivity analysis for each of the methods utilizing data in Tables 1.6-1.10 to determine the variables (uncertainties), which contribute most to the variance of the annual profit of the Company from one borehole. The *tornado diagrams* (Howard, 1988) can help us to determine the variables, which are worth modeling probabilistically. In a related work, Eppel and von Winterfeldt (2008) construct tornado diagrams for a nuclear waste storage tanks problem. Felli and Hazen (2004) introduce *javelin diagrams*, an analog of tornado diagrams for probabilistic sensitivity analysis.

Figure 3.2 illustrates that two variables, *Uranium output* (kg/hour) and *Profit per 1 kg of uranium* ($/kg), contribute to more than 90% of variance when the Airflow method is applied for the deep borehole filter restoration problem. This was found true for other alternatives too. Notice, other three variables from Tables 1.6-1.10 contribute negatively to the annual profit of the Company, i.e. Low values correspond to High values of profit and vice versa, which were also taken into consideration when the diagrams were constructed.

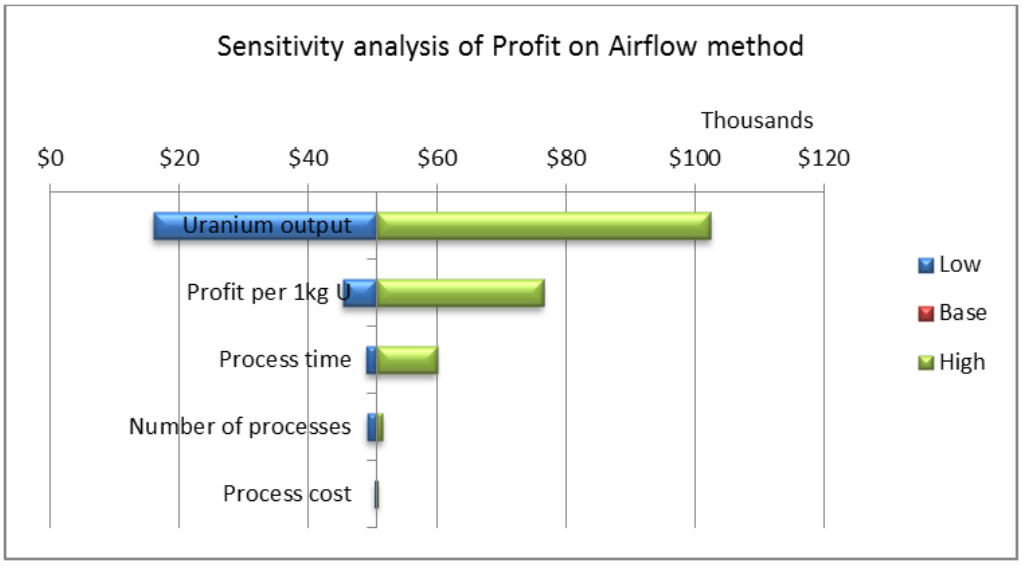


Figure 3.2: Sensitivity analysis of an annual profit from one borehole when Airflow method is applied for the deep borehole filter restoration problem.

3.1.3 Cumulative Distribution Functions of each alternative

Next, we set the other three variables (number of processes, process cost, and process time) to their Base values and construct cumulative distribution functions (CDFs) for each of the alternatives. In a related work, DeWispelare, Herren, and Clemen (1995) discuss the CDF elicitation in a high-level nuclear waste regulation program. A good review and comparison of probability encoding methods can be found here: (Abbas, Budescu, Yu, and Haggerty, 2008; Budescu, Abbas, and Wu, 2011). Note that CDFs in Figure 3.3 do not present the deterministic dominance of one alternative over all others. We just see that the redrill alternative is dominated by the other alternatives, because the probability of lower profit (i.e. loss) is higher compared to other alternatives. Therefore, we need to assess the utility function of profit and determine the best alternative that has the highest expected utility.

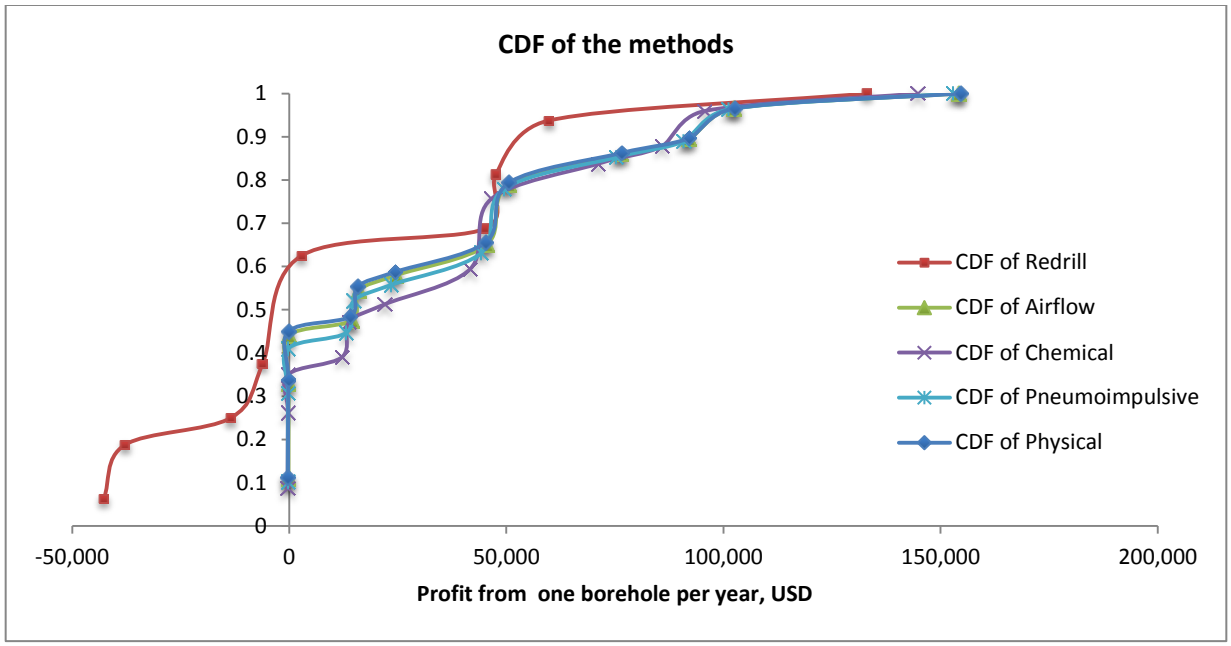


Figure 3.3: CDFs of the annual profit of the Company from one borehole.

3.2 UTILITY ANALYSIS BASED ON ATTRIBUTE PROFIT

3.2.1 Assessing utility function of Profit

We need to incorporate the decision maker's risk attitude by assessing the utility function over the profit. Our decision maker is *risk averse* (see Arrow, 1965; Pratt, 1964; Keeney and Raiffa, 1976; Ross, 1981; Abbas, 2011a; Abbas, 2012) since they prefer x_1 to 50-50 lottery with $< x_1 + d, x_1 - d >$ for various dollar amounts of x_1 and d. Their utility function for profit is *monotonic* since they prefer more money to less, i.e. if $x_1' > x_1''$, then x_1' is always preferred to x_1''. The methods for constructing a single-attribute utility function are well-defined (see for example, Keeney and Raiffa, 1976; Farquhar, 1984). We apply the certainty equivalent method for the construction of a single-attribute utility function.

Table 3.1: Data points assessed from the DM for a single attribute utility function of X_1.

Data point	Assessed value, $
$X_{0.875}$	115,000
$X_{0.75}$	80,000
$X_{0.625}$	45,000
$X_{0.5}$	22,000
$X_{0.375}$	0
$X_{0.25}$	-25,000
$X_{0.125}$	-40,000
X_{check}	22,000

The seven data points assessed from the DM are shown in Table 3.1. The last data point X_{check} serves for a consistency check. We curve fit a function that satisfies the required conditions: *monotonicity and risk aversion*. Figure 3.4 illustrates how well $U_1(x_1) = \ln(x_1 + 199,000) - 11.8$ fits (using a least-square error approach) the data points assessed from the decision maker. Clearly, the DM has a decreasingly risk-averse utility function over money. Here, $U_1^{-1}(x_1) = e^{U_1(x_1)+11.8} - 199,000$ and since $(\ln f(x_1))' = \frac{1}{f(x_1)}(f(x_1))'$

$$U_1'(x_1) = [\ln(x_1 + 199,000) - 11.8]' = [\ln(x_1 + 199,000)]'$$
$$= \frac{1}{x_1 + 199,000}(x_1 + 199,000)' = \frac{1}{x_1 + 199,000};$$
$$U_1''(x_1) = \left[\frac{1}{x_1 + 199,000}\right]' = [(x_1 + 199,000)^{-1}]' = -(x_1 + 199,000)^{-2};$$
$$\gamma = -\frac{U_1''(x_1)}{U_1'(x_1)} = -\frac{-(x_1 + 199,000)^{-2}}{(x_1 + 199,000)^{-1}} = \frac{1}{x_1 + 199,000},$$

where γ is the risk-aversion coefficient (see Arrow, 1965; Pratt, 1964).

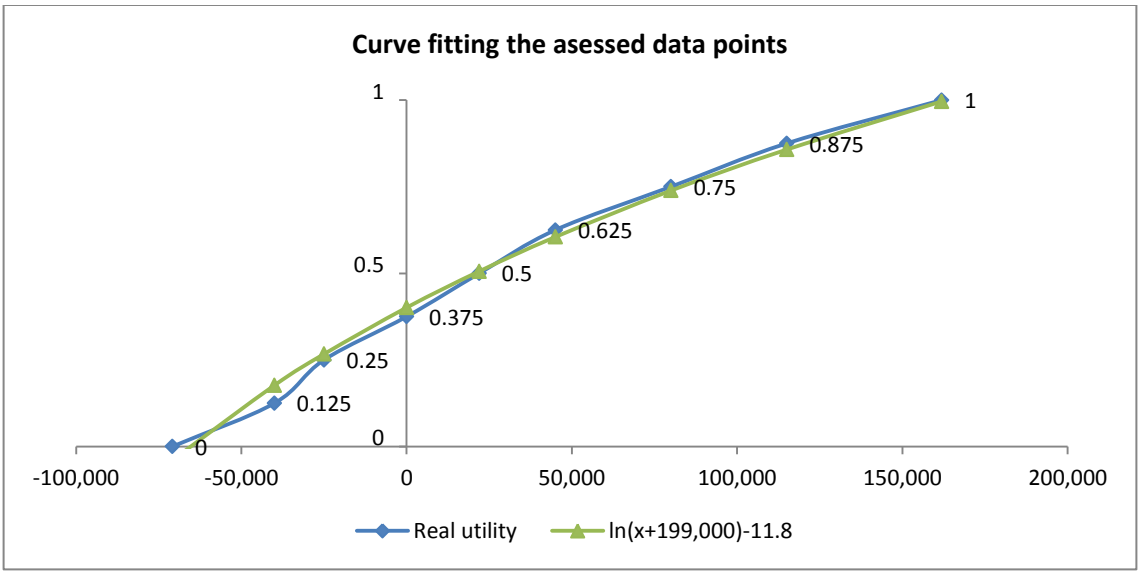


Figure 3.4: Curve fitting the assessed (in blue) data points for attribute X_1.

3.2.2 Evaluating the alternatives of the borehole recovery problem based on Profit

Figure 3.5 represents the partial decision tree constructed for the evaluation of the alternatives on the example of the chemical method. As an example, for the outcome value \$144,752, we take its U-value as $U(144,752) = \ln(144,752 + 199,000) - 11.8 = 0.9477$ and then the E-value of the U-value, EU, as a product of U-value and the joint probability. For chemical method, EU = 0.5557 and so the certainty equivalent (CE) is $\text{CE} = e^{\sum \text{EU} + 11.8} - 199,000 = \$33,276.32$. Similarly, we found the expected utilities of every other alternative and rank-ordered the alternatives as shown in Table 3.2.

To summarize, we could easily rank-order the alternatives of our decision problem by profit analysis as shown above. We found that the pneumoimpulsive method is the best alternative, followed by the chemical method, followed by physical and airflow methods. We are interested now whether this ranking changes if three additional attributes are considered. However, we first

need to check whether or not utility independence conditions are valid for our problem, which is the topic of the next Chapter.

Table 3.2: Rank-order of the alternatives based on Profit

Rank-order	Alternative	Expected Utility
#1	Pneumoimpulsive method	0.5562
#2	Chemical method	0.5557
#3	Physical method	0.5514
#4	Airflow method	0.5440
#5	Redrill	0.4592

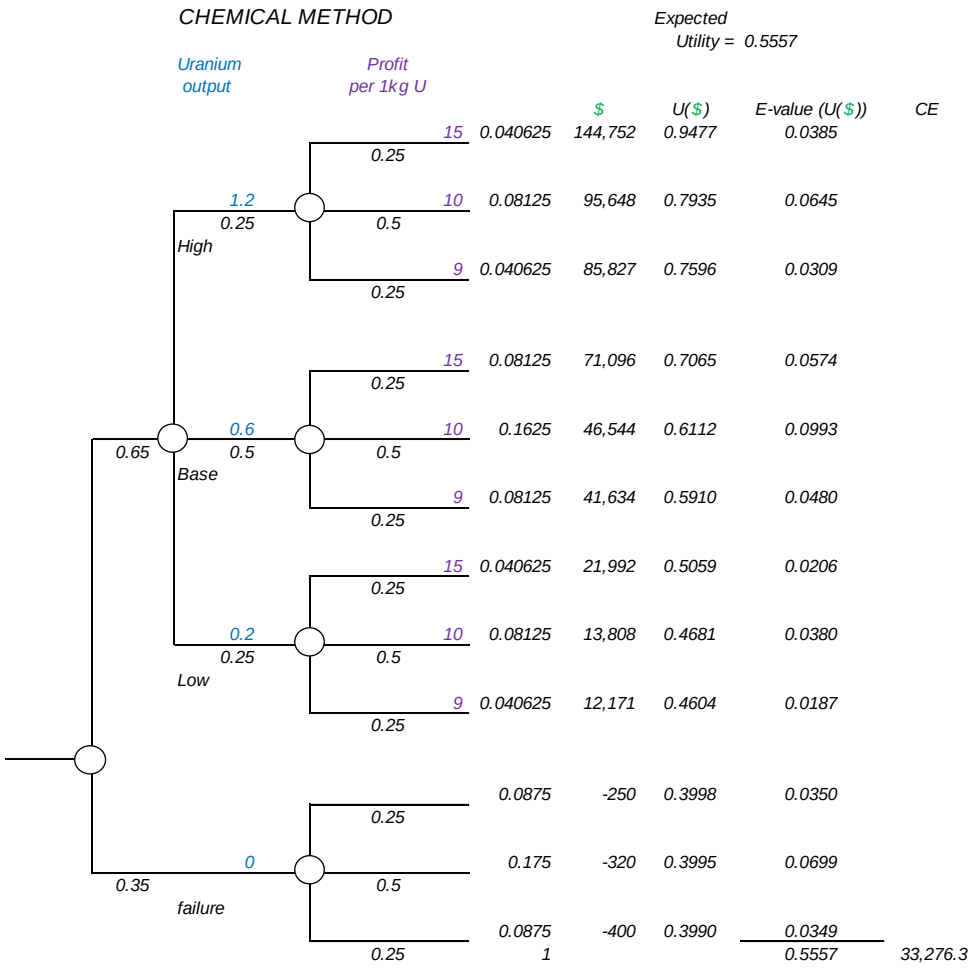


Figure 3.5: Partial decision tree for evaluation of the chemical method.

CHAPTER 4: TESTING INTERDEPENDENCE AMONG THE ATTRIBUTES OF THE BOREHOLE RECOVERY PROBLEM

4.1 OVERVIEW

The goal of this Chapter is to answer the following: (i) whether or not utility independence exists in the borehole recovery problem? (ii) are the canonical functional forms good for this decision problem? (iii) how people perform in answering questions about interdependence of the attributes? For this, we (i) implement new tests, (ii) conduct experiments with the professionals in the uranium extraction industry about utility interdependence among the attributes, and (iii) characterize the number of canonical and non-canonical forms and their convergence as the number of attributes increases (see Appendix F). Some of the initial results on these topics were presented in (Abdildin and Abbas, 2013); this Chapter provides more complete analysis.

4.2 NEW TESTS AND EXPERIMENTS ON PARTIAL UTILITY INDEPENDENCE

In this section we discuss new tests prepared for assessing partial utility independence conditions among the attributes defined in Section 1.3.3 (Table 1.3).

4.2.1 Test 1: Test of partial utility independence conditions in Experiment 1

As we discussed previously, canonical utility dependence matrices (UDMs) incorporate partial utility independence (UI) conditions and significantly simplify the construction of a multiattribute utility function (MUF). In order to verify their applicability in practice, we developed two tests and conducted two experiments with the experts in the uranium extraction industry. The task was the assessment of the utility dependence matrix (UDM) for the deep

borehole filter restoration problem. The experiments were administered via paper and pen. The initial experiment with Test 1 was mainly a one-to-one interview with the experts. The second experiment with more formal Test 2 was conducted remotely two months later.

Experiment 1 with Test 1. During the summer of 2012, I visited several companies in the uranium extraction industry; there I had a series of meetings with the decision makers. The goal was twofold: (i) gather information and materials about the deep borehole filter restoration problem, and (ii) identify the pool of experts, who are familiar with the problem and are willing to participate in the experiment. A trip to a well-field was organized by one of the companies to illustrate the problem. Participants were 105 experts in the uranium extraction industry who desired to take part in the experiment. All experts received the same materials as described below.

After a brief introduction, participants were provided with a two-page questionnaire (Test 1). The questionnaire contained a description of the experiment, parameters of the attributes as shown in Table 1.3, the initial UDM in the form of an 4×4 identity matrix, and a set of 12 questions. The questions were about the preferences of the participants on interdependencies of the attributes and were asked in simple language such as this first question:

"*Will your preferences (decisions) for uncertain situations on the level of the uranium output, X, and the number of hours spent on the process, Y, depend on a particular level (value) of Y given that the other two attributes, namely, Z and W, are held fixed?*"

The other eleven questions were similar (see the full test in Appendix A). Each question was followed by a choice of "*Yes/No*" answers. The assessors were given a tip: "*If your answer is "No", then in the conditional utility function $U(X|Y,Z,W)$, the value of Y is not strategically significant and can be replaced with its least preferred value, y^0, i.e.,*

$U(x \mid y, z, w) = U(x \mid y^0, z, w)$. " The assessors were also instructed to put a circle into the corresponding cells of the matrix for their positive answers and move to the next question if the answers were "No." The final question for the experts was to rate the complexity of the test on a scale between 1 (not difficult at all) and 5 (very difficult).

4.2.2 Test 2: Test of partial utility independence conditions in Experiment 2

Experiment 2 with Test 2. In the 2nd experiment, 40 experts were randomly selected from a large pool of experts. A new, more formal test was developed for the deep borehole filter restoration problem, but this time only three attributes were considered. This reduced the test size from 14 pages to eight. The attribute "Restoration time" was not included, but the description of the test explicitly provided the range of the process duration as from six to 200 hours. The attributes *Additional expenses* and *Health effects of the process* were denoted by Y and Z, respectively. The scaling measure of Z was shifted to the left (0 - 4), and the worst value of attribute X was set to zero. These adjustments were done merely for cognitive simplification of the test. The new table of attributes is presented in Table 4.1.

Table 4.1: Three attributes for the borehole restoration problem, their measures and ranges.

Attribute	**Measure**	**Range**	
		best	worst
X - Uranium output from a borehole	U kg/hour	$x^* = 3.2$	$x^0 = 0$
Y - Additional expenses (for repair or redrilling)	$	$y^* = 73$	$y^0 = 23.3\text{K}$
Z - Health effects of the process (job hazard)	0 – 4 scale	$z^* = 0$	$z^0 = 4$

The test contained eight pages; the 1st page contained a brief description of the experiment and the deep borehole filter restoration problem, the table with attributes, their measures, and ranges. Pages 2-7 contained six questions. The 1st question has three sub-questions. Part a) and part b)

ask the assessor's preference in Scenario 1 and Scenario 2, respectively. Each scenario has two risky projects, A and B. The assessor is requested to select from one of the three choices: project A, project B, and I (indifference between A and B). For example, if the assessor chooses project A, then he gambles on "all or nothing," which is represented by a tree with a circular *chance node*. Part c) asks the assessor whether his choice would change between Scenario 1 and Scenario 2 if the value of Y was fixed at some other value between its boundary values. The design of the other five questions was similar, only attributes and their values were varied. Page 8 had a question about the test complexity. The full test is provided in Appendix B.

As shown in Table 4.2, if the expert's responses in parts a) and b) of the question are different and the third c) part of the question has a positive response, then there is a utility dependence of one attribute from another, while the negative response leads to 1S independence. When the expert's preference does not switch between two scenarios, we may have either BI or UI.

Table 4.2: Possible responses and interdependencies; P is the Preference made out of {A, I, B}, ¬P denotes change of Preference.

Scenario 1 Part a)	Scenario 2 Part b)	Part c) of the question	Interdependence structure
P	¬P	Yes	Utility Dependence
P	¬P	No	One-Switch Independence
P	P	Yes	Boundary Independence
P	P	No	Utility Independence

4.3 ANALYSIS OF THE INTERDEPENDENCE OF THE ATTRIBUTES

In this section, our goal is to check whether or not utility independence (UI) among the attributes of the deep borehole filter restoration problem is a valid assumption. We are interested in the proportion of the assessors who assert UI.

4.3.1 Formulating a statistical test on utility and partial utility independence conditions among the attributes of the borehole recovery problem

Let the number of observations in the random sample that belong to a class of experts who assert

- Partial UI conditions in Experiment 1 (sample size is 105) with Test 1 be denoted by Υ_1;
- UI conditions in Experiment 2 (sample size is 40) be denoted by Υ_2.

Then $\hat{\Theta} = \frac{\Upsilon_c}{m}$, $c \in \{1, 2\}$ is a point estimator of the proportion of the population θ that belongs to class c. Note that m and θ are parameters of the binomial distribution. The sampling distribution of $\hat{\Theta}$ is approximately normal with mean θ and variance $\theta(1-\theta)/m$, if θ is not too close to either 0 or 1 and if m is relatively large. The approximation is better if $m\theta$ and $m(1-\theta)$ are greater than or equal to five. If m is large, then the distribution of $Z = \frac{\Upsilon - m\theta}{\sqrt{m\theta(1-\theta)}} = \frac{\hat{\Theta} - \theta}{\sqrt{\theta(1-\theta)/m}}$ is approximately standard normal. To construct the confidence interval (CI) on θ, note that $P\left(-z_{\alpha/2} \le Z \le z_{\alpha/2}\right) \simeq 1-\alpha$.

So $$P\left(-z_{\alpha/2} \le \frac{\hat{\Theta}-\theta}{\sqrt{\theta(1-\theta)/m}} \le z_{\alpha/2}\right) \simeq 1-\alpha$$ or

$P\left(\hat{\Theta} - z_{\alpha/2}\sqrt{\theta(1-\theta)/m} \le \theta \le \hat{\Theta} + z_{\alpha/2}\sqrt{\theta(1-\theta)/m}\right) \simeq 1-\alpha,$ where $\sqrt{\theta(1-\theta)/m}$ is known as the standard error of the point estimator $\hat{\Theta}$. Unknown θ can be replaced by $\hat{\Theta}$, so

$$P\left(\hat{\Theta} - z_{\alpha/2}\sqrt{\frac{\hat{\Theta}(1-\hat{\Theta})}{m}} \le \theta \le \hat{\Theta} + z_{\alpha/2}\sqrt{\frac{\hat{\Theta}(1-\hat{\Theta})}{m}}\right) \simeq 1-\alpha.$$

Then, the approximate CI on a binomial proportion is

$$\hat{\theta} - z_{\alpha/2}\sqrt{\frac{\hat{\theta}(1-\hat{\theta})}{m}} \le \theta \le \hat{\theta} + z_{\alpha/2}\sqrt{\frac{\hat{\theta}(1-\hat{\theta})}{m}}, \quad (14)$$

where $a = 0.05,\ z_{\alpha/2} = z_{0.025} = 1.96.$

4.3.2 The results of Test 1 in Experiment 1 based on a 95% confidence interval

In the experimental study of partial utility independence of $n = 4$ attributes of the deep borehole filter restoration problem, the participants were $m = 105$ experts from uranium mining. In total, 105 UDMs were collected. $\Upsilon_1 = 100,$ so $\hat{\Theta} = \frac{\Upsilon_1}{m} = \frac{100}{105} = 0.95$ and $\sqrt{\theta(1-\theta)/m} = 0.02.$

The 95% approximate CI on a binomial proportion is found by (14) as $0.91 \le \theta \le 0.99.$

Conclusion: In Experiment 1, 95% of the assessed experts asserted PUI among the attributes of the deep borehole filter restoration problem, and the other 5% did not assert any UI conditions. With 95% confidence we can conclude that the proportion of the experts who assert PUI is at least 0.91. These results imply that UI is not a valid assumption for this decision problem.

4.3.3 The results of Test 2 in Experiment 2 based on a 95% confidence interval

In the experimental study of utility independence of $n = 3$ attributes of the deep borehole filter restoration problem, the participants were $m = 40$ experts from uranium mining. In total, 40 UDMs were collected. $\Upsilon = 5$ UDMs were identity matrices, so $\hat{\Theta} = \frac{\Upsilon}{m} = \frac{5}{40} = 0.125$ and $\sqrt{\theta(1-\theta)/m} = 0.05$.

The 95% approximate CI on a binomial proportion is found by (14) as $0.023 \le \theta \le 0.228$.

Conclusion: In Experiment 2, only 12.5% of the assessed experts asserted UI among the attributes of the deep borehole filter restoration problem. With 95% confidence we can conclude that the proportion of the experts who assert UI is at most 0.228. These results imply that UI is not a valid assumption for this decision problem.

4.4 DISCUSSION OF THE RESULTS OF THE EXPERIMENTS

This sub-section presents further analysis of the data collected during the experiments.

4.4.1 Distributions of the assessors

In this subsection, a brief description of the assessors, the participants in the experiments, is given in terms of:

a) their allocation (city or well-field) and distribution by type of organization (Table 4.3),

b) their job positions (Table 4.4).

For example, Table 4.3 illustrates a diversity of the assessors by organizations' type and Table 4.4 presents the diversity of the experts' specializations.

Table 4.3: Experts' allocations

City					*Well-Field*		
Company A	*Company B*	*Company C*	*Academia, National Labs*	*Independent experts*	*Company 1*	*Company 2*	*Total no. of experts*
6	27	1	4	10	37	32	117

Table 4.4: Experts by job position

Job position	*Number of experts*
Agent of Supply Department	1
Boring master	21
CTO Boring Division	5
CTO Damage Control	1
CTO New Technology	1
CTO of the Expedition	2
Damage control master	1
Deputy CTO Production	1
Deputy Director	1
Deputy Head of Technology	1
Electrician	1
Engineer	8
Engineer-designer	1
Engineer-technologist	9
Head of Finance	1
Head of Geological Survey	1
Nuclear Engineer	2
Head of Production	1
Head of Supply Department	1
Leading engineer	6
Leading technologist	3
Mining surveyor technician	1
Network planning technician	1
Position was not mentioned	12
Operator in a borehole-service unit	23
Assistant Professor	2
Senior manager of Geological Survey	1
Senior manager of Production	1
Senior master in a borehole-service unit	1
Hydrogeologist	1
Technologist	5
Total number of experts	117

4.4.2 Discussion of Experiment 1 with Test 1: novelty, importance, and results

In the 1st experiment with Test 1, the experts were asked to 1) assess the UDM of the four attributes of the deep borehole filter restoration problem, 2) rate the complexity of the test on a 1-

5 scale, 3) put name, position, and signature. In this experiment 105 4×4 binary UDMs were assessed (see Appendix C).

The results of Experiment 1 are the following:

- 100 UDMs (95%) presented partial UI conditions, and five UDMs did not present any UI assertions.
- 59 out of 105 UDMs (56%) were canonical.
- The experts found that the assessments were relatively easy; the average score of 103 experts on the 1-5 scale was 2.55. Two experts did not rate the test.
- Typical positions were as shown in Table 4.4.

The novelty and importance of Experiment 1 are the following.

1) It is the first practical study of partial UI conditions assessed from a large number of experts in the uranium extraction industry.
2) It demonstrates that the attributes of the problem do have partial UI conditions.
3) The study also verifies the applicability of canonical forms for large industrial projects.
4) The experts were comfortable with the experiment and the test was not difficult for them.

4.4.3 Discussion of Experiment 2 with Test 2: novelty, importance, and results

The goal of the 2nd experiment was to verify *indirectly* the existence of partial UI utilizing a formal test. This time, the experts were not asked to fill the UDMs directly as in the 1st experiment.

The novelty of this test is as follows:

1) it is the first test developed for assessing partial UI conditions including boundary independence (BI) and one-switch (1S) independence,

2) it is formal, but elicits the partial UI (PUI) conditions indirectly,
3) it does not contain any special decision analysis terms and is not difficult for assessors, and
4) it is easily scalable for any number of attributes.

The formal test was sent to 40 experts in the uranium extraction industry by e-mail either personally or through a facilitator, a representative of the company, who distributed the test and collected the results.

The results of the formal test are the following:

1. Only five experts found that the attributes are UI of their complements.
2. Out of 40 responses, 32 (80%) lead to canonical utility dependence matrices.
3. The average complexity of the test was 2.97/5, which proves that the test was not difficult in general. One expert did not rate the complexity of the test.
4. The experts in the field (19) found the test more difficult (average is 3.53) compared with the experts in the main office (20), where the test was rated 2.45/5.

In Test 2 (see Appendix B), when the answer for the third part (c) of the question is negative we will have one-switch independence, and if the answers for (a) and (b) are identical then UI (or, zero-switch) condition exists. Zero-switch independence is a proper subset of 1S independence (Abbas and Bell, 2011), see Figure 4.1.

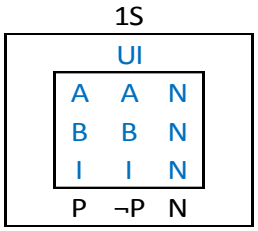


Figure 4.1: Possible relations of preference switch.

For example, one of the experts provided the following answers: 1) ABN 2) ABY 3) BBN 4) BBN 5) BBN 6) BBN, and 2 for a test complexity. In question 1 (see Appendix B) the expert switched from project A to B and he asserted that his preference would not switch anymore. By Table 4.2 we have, $(X\ 1S\ Y \mid Z)$.

In other words, the expert switches to project B only if money spent for the restoration process allows drilling a new borehole, and he might want to have at least $x = 0.5$ kg/hour in uranium output for this money. Since 1S independence does not imply UI, we can put a node into the XY cell of the UDM. In question 2) $(X\ UD\ Z \mid Y)$ and we put a circle into the XY cell. If we accept (without additional tests) that in the other four questions the expert provided UI conditions, the UDM will have the canonical form as shown in Figure 2.3 (a) and also illustrated here in Figure 4.2 for your convenience.

	X	Y	Z
X	○	○	○
Y		○	
Z			○

Figure 4.2: Example of the utility dependence matrix for three attributes from Experiment 2.

The summary of actual responses to the formal test is presented in Figure 4.3. Expert no. 8 did not provide his answer to part c) of question 4, but even without this response the UDM is non-canonical. However, depending on this answer, the cell will represent either UD (if "Yes"), or 1S (if "No"). Expert no.18 did not provide answers to parts c) of questions from three to six, because his "answers depend on the shifted values of the attribute;" hence, we can infer his "Yes" to these questions.

Answers to the questions:

Experts	1			2			3			4			5			6			Complexity
	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	of the test
1	I	A	N	B	B	Y	I	A	Y	A	B	Y	I	A	N	A	B	N	5
2	A	I	Y	A	A	Y	I	A	N	I	I	Y	B	B	Y	A	I	N	3
3	A	I	Y	A	A	Y	I	B	N	A	A	Y	I	A	Y	A	I	N	4
4	B	I	Y	I	A	Y	A	B	N	A	A	N	I	A	N	I	A	N	4
5	A	A	N	B	A	Y	B	I	N	A	A	N	I	A	Y	A	I	N	5
6	I	A	N	A	I	N	A	I	N	B	I	N	A	I	N	A	I	N	3
7	A	A	Y	I	A	N	I	I	Y	A	I	Y	B	I	Y	I	A	Y	3
8	A	B	Y	B	B	N	A	I	Y	B	I		I	A	Y	I	I	N	
9	B	A	Y	A	A	N	B	I	Y	A	A	Y	B	A	Y	A	B	N	5
10	A	I	Y	A	A	N	A	B	Y	A	B	N	I	I	N	A	B	Y	5
11	A	I	N	A	I	Y	B	A	N	B	I	Y	A	A	N	A	I	N	4
12	I	A	N	B	I	N	A	I	Y	B	I	N	I	A	N	A	I	Y	3
13	A	I	N	A	I	N	B	I	Y	A	I	Y	B	I	N	I	B	Y	3
14	A	I	N	B	I	Y	A	I	N	B	B	Y	A	I	N	A	I	Y	4
15	A	I	N	B	A	N	A	I	N	B	A	Y	B	A	Y	B	A	N	4
16	B	B	N	B	I	N	I	I	Y	I	I	Y	B	B	Y	B	B	Y	2
17	B	A	Y	B	B	Y	A	B	N	A	A	Y	B	B	Y	A	B	N	3
18	A	A	Y	A	A	Y	B	I		B	B		A	I		A	A		3
19	A	A	Y	A	A	Y	A	B	Y	B	B	Y	A	A	Y	A	A	Y	4
20	B	B	N	B	B	N	A	A	N	B	B	N	A	A	N	A	A	Y	3
21	A	A	N	A	A	N	B	B	N	B	B	N	B	B	N	B	B	N	2
22	B	I	Y	A	B	N	A	A	Y	I	B	Y	B	I	Y	B	B	N	2
23	I	I	Y	A	A	N	B	B	N	I	B	Y	B	B	N	A	B	Y	2
24	B	B	Y	B	B	Y	B	A	Y	A	A	Y	B	B	Y	B	B	Y	3
25	B	B	Y	B	B	Y	B	A	Y	B	B	Y	A	A	N	B	A	Y	2
26	A	A	N	A	B	Y	B	B	N	B	B	N	B	B	N	B	B	N	3
27	A	A	N	A	A	N	B	B	N	B	B	N	B	B	N	B	B	N	2
28	A	B	N	A	B	Y	B	B	N	B	B	N	B	B	N	B	B	N	2
29	I	B	N	B	A	Y	A	A	N	B	A	Y	A	A	N	A	B	N	2
30	B	B	N	B	A	Y	B	A	Y	A	A	N	A	B	Y	B	A	N	2
31	B	B	Y	B	B	Y	B	A	Y	A	A	Y	B	B	Y	B	B	Y	2
32	B	B	Y	B	B	Y	B	A	Y	A	A	Y	B	B	Y	B	B	Y	3
33	I	B	N	B	A	Y	A	A	N	B	A	Y	A	A	N	A	B	N	2
34	B	B	Y	B	B	Y	B	A	Y	A	A	Y	B	B	Y	B	B	Y	2
35	A	B	Y	B	B	N	A	B	Y	B	B	N	B	B	N	B	B	N	2
36	A	A	N	A	A	N	A	A	N	I	I	N	I	I	N	I	I	N	3
37	A	B	Y	A	B	Y	A	B	Y	I	I	N	B	I	Y	B	B	N	2
38	B	I	Y	B	I	Y	B	I	Y	B	I	Y	B	I	Y	B	B	N	3
39	B	A	Y	B	A	Y	A	I	Y	A	I	Y	I	I	N	I	I	N	3
40	B	B	Y	B	B	Y	B	A	Y	A	A	Y	B	B	Y	B	B	Y	2
															Average:				2.97

Figure 4.3: Experts' responses to Test 2 (with 3 attributes).

Based on data in Figure 4.3, we can construct utility dependence matrices as shown in Figure 4.4, where “o”, “b,” and “s” represent utility dependence, boundary independence, and one-switch independence, respectively, and the vacant cell assumes UI.

Boundary independence condition has a convenience property that it enables testing of the conditional utility functions at the boundary values of the attributes instead of at their whole domains. This definitely simplifies the assessment complexity for the decision maker. In addition to that, it allows the use of certain decomposition rules. Besides that, there may exist a situation having an attribute BI, but not necessarily UI on another attribute since BI is a weaker condition than UI.

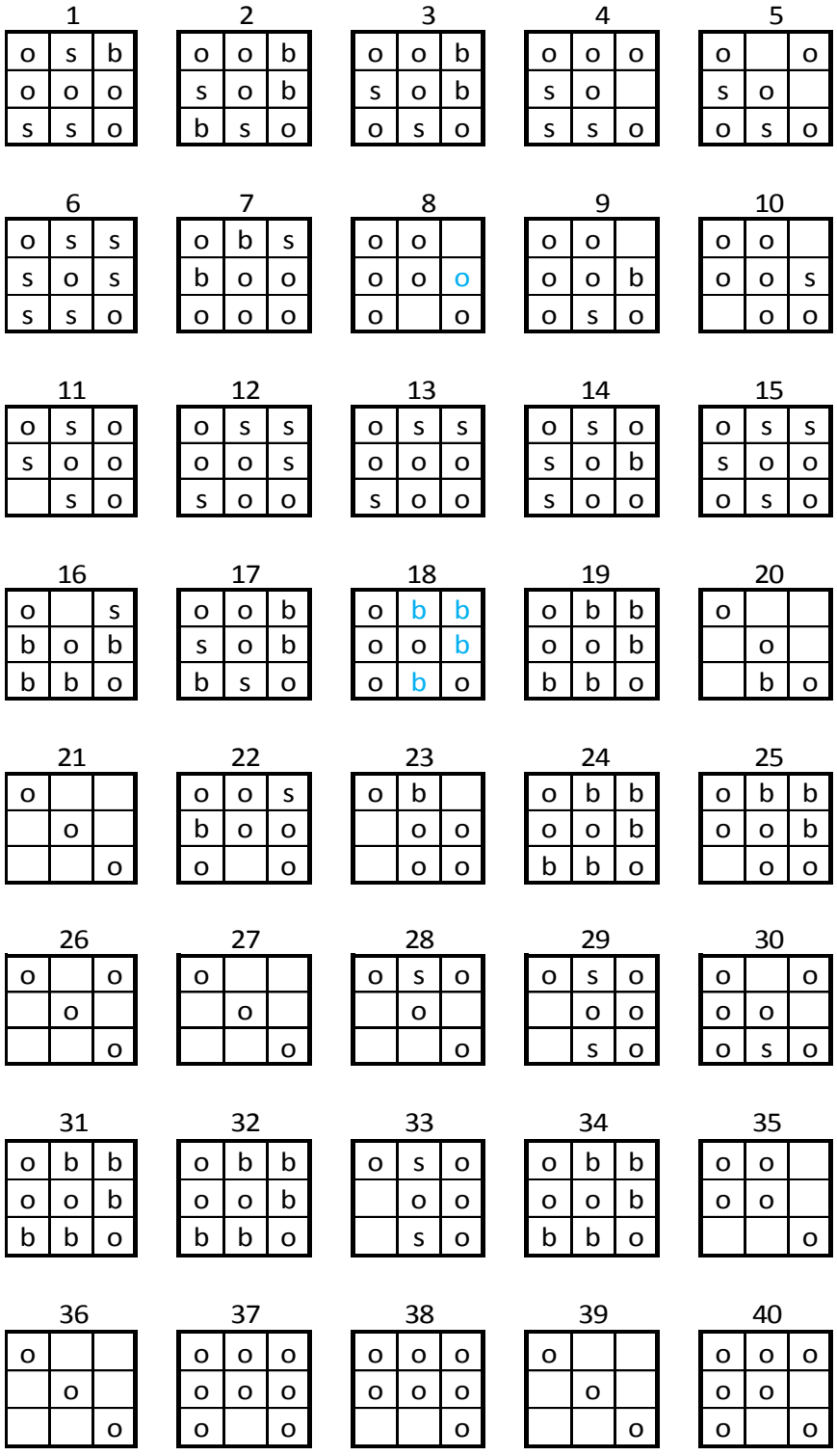


Figure 4.4: UDMs with BI and 1S conditions based on experts' actual responses.

When BI and UI exist in the assessed UDM, we can utilize the reduction rule on "conditional UI of the second attribute on the first" (Abbas, 2011b).

Proposition 1 (Abbas, 2011b): *If (i) X is conditionally UI of Y given Z,* $U(x \mid y,z) = U(x \mid y^0,z)$*, and (ii) Y is conditionally BI of X given Z,* $U(y \mid x^*,z) = U(y \mid x^0,z)$*, then Y is conditionally UI of X given Z,* $U(y \mid x,z) = U(y \mid x^0,z)$.

This Proposition is applied on four UDMs in Figure 4.4: {16, 20 23, 25}. For example, for UDM #16, this Proposition can be applied directly as follows: since *X UI Y|Z* and *Y BI X|Z* (see Figure 4.5(a)), then *Y UI X|Z* (Figure 4.5(b)).

	X	*Y*	*Z*
X	○		s
Y	b	○	b
Z	b	b	○

(a)

	X	*Y*	*Z*
X	○		s
Y		○	b
Z	b	b	○

(b)

Figure 4.5: UDM #16 assessed from expert no. 16 during the formal test for the deep borehole filter restoration problem before (a), and after (b) Proposition 1 is applied.

For UDM #20, notice: since *Y UI Z|X* and *Z BI Y|X*, then *Z UI Y|X*.

	X	*Y*	*Z*
X	○		
Y		○	
Z		b	○

(a)

	X	*Y*	*Z*
X	○		
Y		○	
Z			○

(b)

Figure 4.6: UDM #20 assessed from expert no. 20 during the formal test for the deep borehole filter restoration problem before (a), and after (b) Proposition 1 is applied.

For UDM #23, notice: since *Y UI X|Z* and *X BI Y|Z*, then *X UI Y|Z*.

	X	*Y*	*Z*
X	○	b	
Y		○	○
Z		○	○

(a)

	X	*Y*	*Z*
X	○		
Y		○	○
Z		○	○

(b)

Figure 4.7: UDM #23 assessed from expert no. 23 during the formal test for the deep borehole filter restoration problem before (a), and after (b) Proposition 1 is applied.

For UDM #25, notice that since *Z UI X|Y* and *X BI Z|Y*, then *X UI Z|Y*.

	X	*Y*	*Z*
X	○	b	b
Y	○	○	b
Z		○	○

(a)

	X	*Y*	*Z*
X	○	b	
Y	○	○	b
Z		○	○

(b)

Figure 4.8: UDM #25 assessed from expert no. 25 during the formal test for the deep borehole filter restoration problem (a) before, and (b) after Proposition 1 is applied.

Final UDMs with BI and 1S conditions after applying the Proposition 1 are illustrated in Figure 4.9.

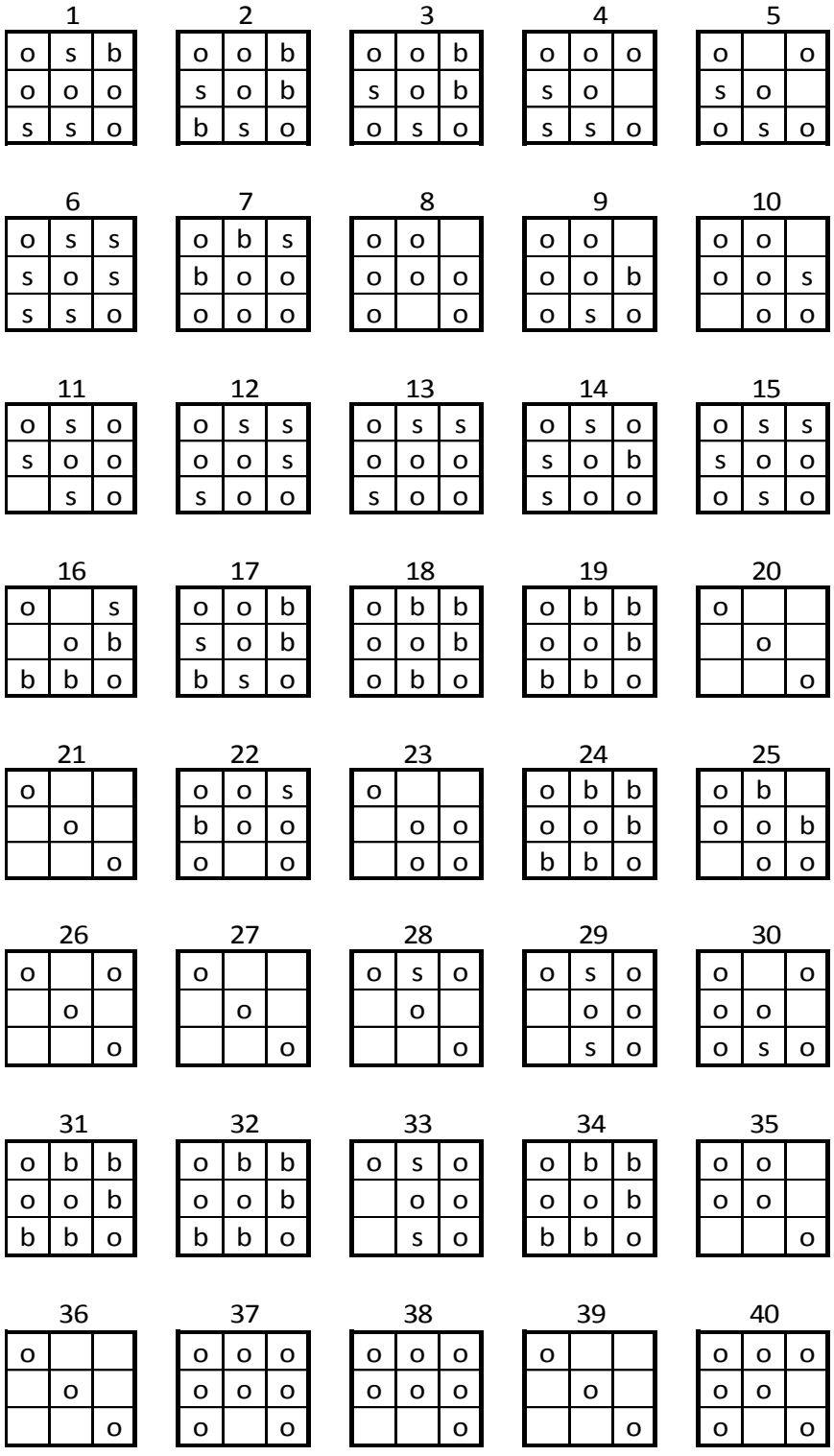


Figure 4.9: Final UDMs with BI and 1S conditions after applying Proposition 1.

We can also convert these UDMs to binary UDMs for comparison purposes only. Since neither boundary independence nor one-switch independence imply UI, we can replace them by a utility dependence node. We now easily determine that all matrices except the set {5, 8, 23, 26, 29, 30, 33} are canonical. The resulting binary UDMs are shown in Figure 4.10.

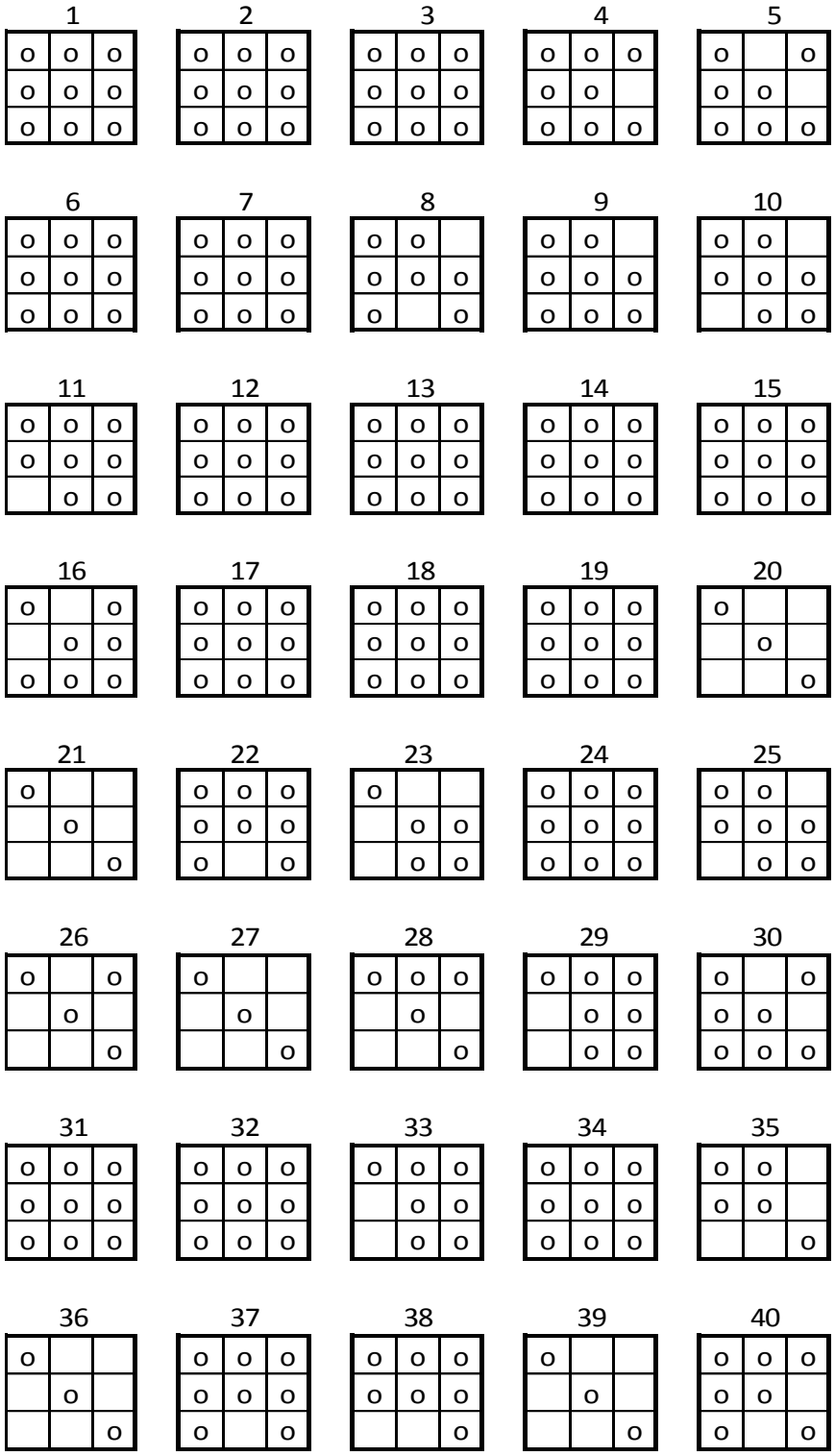


Figure 4.10: Actual UDMs in Figure 4.9 converted to binary UDMs.

4.5 SUMMARY

This Chapter verified that the assumption of utility independence may not be appropriate for the deep borehole filter restoration problem. Two tests were modelled for the assessment of partial utility independence conditions, which were utilized in two experiments. The experimental analysis of UI conditions among the attributes of the deep borehole filter restoration problem included the assessment of the preferences of the 117 experts in uranium mining.

In the first experimental study, 95% of the 105 experts asserted partial utility independence conditions among four attributes of our decision problem, while the other 5% did not assert any UI conditions. Based on the results of the first experiment, with 95% confidence we can conclude that the proportion of the population of the thousands of experts working in uranium mining who assert partial utility independence of the attributes is at least 0.91.

In the second experiment with Test 2, 12.5% of assessors asserted utility independence conditions and 80% asserted canonical forms. Based on the results of the second experiment, with 95% confidence we can conclude that the proportion of the population of the thousands of experts who assert utility independence of the attributes is at most 0.228.

The results imply that utility independence may not be a valid assumption for this decision problem. The experiments also verify the applicability of canonical forms to this complex decision problem.

We also determine of the number and convergence of (i) canonical forms, and (ii) newly introduced proper partial forms in Appendix F.

In the following Chapter the alternatives of the deep borehole filter restoration problem are analyzed based on the four most important attributes from the ultimate decision maker’s point of

view. First, the multiattribute analysis is conducted based on UI assumption (multilinear form), then based on the preferences of the DM (partial UI). The alternatives of the decision problem are rank-ordered and the results are compared.

CHAPTER 5: EVALUATING ALTERNATIVES OF THE DEEP BOREHOLE FILTER RESTORATION PROBLEM

In this Chapter, we provide multiattribute analysis of the deep borehole filter restoration problem using four attributes selected by the ultimate decision maker, the Deputy General Director of a transnational corporation: *Profit* (X_1), *Labor protection* (X_2), *Safety of operations* (X_3), and *Preserved area* (X_4). These attributes were discussed in Chapter 1, see Table 1.4.

5.1 MULTIATTRIBUTE ANALYSIS UNDER THE UTILITY INDEPENDENCE ASSUMPTION

First, we can assume utility independence among these four attributes and evaluate the alternatives. Then, we can assess the true preferences of the decision maker and check whether or not our approximation is good.

5.1.1 Multilinear form of the MUF for the deep borehole filter restoration problem

Let us construct the multilinear form of the MUF for our decision problem. Utilizing (11) we have the multilinear form of the MUF for $n = 4$ attributes:

$$\begin{aligned} U(x_1, x_2, x_3, x_4) = {} & k_1U_1(x_1) + k_2U_2(x_2) + k_3U_3(x_3) + k_4U_4(x_4) \\ & + k_{12}U_1(x_1)U_2(x_2) + k_{13}U_1(x_1)U_3(x_3) + k_{14}U_1(x_1)U_4(x_4) \\ & + k_{23}U_2(x_2)U_3(x_3) + k_{24}U_2(x_2)U_4(x_4) + k_{34}U_3(x_3)U(x_4) \\ & + k_{123}U_1(x_1)U_2(x_2)U_3(x_3) + k_{124}U_1(x_1)U_2(x_2)U_4(x_4) + k_{134}U_1(x_1)U_3(x_3)U_4(x_4) \\ & + k_{234}U_2(x_2)U_3(x_3)U_4(x_4) + k_{1234}U_1(x_1)U_2(x_2)U_3(x_3)U_4(x_4), \end{aligned} \tag{15}$$

where there are four one-attribute utility functions (size one) and 2^n - 2 = 14 independent scaling constants to assess.

Borcherding, Eppel, and von Winterfeldt (1991) discuss elicitation of weights by comparing four weighting methods: the ratio method, the swing weight method, the tradeoff method, and the pricing-out method.

In practice, we can assess $U_i(x_i) = U(x_i \mid \bar{x}_i^*) = \dfrac{U(x_i, \bar{x}_i^*) - U(x_i^0, \bar{x}_i^*)}{1 - U(x_i^0, \bar{x}_i^*)}$, $U(x_i^0, \bar{x}_i^*) \neq 1$, by the standard *certainty equivalent* method and the scaling constants by simple questions as shown below for k_1. In this example, the answer $p = 85\% \Rightarrow U(x_1^*, x_2^0, x_3^0, x_4^0) = 0.85$.

What probability p makes you indifferent between LHS and RHS?

$(x_1^*, x_2^0, x_3^0, x_4^0)$ $\sim$ *p%* $(x_1^*, x_2^*, x_3^*, x_4^*)$ / *(1-p)%* $(x_1^0, x_2^0, x_3^0, x_4^0)$

Your answer: $p =$ _____ %

5.1.2 Assessments of the multilinear form of the MUF

We now discuss the assessments of the four one-attribute utility functions and the scaling constants in (15). There are different methods for assessing a single-attribute utility function (Farquhar, 1984; Abbas, 2002a). We apply the certainty equivalent method; as an example, see the complete test for attribute X_3 in Appendix G. The data points assessed from the DM are shown in Table 5.1. The last data point, X_{check}, serves for a consistency check; it should be equal to $X_{0.5}$.

First, we assessed the single-attribute utility functions of (15). The utility function for X_1 is the same as in Chapter 3, $U_1(x_1) = \ln(x_1 + 199,000) - 11.8$, so it was not assessed twice.

Table 5.1: Data points assessed from the DM for single attribute utility functions of X_2, X_3, X_4.

Data point	**Assessed values**		
	X_2, employees	X_3, days	X_4, m^2
$X_{0.875}$	-	-	6,000
$X_{0.75}$	63	1,270	5,700
$X_{0.625}$	-	-	5,300
$X_{0.5}$	38	900	5,000
$X_{0.375}$	-	-	4,650
$X_{0.25}$	15	575	4,400
$X_{0.125}$	-	-	4,150
X_{check}	38	900	5,000

The other three functions were curve fitted using the least-square error approach as:

$$U_2(x_2) = \frac{1-e^{-\gamma_2 x_2}}{1-e^{-\gamma_2}}, \quad 0 \le x_2 \le 1, \quad \gamma_2 = 1.1$$

$$U_3(x_3) = \frac{1-e^{-\gamma_3 x_3}}{1-e^{-\gamma_3}}, \quad 0 \le x_3 \le 1, \quad \gamma_3 = 1.2$$

$$U_4(x_4) = 2.0223 \ln(x_4) - 16.73$$

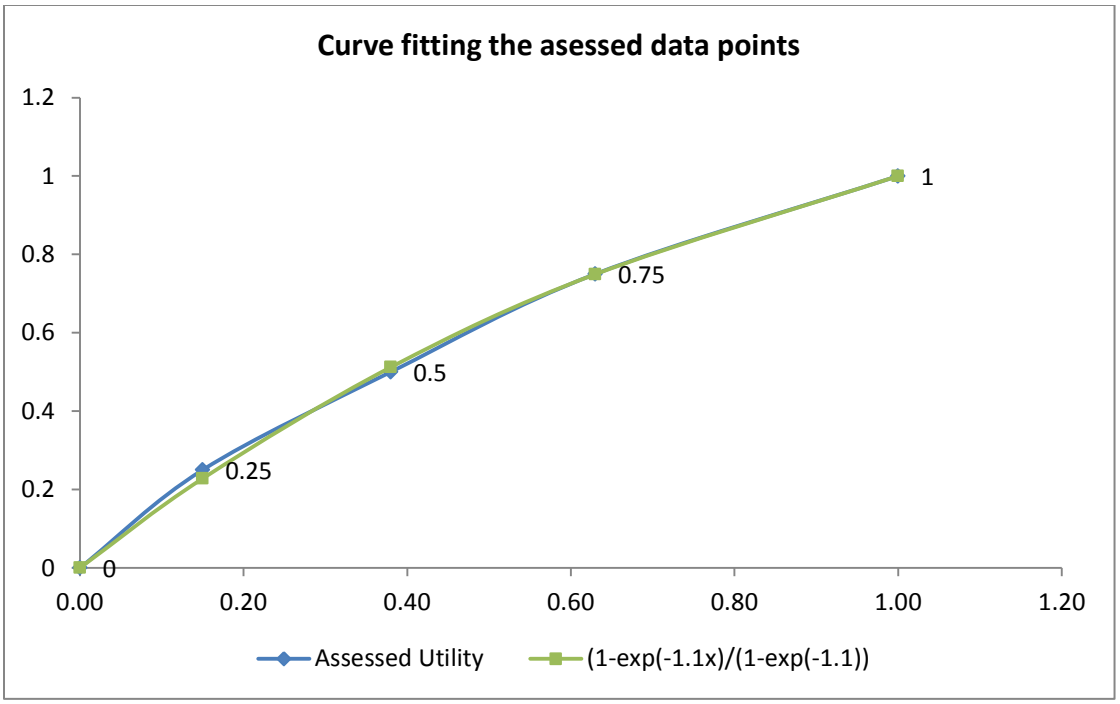


Figure 5.1: Curve fitting the assessed (in blue) data points for attribute X_2.

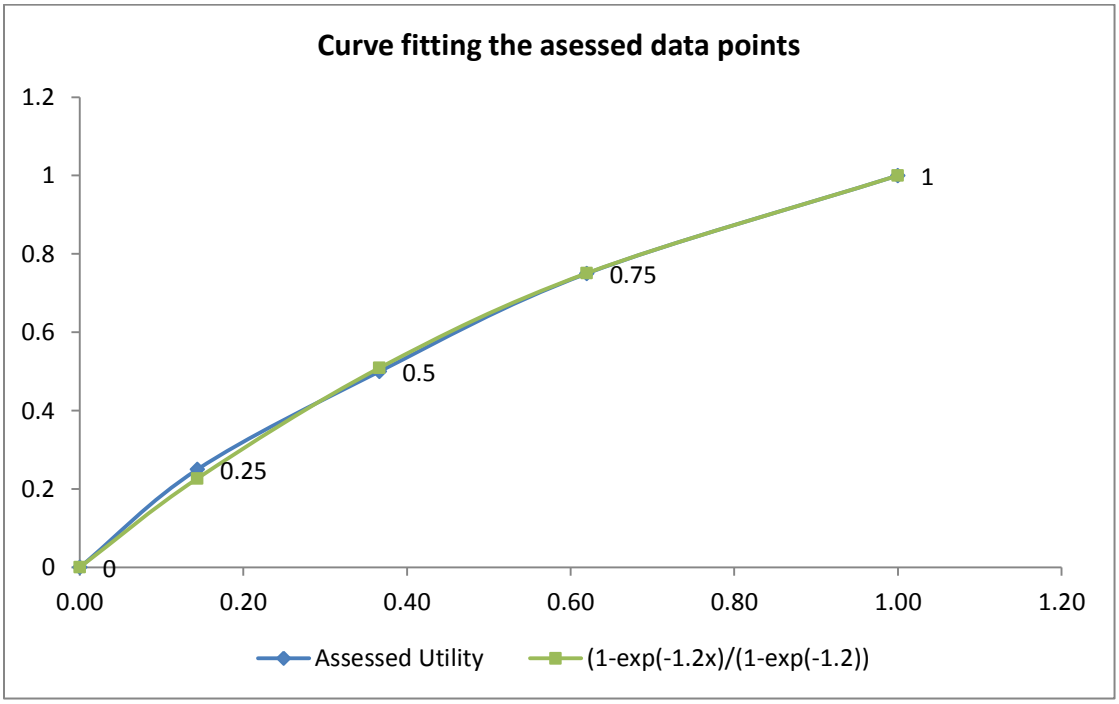


Figure 5.2: Curve fitting the assessed (in blue) data points for attribute X_3.

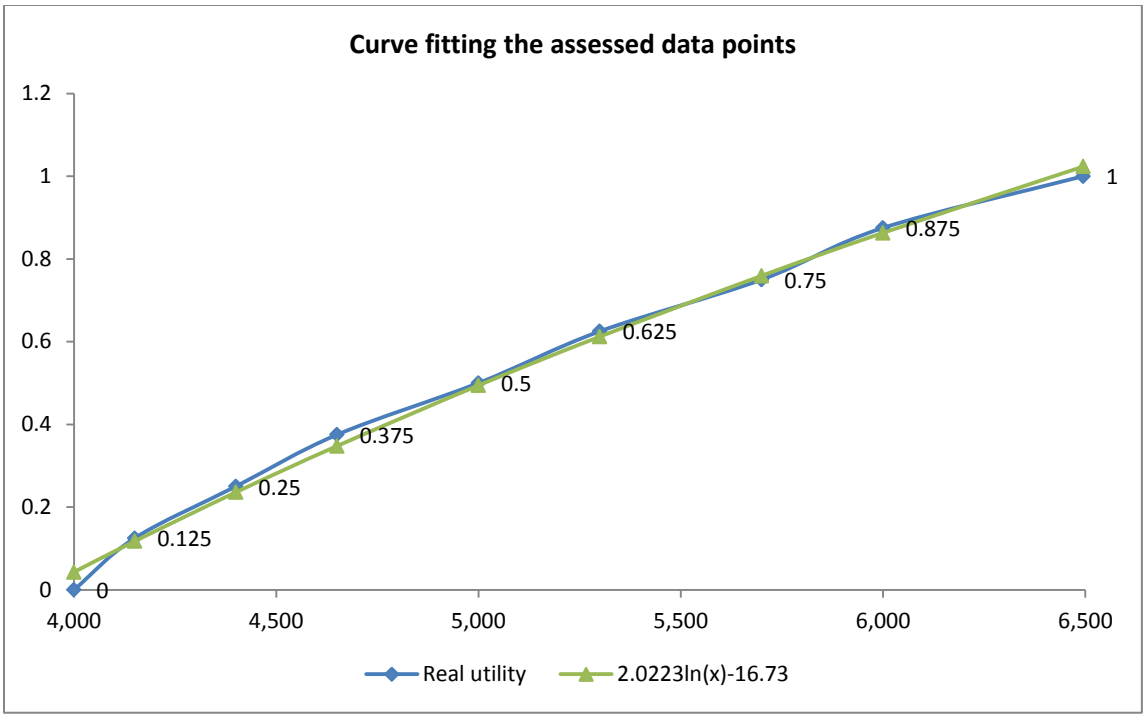


Figure 5.3: Curve fitting the assessed (in blue) data points for attribute X_4.

Figures 5.1-5.3 illustrate how well these functions fit with the data points we have assessed from the decision maker.

The assessment of the scaling constants was done as shown in Appendix H. Let us denote the consequences as:

$$\mathrm{A}:(x_1^0, x_2^*, x_3^*, x_4^*), \mathrm{B}:(x_1^*, x_2^0, x_3^*, x_4^*), \mathrm{C}:(x_1^*, x_2^*, x_3^0, x_4^*), \text{ and D: } (x_1^*, x_2^*, x_3^*, x_4^0).$$

We asked the DM's preferences on each pair and the answers were: $\mathrm{A} \prec \mathrm{B}$, $\mathrm{B} \prec \mathrm{C}$, $\mathrm{C} \succ \mathrm{D}$, $\mathrm{A} \prec \mathrm{C}$, $\mathrm{A} \prec \mathrm{D}$, $\mathrm{B} \succ \mathrm{D}$, from which we conclude that $\mathrm{A} \prec \mathrm{D} \prec \mathrm{B} \prec \mathrm{C}$, which implies that the DM ranks the attributes in the following order (from most to least important): X_1, X_4, X_2, X_3. In questions 2 and 3, the DM's preferences were the LHS and RHS of the deals, respectively, which show that the DM is risk averse.

The assessed scaling constants in (15) are the following:

$k_1 = U(x_1^*, x_2^0, x_3^0, x_4^0) = 0.85,$
$k_2 = k_3 = k_4 = 0,$
$k_{12} = U(x_1^*, x_2^*, x_3^0, x_4^0) - k_1 - k_2 = 0.88 - 0.85 - 0 = 0.03;$
$k_{13} = U(x_1^*, x_2^0, x_3^*, x_4^0) - k_1 - k_3 = 0.86 - 0.85 - 0 = 0.01;$
$k_{14} = U(x_1^*, x_2^0, x_3^0, x_4^*) - k_1 - k_4 = 0.90 - 0.85 - 0 = 0.05;$
$k_{23} = U(x_1^0, x_2^*, x_3^*, x_4^0) - k_2 - k_3 = 0 - 0 - 0 = 0;$
$k_{24} = U(x_1^0, x_2^*, x_3^0, x_4^*) - k_2 - k_4 = 0 - 0 - 0 = 0;$
$k_{34} = U(x_1^0, x_2^0, x_3^*, x_4^*) - k_3 - k_4 = 0 - 0 - 0 = 0;$

$k_{123} = U(x_1^*, x_2^*, x_3^*, x_4^0) - k_{12} - k_{13} - k_{23} - k_1 - k_2 - k_3 = 0.92 - 0.03 - 0.01 - 0 - 0.85 - 0 - 0 = 0.03;$
$k_{124} = U(x_1^*, x_2^*, x_3^0, x_4^*) - k_{12} - k_{14} - k_{24} - k_1 - k_2 - k_4 = 0.96 - 0.03 - 0.05 - 0 - 0.85 - 0 - 0 = 0.03;$
$k_{134} = U(x_1^*, x_2^0, x_3^*, x_4^*) - k_{13} - k_{14} - k_{34} - k_1 - k_3 - k_4 = 0.94 - 0.01 - 0.05 - 0 - 0.85 - 0 - 0 = 0.03;$
$k_{234} = U(x_1^0, x_2^*, x_3^*, x_4^*) - k_{23} - k_{24} - k_{34} - k_2 - k_3 - k_4 = 0,$
$k_{1234} = 1 - k_1 - k_2 - k_3 - k_4 - k_{12} - k_{13} - k_{14} - k_{23} - k_{24} - k_{34} - k_{123} - k_{124} - k_{134} - k_{234} = -0.03.$

The DM stated that attribute profit, X_1, is an *utility-dominant attribute* (Abbas and Howard, 2005). This means that if attribute profit, X_1, takes the worst value (i.e. loss), then the DM's utility function is equal to zero, i.e. there is the *attribute dominance condition* (Abbas and Howard, 2005).

Since attribute X_1 (Profit) is a *utility-dominant attribute*, i.e.

$U(x_1, x_2, x_3, x_4) = 0,$ if $x_1 = x_1^0,$ then $k_2 = k_3 = k_4 = k_{23} = k_{24} = k_{34} = k_{234} = 0,$ and (15) becomes

$$\begin{aligned} U(x_1, x_2, x_3, x_4) = {} & k_1 U_1(x_1) + k_{12} U_1(x_1) U_2(x_2) + k_{13} U_1(x_1) U_3(x_3) + k_{14} U_1(x_1) U_4(x_4) \\ & + k_{123} U_1(x_1) U_2(x_2) U_3(x_3) + k_{124} U_1(x_1) U_2(x_2) U_4(x_4) \\ & + k_{134} U_1(x_1) U_3(x_3) U_4(x_4) + k_{1234} U_1(x_1) U_2(x_2) U_3(x_3) U_4(x_4). \end{aligned} \tag{16}$$

Or,

$$\begin{aligned} U(x_1, x_2, x_3, x_4) = {} & U_1(x_1)[k_1 + k_{12} U_2(x_2) + k_{13} U_3(x_3) + k_{14} U_4(x_4) \\ & + k_{123} U_2(x_2) U_3(x_3) + k_{124} U_2(x_2) U_4(x_4) \\ & + k_{134} U_3(x_3) U_4(x_4) + k_{1234} U_2(x_2) U_3(x_3) U_4(x_4)]. \end{aligned} \tag{17}$$

Further,

$$\begin{aligned} U(x_1, x_2, x_3, x_4) = {} & U_1(x_1)[k_1 + U_2(x_2)[k_{12} + k_{123} U_3(x_3) + k_{124} U_4(x_4) + k_{1234} U_3(x_3) U_4(x_4)] \\ & + k_{13} U_3(x_3) + k_{14} U_4(x_4) + k_{134} U_3(x_3) U_4(x_4)]. \end{aligned}$$

(18)

Finally,

$$U(x_1, x_2, x_3, x_4) = U_1(x_1)[k_1 + U_2(x_2)[k_{12} + k_{123}U_3(x_3) + (k_{124} + k_{1234}U_3(x_3))U_4(x_4)] + k_{13}U_3(x_3) + (k_{14} + k_{134}U_3(x_3))U_4(x_4)]. \quad (19)$$

Then the multilinear form of the MUF will be:

$$U(x_1, x_2, x_3, x_4) = U_1(x_1)\left[\begin{array}{r} k_1 + U_2(x_2)\left[k_{12} + k_{123}U_3(x_3) + \left(k_{124} + k_{1234}U_3(x_3)\right)U_4(x_4)\right] \\ + k_{13}U_3(x_3) + \left(k_{14} + k_{134}U_3(x_3)\right)U_4(x_4) \end{array}\right],$$

where

$$\begin{array}{ll} U_1(x_1) = \ln(x_1 + a_1) - b_1; & a_1 = 199{,}000;\ b_1 = 11.8; \\ U_2(x_2) = \dfrac{1 - e^{-\gamma_2 x_2}}{1 - e^{-\gamma_2}}; & \gamma_2 = 1.1; \\ U_3(x_3) = \dfrac{1 - e^{-\gamma_3 x_3}}{1 - e^{-\gamma_3}}; & \gamma_3 = 1.2; \\ U_4(x_4) = c_4 \ln(x_4) - b_4; & c_4 = 2.0223;\ b_4 = 16.73; \\ k_1 = 0.85; \quad k_{12} = 0.03; \quad k_{13} = 0.01; \quad k_{14} = 0.05; & \\ k_{123} = 0.03; \quad k_{124} = 0.03; \quad k_{134} = 0.03; \quad k_{1234} = -0.03. & \end{array} \quad (20)$$

5.1.3 Ranking of the alternatives under the utility independence assumption

Evaluating the alternatives of the deep borehole filter restoration problem by the multilinear form in (20) we rank-order the alternatives as shown in the Table 5.2. As we can see, now the physical method is ranked as number 1.

Table 5.2: Rank-order of the alternatives under UI assumption

Rank-order	Alternative	Expected Utility
#1	Physical method	0.5317
#2	Pneumoimpulsive method	0.5315
#3	Chemical method	0.5306
#4	Airflow method	0.5251
#5	Redrill	0.4015

5.2 MULTIATTRIBUTE ANALYSIS UNDER PARTIAL UTILITY INDEPENDENCE

5.2.1 Decomposition of the MUF for the deep borehole filter restoration problem

Consider the UDM in Figure 5.4 assessed from the DM for the deep borehole filter restoration problem.

	X_1	X_2	X_3	X_4
X_1	○			
X_2	○	○	○	○
X_3	○	○	○	○
X_4				○

Figure 5.4: The utility dependence matrix assessed from the decision maker

The UDM implies the following UI conditions.

- The row of attribute X_1 shows:

$$U(x_1 \mid x_2, x_3, x_4) = U(x_1 \mid x_2^0, x_3^0, x_4^0) = U(x_1 \mid x_2^*, x_3^*, x_4^*),$$

- The row of attribute X_4 shows:

$$U(x_4 \mid x_1, x_2, x_3) = U(x_4 \mid x_1^0, x_2^0, x_3^0) = U(x_4 \mid x_1^*, x_2^*, x_3^*).$$

Using (7), we first expand the MUF around X_1:

$$\begin{aligned} U(x_1, x_2, x_3, x_4) = {} & U(x_1^*, x_2, x_3, x_4) U(x_1 \mid x_2, x_3, x_4) \\ & + U(x_1^0, x_2, x_3, x_4) \bar{U}(x_1 \mid x_2, x_3, x_4). \end{aligned} \tag{21}$$

Then, we expand around attribute X_4:

$$\begin{aligned} U(x_1, x_2, x_3, x_4) = {} & U(x_1^*, x_2, x_3, x_4^*) U(x_1 \mid x_2, x_3, x_4) U(x_4 \mid x_1^*, x_2, x_3) \\ & + U(x_1^*, x_2, x_3, x_4^0) U(x_1 \mid x_2, x_3, x_4) \bar{U}(x_4 \mid x_1^*, x_2, x_3) \\ & + U(x_1^0, x_2, x_3, x_4^*) \bar{U}(x_1 \mid x_2, x_3, x_4) U(x_4 \mid x_1^0, x_2, x_3) \\ & + U(x_1^0, x_2, x_3, x_4^0) \bar{U}(x_1 \mid x_2, x_3, x_4) \bar{U}(x_4 \mid x_1^0, x_2, x_3) \end{aligned} \tag{22}$$

Due to the attribute dominance condition ($U(x_1, x_2, x_3, x_4) = 0$, if $x_1 = x_1^0$), we truncate (22) as

$$\begin{aligned} U(x_1, x_2, x_3, x_4) = {} & U(x_1^*, x_2, x_3, x_4^*)U(x_1 \mid x_2, x_3, x_4)U(x_4 \mid x_1^*, x_2, x_3) \\ & + U(x_1^*, x_2, x_3, x_4^0)U(x_1 \mid x_2, x_3, x_4)\bar{U}(x_4 \mid x_1^*, x_2, x_3) \end{aligned} \tag{23}$$

We now incorporate UI conditions assessed from the DM into (23) and have

$$\begin{aligned} U(x_1, x_2, x_3, x_4) = {} & U(x_1^*, x_2, x_3, x_4^*)U(x_1 \mid x_2^0, x_3^0, x_4^0)U(x_4 \mid x_1^0, x_2^0, x_3^0) \\ & + U(x_1^*, x_2, x_3, x_4^0)U(x_1 \mid x_2^0, x_3^0, x_4^0)\bar{U}(x_4 \mid x_1^0, x_2^0, x_3^0) \end{aligned} \tag{24}$$

For the DM's convenience and due to the property of UI, we can incorporate UI conditions into (23) as:

$$\begin{aligned} U(x_1, x_2, x_3, x_4) = {} & U(x_1^*, x_2, x_3, x_4^*)U(x_1 \mid x_2^0, x_3^0, x_4^0)U(x_4 \mid x_1^*, x_2^0, x_3^0) \\ & + U(x_1^*, x_2, x_3, x_4^0)U(x_1 \mid x_2^0, x_3^0, x_4^0)\bar{U}(x_4 \mid x_1^*, x_2^0, x_3^0) \end{aligned} \tag{25}$$

or even as:

$$\begin{aligned} U(x_1, x_2, x_3, x_4) = {} & U(x_1^*, x_2, x_3, x_4^*)U(x_1 \mid x_2^*, x_3^*, x_4^*)U(x_4 \mid x_1^*, x_2^*, x_3^*) \\ & + U(x_1^*, x_2, x_3, x_4^0)U(x_1 \mid x_2^*, x_3^*, x_4^*)\bar{U}(x_4 \mid x_1^*, x_2^*, x_3^*). \end{aligned} \tag{26}$$

As we can see, the required assessments are: $U(x_1^*, x_2, x_3, x_4^{*0}), U(x_1, x_2^*, x_3^*, x_4^*), U(x_1^*, x_2^*, x_3^*, x_4)$.

5.2.2 Assessments of the MUF for the deep borehole filter restoration problem

This subsection describes results of the assessments of the MUF under partial UI and attributes dominance conditions. As we found previously, the assessments required for the construction of the MUF under the partial UI include two assessments of size two, namely: $U(x_1^*, x_2, x_3, x_4^{*0})$, and $U(x_1 \mid x_2^*, x_3^*, x_4^*), U(x_4 \mid x_1^*, x_2^*, x_3^*)$. The last two utility functions are equivalent to the one-attribute utility functions of attributes X_1 discussed earlier in Chapter 3 and X_4, discussed in section 5.1. So, we have:

$$U(x_1 \mid x_2^*, x_3^*, x_4^*) = \ln(x_1 + 199,000) - 11.8 \tag{27}$$

$$U(x_4 \mid x_1^*, x_2^*, x_3^*) = 2.0223\ln(x_4) - 16.73 \tag{28}$$

We denote two cases for the assessment of $U(x_1^*, x_2, x_3, x_4^{*0})$:

Case 1: $U(x_1^*, x_2, x_3, x_4^*)$

Case 2: $U(x_1^*, x_2, x_3, x_4^0)$

We first assess data points for each of the two cases (the test for Case 1 is partially shown in Appendix I) and determine the value functions:

- for Case 1: $V_1(x_2, x_3) = 0.46(x_2^{0.76} + 0.8x_3^{0.76})^{1/0.76}$,
- for Case 2: $V_2(x_2, x_3) = 0.546(x_2^{0.76} + 0.58x_3^{0.76})^{1/0.76}$.

These functions were curve fitted by using the least-square error approach to $V(x_2, x_3) = c(x_2^a + \eta x_3^a)^{1/\alpha}$, $0 < \alpha < 1$, type of functions for Case 1 when $(x_1 = x_1^*, x_4 = x_4^*)$ and then for Case 2 when $(x_1 = x_1^*, x_4 = x_4^0)$. Figure 5.5 illustrates the difference in trade-offs between these two attributes for our DM in the form of isopreference contours (Matheson and Abbas, 2005; Abbas and Matheson, 2010; Abbas, 2011a).

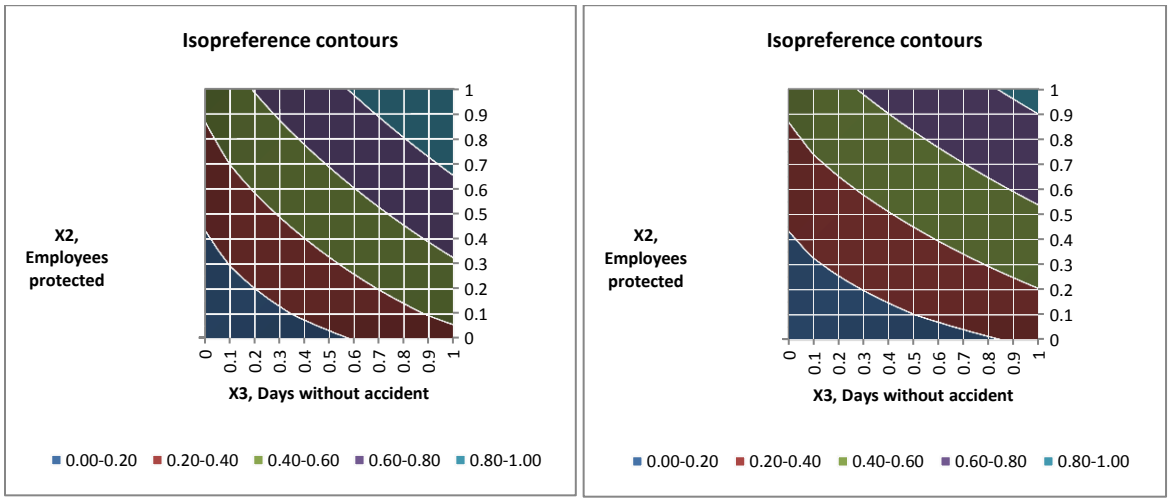


Figure 5.5: The (a) case 1, and (b) case 2.

We then assessed from the DM conditional utility functions over these value functions as (the assessments for Case 1 are shown in Appendix J)

- for Case 1: $U(x_2, x_3 \mid x_1^*, x_4^*) = U_{V1}(V_1(x_2, x_3)) = U_{V1}(0.46(x_2^{0.76} + 0.8x_3^{0.76})^{1/0.76})$,
- for Case 2: $U(x_2, x_3 \mid x_1^*, x_4^0) = U_{V2}(V_2(x_2, x_3)) = U_{V2}(0.546(x_2^{0.76} + 0.58x_3^{0.76})^{1/0.76})$.

In both cases, utility functions were exponential and were curve fitted utilizing the least-square error approach

- for Case 1: $U_{V1}(V_1(x_2, x_3)) = \dfrac{1 - e^{-1.1(V_1(x_2, x_3))}}{1 - e^{-1.1}}$,
- for Case 2: $U_{V2}(V_2(x_2, x_3)) = \dfrac{1 - e^{-1.6(V_2(x_2, x_3))}}{1 - e^{-1.6}}$.

Since the conditional utility function $U(x_2, x_3 \mid x_1^*, x_4^*)$ is normalized between zero and one as

$$U(x_2, x_3 \mid x_1^*, x_4^*) = \frac{U(x_1^*, x_2, x_3, x_4^*) - U(x_1^*, x_2^0, x_3^0, x_4^*)}{U(x_1^*, x_2^*, x_3^*, x_4^*) - U(x_1^*, x_2^0, x_3^0, x_4^*)},$$

we can easily find

$$U(x_1^*, x_2, x_3, x_4^*) = [U(x_1^*, x_2^*, x_3^*, x_4^*) - U(x_1^*, x_2^0, x_3^0, x_4^*)]U(x_2, x_3 \mid x_1^*, x_4^*) + U(x_1^*, x_2^0, x_3^0, x_4^*).$$

Similarly,

$$U(x_1^*, x_2, x_3, x_4^0) = [U(x_1^*, x_2^*, x_3^*, x_4^0) - U(x_1^*, x_2^0, x_3^0, x_4^0)]U(x_2, x_3 \mid x_1^*, x_4^0) + U(x_1^*, x_2^0, x_3^0, x_4^0).$$

Now, the maximum and minimum values of $U(x_1^*, x_2, x_3, x_4^*)$ should be $U(x_1^*, x_2^*, x_3^*, x_4^*) = 1$ and $U(x_1^*, x_2^0, x_3^0, x_4^*) = 0.90$, respectively. The maximum and minimum values of $U(x_1^*, x_2, x_3, x_4^0)$ should be $U(x_1^*, x_2^*, x_3^*, x_4^0) = 0.92$ and $U(x_1^*, x_2^0, x_3^0, x_4^0) = 0.85$, respectively. Recall, the latter three were assessed from the DM during the determination of the scaling constants of the multilinear form (see Appendix H).

Hence,

$$U(x_1^*, x_2, x_3, x_4^*) = (1-0.9)U(x_2, x_3 \mid x_1^*, x_4^*) + 0.9 =$$
$$0.1U_{V1}(V_1(x_2, x_3)) + 0.9 = 0.1\left(\frac{1-e^{-1.1(V_1(x_2,x_3))}}{1-e^{-1.1}}\right) + 0.9.$$

Similarly,

$$U(x_1^*, x_2, x_3, x_4^0) = (0.92-0.85)U(x_2, x_3 \mid x_1^*, x_4^0) + 0.85$$
$$= 0.07U_{V2}(V_2(x_2, x_3)) + 0.85 = 0.07\left(\frac{1-e^{-1.6(V_2(x_2,x_3))}}{1-e^{-1.6}}\right) + 0.85.$$

Substituting these assessments into (26), we can construct the multiattribute utility function under PUI conditions:

$$U(x_1, x_2, x_3, x_4) = U(x_1 \mid x_2^*, x_3^*, x_4^*)[U(x_1^*, x_2, x_3, x_4^*)U(x_4 \mid x_1^*, x_2^*, x_3^*) + U(x_1^*, x_2, x_3, x_4^0)(1-U(x_4 \mid x_1^*, x_2^*, x_3^*))],$$

where

$$\begin{aligned}
&U(x_1 \mid x_2^*, x_3^*, x_4^*) = \ln(x_1 + a_1) - b_1; && a_1 = 199{,}000;\ \ b_1 = 11.8;\\
&U(x_1^*, x_2, x_3, x_4^*) = \frac{1-e^{-\gamma_{case1}\left(c_1(x_2^\alpha + \eta_1 x_3^\alpha)^{1/\alpha}\right)}}{1-e^{-\gamma_2}}; && \gamma_{case1} = 1.1;\ \ c_1 = 0.46;\ \ \alpha = 0.76;\ \ \eta_1 = 0.8;\\
&U(x_1^*, x_2, x_3, x_4^0) = \frac{1-e^{-\gamma_{case2}\left(c_2(x_2^\alpha + \eta_2 x_3^\alpha)^{1/\alpha}\right)}}{1-e^{-\gamma_3}}; && \gamma_{case2} = 1.6;\ \ c_2 = 0.546;\ \ \eta_2 = 0.58.\\
&U(x_4 \mid x_1^*, x_2^*, x_3^*) = c_4 \ln(x_4) - b_4; && c_4 = 2.0223;\ \ b_4 = 16.73.
\end{aligned} \tag{29}$$

In (29), both X_2 and X_3 are normalized between zero and one.

5.2.3 Ranking of the alternatives with assessed partial utility independence conditions

Calculating expected utilities for each method by (29) for each of the five branches of the decision tree in Figure 1.9 we found the ranking of the alternatives as shown in Table 5.3.

Recall, we had different rankings after the analysis of the problem under the UI assumption. As we see, the rank orders of two methods (physical and pneumoimpulsive) are switched and rank orders of three other methods remain the same. This way, the MUF in (29) addresses the DM's preferences.

Table 5.3: Rank-order of the alternatives under PUI conditions

Rank-order	Alternative	Expected Utility
#1	Pneumoimpulsive method	0.5336
#2	Physical method	0.5334
#3	Chemical method	0.5330
#4	Airflow method	0.5270
#5	Redrill	0.4039

5.3 DISCUSSION OF THE RESULTS

We now compare the results of the profit analysis, the UI assumption, and the PUI approach in Table 5.4.

Table 5.4: Rank-order of the alternatives according to three different approaches

Rank-order	Approaches		
	Profit analysis	MUF under UI assumptions	MUF under PUI conditions
#1	Pneumoimpulsive	*Physical*	Pneumoimpulsive
#2	*Chemical*	*Pneumoimpulsive*	Physical
#3	*Physical*	Chemical	Chemical
#4	Airflow	Airflow	Airflow
#5	Redrill	Redrill	Redrill

The evaluation of the alternatives of the borehole recovery problem considering only profit shows that the pneumoimpulsive method is the least expensive and the redrill alternative is the most expensive. However, in profit analysis, we modeled the annual profit from the extraction borehole by varying only Q and I in (3). In other words, we set $N, T,$ and C to their base values since they contribute to less than 10% of the variance in the profit.

The ranking of the alternatives under the UI assumption shows slightly different rankings, but the two worse alternatives according to the UI approach and the profit analysis are the same.

We now consider the results with the assessed PUI conditions and the attribute dominance condition. Based on the decision maker's true preferences, the best alternative is the pneumoimpulsive method and the redrill alternative remains the worst. The physical method is the 2nd most preferable, followed by the chemical and the airflow methods. As we can see, the profit analysis and the assumption of UI do not provide exactly the same rank-order of the restoration methods as the approach with the true preferences of the DM. Utilizing these approaches, only three out of the five alternatives match the ranking determined by the MUF under PUI assertions. This means, for example, that the DM cannot always approximate their real MUF with the multilinear form under UI assumptions.

5.4 SUMMARY

In this Chapter we have evaluated the multiattribute decision problem utilizing two different approaches: (i) multiattribute analysis under UI assumptions, and (ii) multiattribute analysis under PUI conditions. The results indicated that the ranking of the alternatives under the assumption of UI (multilinear form) is different from the ranking under PUI assertions assessed from the DM.

The multilinear form is ideal when utility independence (UI) holds, but if the DM is uncomfortable with UI, we need to address PUI conditions. However, PUI may have complications in the assessment of the MUF. Our results also indicate that the proposed method of partial utility independence is slightly difficult for the assessment compared to the UI approach, but represents the true preferences of the DM.

The ranking of the alternatives of the deep borehole filter restoration problem determined in this work should not be considered as a policy for every company in the uranium extraction industry. Our results are specific for a concrete uranium ore deposit, the parameters of which

may differ from parameters of other uranium ore deposits. However, once a model is built for a concrete uranium ore deposit, it should clarify the way the decisions are made. We demonstrated that the evaluation of the alternatives of the borehole recovery problem based on profit only, or with the assumption of utility independence of the attributes may lead to different decisions 40% of the time compared with the true preferences of the decision maker.

CHAPTER 6: AN ALGORITHM FOR EXCLUDING REDUNDANT ASSESSMENTS

This Chapter addresses the problem of determination of the minimal set of utility assessments required for a construction of a MUF by developing an algorithm for excluding duplicate and redundant assessments in a multiattribute utility function with partial utility independence. This Chapter presents a ternary matrix representation of utility assessments and the Twos-Complement Exclusion algorithm, which were introduced in (Abdildin and Abbas, 2012). The detailed pseudocode of the algorithm is also provided in this Chapter.

The recent development of decision analysis algorithms for multiattribute problems under partial utility independence motivated the construction of the exclusion algorithms. For example, the Twos-Complement Exclusion algorithm was motivated by the Iterative Decomposition algorithm (Abbas, 2010a) which enables a decomposition of a complex multiattribute utility function (MUF) into lower order terms. These terms (or utility assessments) are the building blocks of the MUF and must be assessed from the decision maker. The decomposition of a MUF often comes with the problem of having some redundant utility assessments, which means extra set of questions to the decision maker. Question-selection algorithms to elicit utility values based on entropy methods were discussed in (Abbas, 2004; Abbas, 2005; Abbas, 2006). To avoid redundant utility assessments, practitioners either try to find utility independence conditions and apply the multilinear form of MUF or minimize the number of attributes of a decision problem. Lower number of attributes enables the elimination of the redundant utility assessments by inspection. In more complex problems, they need a tool for elimination of duplicate and redundant utility assessments.

While in different contexts, the problem of elimination of duplicates and redundancies can be found, for example, in computer science and biomedical science.

In compiler optimization, the elimination of expressions that are redundant on some paths through a program is an important problem. Here, the redundancy or partial redundancy implies unnecessary computations, e.g. computations which are performed twice. One can find many algorithms for redundancy elimination in this context (see, for example, Morel and Renvoise, 1979; Knoop et al., 1992; Paleri et al., 2003).

In biomedical science, gene ontology (GO) analysis implies the interpretation and summarization of biological functions. Broadly speaking, due to "interdependence between terms in the gene ontology hierarchy," some terms are redundant (i.e. "provide little additional information") and must be eliminated from enriched GO lists (Jantzen et al., 2011).

In the decision analysis literature, however, we could not find exclusion algorithms. Therefore, this Chapter will present two versions of the exclusion algorithm; the brute-force version will be implemented only for comparison purposes with our Twos-Complement Exclusion algorithm. In Chapter 7, we will introduce the Ternary-Decimal Exclusion algorithm.

6.1 TERNARY MATRIX REPRESENTATION OF UTILITY ASSESSMENTS

Motivating Example. Suppose we have an assertion of the following partial utility independence conditions:

$$(X_3\ UI\ X_4 \mid X_1, X_2) \text{ and } (X_4\ UI\ X_2, X_3 \mid X_1).$$

Using a normalized conditional utility function, we can express these conditions as

- Attribute X_3 asserts: $U(x_3 \mid x_1, x_2, x_4) = U(x_3 \mid x_1, x_2, x_4^0)$.

- Attribute X_4 asserts: $U(x_4 \mid x_1, x_2, x_3) = U(x_4 \mid x_1, x_2^0, x_3^0)$.

Expanding the MUF through the attributes X_3 and X_4 by (7) and incorporating the given UI assertions into the basic expansion Theorem (Abbas, 2010a), we have

$$\begin{aligned} U(x_1, x_2, x_3, x_4) = & U(x_1, x_2, x_3^*, x_4^*)U(x_3 \mid x_1, x_2, x_4^0)U(x_4 \mid x_1, x_2^0, x_3^0) \\ & + U(x_1, x_2, x_3^*, x_4^0)U(x_3 \mid x_1, x_2, x_4^0)\bar{U}(x_4 \mid x_1, x_2^0, x_3^0) \\ & + U(x_1, x_2, x_3^0, x_4^*)\bar{U}(x_3 \mid x_1, x_2, x_4^0)U(x_4 \mid x_1, x_2^0, x_3^0) \\ & + U(x_1, x_2, x_3^0, x_4^0)\bar{U}(x_3 \mid x_1, x_2, x_4^0)\bar{U}(x_4 \mid x_1, x_2^0, x_3^0). \end{aligned} \quad (30)$$

Excluding duplicate assessments from (30), gives these six terms: $U(x_3 \mid x_1, x_2, x_4^0)$, $U(x_4 \mid x_1, x_2^0, x_3^0)$, $U(x_1, x_2, x_3^*, x_4^*)$, $U(x_1, x_2, x_3^*, x_4^0)$, $U(x_1, x_2, x_3^0, x_4^*)$, and $U(x_1, x_2, x_3^0, x_4^0)$. We would like to represent these terms using utility assessments and not conditional utility assessments.

A convenient representation of the utility assessments required in (30) is the tree representation shown in Figure 6.1(a). The tree representation was introduced in (Abbas, 2010a). Recall, equation (3) requires the following three utility assessments based on UI assertion of the attribute X_3: $U(x_1, x_2, x_3^0, x_4)$, $U(x_1, x_2, x_3^*, x_4)$, and $U(x_3 \mid x_1, x_2, x_4^0)$. These three terms respectively correspond to $U(x_1, x_2, x_3^0, x_4)$, $U(x_1, x_2, x_3^*, x_4)$, $U(x_1, x_2, x_3, x_4^0)$ in the first branch in the tree. Notice that the first two utility assessments are derived from $U(x_1, x_2, x_3, x_4)$ by replacing the expanding attribute X_3 into its least and most preferable instantiations, respectively. The third term, $U(x_1, x_2, x_3, x_4^0)$, incorporates UI conditions asserted by the expanded attribute, X_3. By induction, we expand further these three terms incorporating the UI conditions provided by the attribute X_4. The leaves of the expanded tree contain the list of seven utility assessments as illustrated in Figure 6.1(b). As we mentioned, however, there are only six

independent assessments for this MUF. How do we exclude the duplicate and redundant assessments?

Obviously, the 6th term from the top of the list in Figure 6.1(b), namely, $U(x_1, x_2^0, x_3^0, x_4)$, is a *duplicate* of the 3rd one and must be deleted. The closer look on other terms of the list shows that the 2nd and the 5th utility assessments are *redundant* due to the 1st term, because they are subsets of the 1st one, i.e. they are "covered" by the 1st term. Notice, the only difference between them is in attribute X_3, which is in its "full" state in the 1st utility function, while it is in states x_3^0 and x_3^* in the 2nd and the 5th terms, respectively. Therefore, we should exclude the 2nd, $U(x_1, x_2, x_3^0, x_4^0)$, and the 5th, $U(x_1, x_2, x_3^*, x_4^0)$, utility assessments. There are no other duplicate and redundant terms: the four utility assessments that are required for the MUF are shown in bold rectangles in Figure 6.1(a). However, the larger the number of functions and attributes of the decision problem, the larger the complexity of the process of checking and eliminating redundancy. This requires an automation of the process. The question is what would be the best representation of the list of functions as in Figure 6.1(b), and how the list can be checked efficiently?

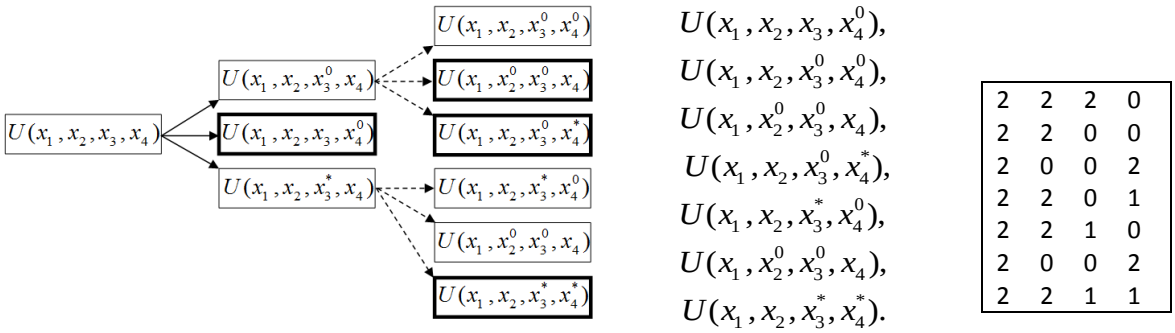


Figure 6.1: (a) Expansion tree for a four-attribute problem (b) Utility assessments, and their (c) Ternary matrix representation, M

In order to automate the process of elimination of duplicate and redundant assessments from

the list of functions, we need some reliable and simple representation of utility functions. Such representation should be compact, memory efficient, and intuitive. We found that a *ternary matrix representation* can serve for this purpose.

Definition 8: *The ternary matrix of utility assessments, **M**, is an* $m \times n$ *matrix storing integers 0, 1, and 2, representing respectively the least preferable value, the most preferable value, and all values of an attribute, where* m *is the number of utility assessments, and* n *is the number of attributes of the decision problem.*

The ternary matrix representation is based on the base three number system, in which ternary digits, or trits (analogous to bits in the binary number system) can be exactly in one of the three states: 0, 1, or 2. A very good analysis of the base-3 number system can be found here (Hayes, 2001). Compared with the *balanced ternary representation* (Knuth, 2000), which uses -1, 0, and 1 as trits, the *ordinary ternary notation* ideally matches to our needs. For example, we can represent the instantiation, x^0, the least preferable value of an attribute as 0, x^*, the most preferable value as 1, and, x, which represents all values, as 2. By definition, the following two conditions must hold for any instantiation of an attribute: (i) 2s include both 1s and 0s, and (ii) 1s and 0s are distinct.

We can now encode any utility assessment into a sequence of zeros, ones, and twos, or a *ternary vector* (Knuth, 2011). For example, the first utility function in Figure 6.1(b), $U(x_1, x_2, x_3, x_4^0)$, will be represented by the ternary vector: 2 2 2 0. Similarly, we can encode other six functions and create a 7×4 ternary matrix **M** as shown in Figure 6.1(c). The ternary matrix representation significantly simplifies the comparison of elements of **M** since 2 > 1, 2 > 0, and $1 \neq 0$. Some other applications of the ternary vector we have found in the literature include a three-valued logic, where the logical values "false," "true," and "maybe," can be denoted

respectively by 0, 1, and * (Knuth, 2011).

In general, given a list of m functions of n attributes, we will represent the list as $m \times n$ ternary matrix. Although there might be representations other than the matrix form, the ternary matrix representation gives us the flexibility in developing algorithms and connecting them to other methods in decision analysis. We now ready to present the exclusion algorithm.

6.2 EXCLUSION ALGORITHM

Having encoded the given list of utility functions into the ternary matrix **M** as shown in Figure 6.1(c), we can easily exclude the possible duplicate and redundant utility assessments as following. First, we develop an algorithm that eliminates the duplicate and redundant rows from matrix **M**, dynamically adjusting its size in the process. Then, we decode the resulting output matrix **M** back into the list of utility assessments. In this Section, we present two versions of the exclusion algorithm: (i) the naïve (brute-force) approach, and (ii) the twos-complement approach.

6.2.1 Brute-force approach

The naïve (brute-force) exclusion algorithm is straightforward; it compares each row of the ternary matrix **M** with all other rows element-wise and excludes duplicate and redundant rows as they are found. If all elements of the 1st (top) row of **M** are greater than or equal to the corresponding elements of the 2nd row, then the 2nd row can be deleted from **M** as either duplicate or redundant, namely:

(i) if both rows are identical, then row two is duplicate;

(ii) if all elements of the 1st row, which are greater than the corresponding elements of the 2nd row, are twos, then the 2nd row is redundant.

Figure 6.2(a) illustrates the comparison of the first two rows of matrix **M** from Figure 6.1(c), where row two is redundant.

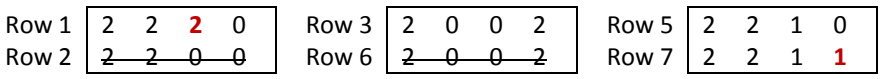


Figure 6.2: (a) Row 2 of matrix M from Figure 6.1(c) is redundant due to row 1, because all elements of row 1 ≥ elements of row 2, and the element (in dark red), which is > than the respective element of row 1, is equal to two; (b) Row 6 is a duplicate of row 3; hence, deleted; (c) Row 7 dominates row 5, but cannot exclude it, because the 4th element of row 7, which is greater than the corresponding element of row 5, is equal to one.

Figure 6.2(b) depicts the exclusion of row six that is a duplicate of row three. Notice, however, if we compare rows five and seven as shown in Figure 6.2(c), the former does not dominate the latter, but the latter dominates the top row; hence, the order is important. Also notice, even that row seven dominates row five element-wise, the latter cannot be eliminated from the matrix, because the element of the dominating row that is greater than the corresponding element of the dominated row is equal to one.

Table 6.1: High-level pseudocode of algorithm exclusionV1

```
Algorithm exclusionV1(M):
Input: An m×n ternary matrix M storing integers: 0, 1, 2.
Output: The matrix M without duplicate and redundant rows.
1:  Set index r to be the 1st (unprocessed) row of the matrix M.
2:  WHILE an unprocessed row of M exists DO compare all other rows of M with row r as follows:
3:    Set index i to be the 1st (uncompared) row of the matrix M.
4:    WHILE an uncompared row of M exists DO following:
5:      IF i ≠ r THEN
6:        FORALL n: compare rows r and i element-wise and break out of the For loop IF any element of r < i
7:        IF all elements of r are greater than or equal to the corresponding elements of row i THEN
8:          FORALL n:
9:              IF any element of row r, which is greater than the corresponding element of row i, is equal to 1 THEN
10:                 break out of the For loop
11:             ELSE delete row i since it is either duplicate or redundant due to row r
12: RETURN M
```

Table 6.1 presents the high-level pseudocode of the brute-force algorithm, `exclusionV1`. The algorithm contains two nested loops, an outer loop by index *r* and an inner loop by index *i*. Starting with the first top row of the ternary matrix **M**, it goes down row by row through the matrix (*outer loop's row*) comparing every other row (*inner loop's row*) with the current outer

row r, and eliminating the inner row i, if necessary. Notice, in line 6, if any element of row r is less than the respective element of i, we skip row i. The rest of the pseudocode in Table 6.1 is self-explanatory.

We now provide asymptotic analysis (see, for example, Shaffer, 2001 and Cormen et al. 1997) for the running time of algorithm `exclusionV1`. The *running time* is measured by counting the number of primitive operations (see, for example, Goodrich et al. 2004 and Kleinberg and Tardos, 2006) that are executed on the best and worst inputs of the algorithm. An example of the primitive operation is assigning a value to a variable, comparing two numbers, or performing an arithmetic operation as shown in pseudocodes in the following sub-sections. The best input of the exclusion algorithm is considered to be a ternary matrix with only 2s in its first top row. In the worst case, the input matrix will not contain any duplicate and redundant rows; all rows of matrix **M** can be unique.

Proposition 2. *The growth rate for the running time of algorithm* exclusionV1 *for excluding duplicate and redundant assessments (rows) from an* $m \times n$ *ternary matrix of utility assessments is* $\Theta(nm)$ *in the best case and is* $\Theta(nm^2)$ *in the worst case.*

6.2.2 Twos-Complement method

Similar to the brute-force algorithm, the *Twos-Complement Exclusion algorithm* eliminates duplicate and redundant rows of the ternary matrix **M** by comparing each row of the matrix with all other rows. Unlike the brute-force approach, however, the twos-complement method compares the rows of **M** partly. In addition, if a full row of twos is found in the matrix, the twos-complement algorithm immediately terminates returning **M** with a single row of twos.

The main idea of the twos-complement approach is the following. Since twos in **M** include ones and zeros by definition, when comparing row *i* with row *r* element-wise, the twos-complement algorithm makes the comparison only by the elements, whose values in row *r* are the *complements of twos*, i.e. ones and zeros. Hence, starting with the 1st (top) row, search for elements that are complements of 2, i.e. 0 or 1. Identify their corresponding columns. Any other row that has identical elements in the corresponding columns is duplicate or redundant. Figure 6.3(a, b) demonstrates exclusion of rows two and five from matrix **M** due to their redundancy with row one. Figure 6.3 (c) shows the elimination of row six as a duplicate of row three.

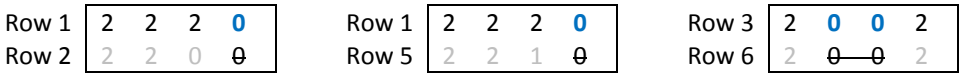


Figure 6.3. (a) Row 2 is redundant due to row 1, because the element of row 2 in the 4th column is equal to the element (in blue) in the twos-complement column of row 1; hence, eliminated; (b) Row 5 is redundant due to row 1; therefore, deleted; (c) Row 6 duplicates row 3 and will be excluded, because its elements in columns 2 and 3 are identical to the elements of row 3.

The algorithm first stores the indices of twos-complement elements of row *r* in a vector, `twos_compl`.

Definition 9: *The indices of columns of the ternary matrix **M**, which do not contain 2s in the current row r being processed, create a vector of twos' complements,* twos_compl, *for row r.*

Then, the algorithm compares only parts of the rows *r* and *i*, by the columns, which indices are found in the `twos_compl` vector of row *r*, and excludes row *i*, if necessary. Hence, the name of the algorithm is the "twos-complement exclusion algorithm."

Table 6.2 presents the high-level pseudocode of the twos-complement exclusion algorithm exclusionV2 (the more detailed pseudocode is in the Appendix). The algorithm creates vector twos_compl in line 4 to store indices of 0s and 1s of the current outer row *r*. In line 5 it returns a

vector of twos and terminates the program if twos_compl is empty; otherwise, it compares elements of rows r and i by the indices in twos_compl and, if all of them are pairwise identical, removes the row i. The twos-complement exclusion algorithm minimizes the comparison of all elements of the rows of matrix **M**, and it is even more efficient when attributes are mostly utility dependent (more twos in the rows of the ternary matrix). The extra space requirement of exclusionV2 due to vector twos_compl is in $O(n)$ in the worst case.

Table 6.2. High-level pseudocode of algorithm exclusionV2

```
Algorithm exclusionV2(M):
Input: An m x n ternary matrix M storing integers: 0, 1, 2.
Output: The matrix M without duplicate and redundant rows.
1:    Initialize a vector twos_compl to store indices of 1s and 0s of row r.
2:    Set index r to be the 1st (unprocessed) row of matrix M.
3:    WHILE an unprocessed row of M exists DO compare all other rows of M with row r as follows:
4:        FORALL n: IF an element of row r is not equal to two THEN set its (column) index into vector twos_compl.
5:        IF twos_compl is empty THEN delete matrix M, return a vector of n twos, and terminate the program.
6:        Set index i to be the 1st (uncompared) row of matrix M.
7:        WHILE an uncompared row of M exists DO following:
8:            IF i ≠ r THEN
9:                IF any corresponding elements of rows r and i indexed in twos_compl are different THEN next i.
10:               ELSEIF all of them are equal THEN delete row i since it is either redundant or duplicate
11:   RETURN M
```

Proposition 3. *The twos-complement exclusion algorithm* exclusionV2 *runs in* $\Theta(n)$ *in the best case and in* $\Theta(nm^2)$ *in the worst case.*

Sketch of the Proof:

Correctness: The input $m \times n$ ternary matrix **M** either contains a vector of 2s or does not contain it. In the former case, `exclusionV2` returns the vector of 2s and terminates (line 5). In the latter case, all possible duplicate and redundant rows will be removed from **M** in line 10, and the algorithm terminates in line 11 returning the updated matrix **M**. Hence, the algorithm works correctly for any input.

Running time: The best input of `exclusionV2` should contain the vector of 2s in the first (top) row of matrix **M**; that is, `twos_compl` of the 1st row should be empty. In line 4, the algorithm

first creates the vector `twos_compl` for the current row *r*, which requires *n* comparison operations. Then it checks the vector `twos_compl` for emptiness and, since `twos_compl` is empty, deletes matrix **M**, returns the vector of 2s, and terminates (line 5). Therefore, `exclusionV2` runs in $\Theta(n)$ in the best case.

In the worst case, the input matrix will not contain the vector of 2s at all; all rows of matrix **M** can be unique. The algorithm `exclusionV2` has two nested loops in lines 3 and in line 7, each repeated $m+1$ times at most; and their bodies in lines 4-10 and 8-10, respectively, each repeated m times at most. This would require $\Theta(m^2)$ row comparisons for the number of elements indicated by the size of vector `twos_compl`, which is in $\Theta(n)$ in the worst case; therefore, the growth rate for the running time of the twos-complement algorithm `exclusionV2` is in the set $O(nm^2)$, in the worst case. Analogously, the worst-case time complexity of `exclusionV2` is in the set $\Omega(nm^2)$. Since the growth rate is in $O(nm^2)$ and it is in $\Omega(nm^2)$, it is $\Theta(nm^2)$ in the worst case. Q.E.D.

Although, both algorithms have, similar up to constant factors, asymptotic upper and lower bounds in the worst case, the twos-complement algorithm outperforms the brute-force algorithm in simulations, as shown in the next section. In addition, twos-complement algorithm asymptotically outperforms the brute-force algorithm for the best-case input. Moreover, the larger the number of twos in rows of the input ternary matrix **M**, the smaller the number of operations that the twos-complement algorithm demands compared to the brute-force algorithm. This means that the twos-complement algorithm works faster especially when the attributes of the decision problem provide a milder set of UI conditions, i.e. the problem has higher complexity. We now provide simulation analyses of the brute-force and twos-complement algorithms.

6.2.3 The detailed pseudocodes of both approaches

We now present the detailed pseudocodes of the both algorithms. Implementation in C++ can be found in Appendix K.

The detailed pseudocode of the brute-force approach.

```
Algorithm exclusionV1(M):
Input: An m x n ternary matrix M storing integers: 0, 1, 2.
Output: The matrix M without duplicate and redundant rows.
1:      rows ← m
2:      r ← 0                   // index of outer loop's row
3:      while r < rows do
4:              i ← 0           // index of inner loop's row
5:              while i < rows do
6:                      if i ≠ r then
7:                              flgGE ← true                    // flag "Greater or Equal"
8:                              for j = 0 to n-1 do
9:                                      if M[r][j] < M[i][j] then
10:                                             flgGE ← false
11:                                             break                   // break out of the for loop
12:                             if flgGE = true then
13:                                     redundant ← true  // flag "redundant"
14:                                     for j  = 0 to n-1  do
15:                                             if M[r][j] > M[i][j] & M[r][j] = 1 then
16:                                                     redundant ← false
17:                                                     break     // row i is not redundant
18:                                     if redundant = true then
19:                                             M.erase(M.begin() + i) // delete row i
20:                                             rows ← rows - 1
21:                                             if i < r then   // if row i was before row r
22:                                                     r ← r - 1  // shift up rows r and i
23:                                             i ← i - 1
24:                             i ← i + 1
25:             r ← r + 1
26:     return M
```

The detailed pseudocode of the Twos-Complement approach.

```
Algorithm exclusionV2(M):
Input: An m x n ternary matrix M storing integers: 0, 1, 2.
Output: The matrix M without duplicate and redundant rows.
1:      rows ← m
2:      vector twos_compl   // store indices of 0s and 1s of row r
3:      r ← 0               // index of outer loop's row
4:      while r < rows  do
5:          for j ← 0 to n-1 do
6:              if M[r][j] ≠ 2 then
7:                  twos_compl.push_back(j)
8:          if twos_compl.empty() then
9:              vector twos
10:             twos.assign(n,2)
11:             M.clear()
12:             M.push_back(twos)   // return vector of 2s
13:             break               // break out of the WHILE loop
14:         i ← 0       // inner loop index
15:         while i < rows do
16:             if i ≠ r then
17:                 size ←  twos_compl.size()
18:                 indentical ← true                   // flag "identical"
19:                 for j ← 0 to size-1 do
20:                     curr ← twos_compl[j]
21:                     if M[r][curr] ≠  M[i][curr] then
22:                         indentical ← false
23:                         break               // break out of the FOR loop
24:                 if indentical = true then
25:                     M.erase(M.begin() + i) // delete row i
26:                     rows ← rows - 1
27:                     if i < r then   // if row i was before row r
28:                         r ← r - 1  // shift up rows r and i
29:                     i ← i - 1
30:             i ← i + 1
31:         twos_compl.clear() // clear twos_compl of r
32:         r ← r + 1
33:     return M
```

6.3 COMPARISON OF THE RUNNING TIMES: SIMULATION RESULTS

Both versions of the algorithm were implemented in C++ with Microsoft Visual C++ 2010 Express on a PC with the following environment: Intel® Core™ i5-2320, 6GB SDRAM, Windows® 7 Home Premium, 64Bit. The input matrix **M** was implemented as a "vector of vectors:" the sequential representation of functions can be more space efficient than the linked representations (Standish, 1995). For example, representing a utility assessment $U(x_1, x_2, ..., x_n)$ as a vector of short integers requires $2n$ bytes, while one-way linked list representation would

require 6*n* bytes, i.e., 4 bytes for each pointer and 2 bytes for each item of the function. The memory efficiency of the implementation is important for certain applications.

The simulations were done in the following order:

(i) Randomly generate the ternary matrix **M** with the given number of rows *m* and columns *n*.

(ii) Run `exclusionV1` with **M** and accumulate its running time in *Time1*.

(iii) Run `exclusionV2` with **M** and accumulate its running time in *Time2*.

(iv) Repeat steps from (i) to (iii) 10,000 times.

(v) Return average running times of 10,000 tests as *AverageTime1* and *AverageTime2*.

The algorithms were compared in two types of tests. In the first type, we changed the number of functions, m, keeping the number of attributes, n, fixed. In the second, we varied n keeping m fixed. We randomly generated $m \times n$ ternary matrix **M** synchronizing the seed of the random generator with the time function as follows: srand(time(0) + i * 2 + 1), where i is the current test number, which was incremented from 0 to 9,999. This way the random generator provided completely different matrices, which served as input for both versions of the algorithm. The time of matrix generation was not included in the running times of the algorithms.

Figure 6.4 illustrates average running times of 10,000 tests when (a) the number of functions varies from 50 to 250 with step 50 for a seven-attribute decision problem, (b) the number of attributes increases from 3 to 11 for a fixed number of functions, 150. We see that the twos-complement algorithm shows much better performance. Figure 6.5 demonstrates average running times of the algorithms for different input matrices in 2D, and Figure 6.6 in 3D. Again, the twos-complement algorithm shows better running times on the whole domain of m and n.

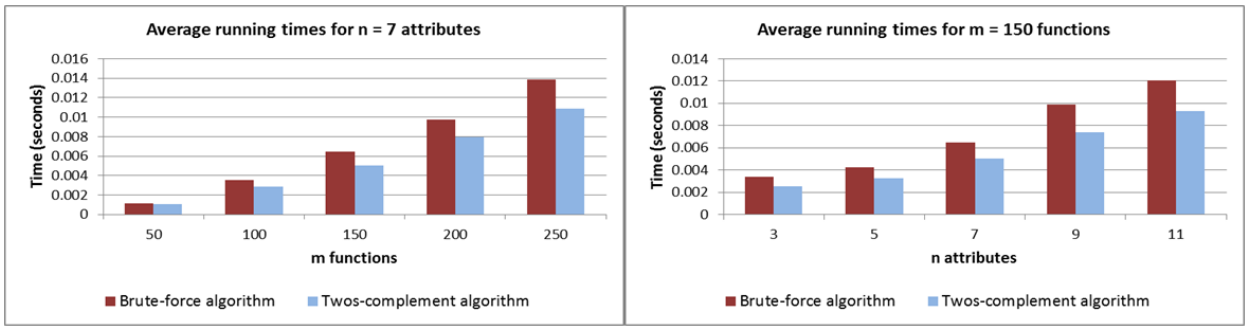


Figure 6.4. Average running times for: (a) fixed number of attributes; (b) fixed number of utility functions.

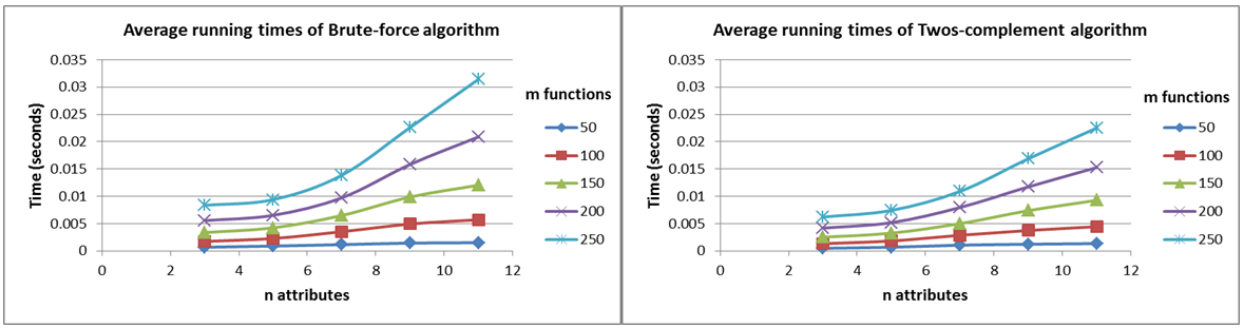


Figure 6.5. Average running times for different number of attributes and functions for: (a) Brute-force algorithm; (b) Twos-complement algorithm

The running time in milliseconds may not provide an appropriate comparison of the algorithms, so we compared algorithms for larger sets of problems. Figure 6.7 illustrates average running times of the algorithms when (a) n changes from 50 to 250 while m is fixed at 5,000; and (b) m varies from 5,000 to 25,000 while n is fixed at 10. As we can see, both algorithms are linear in the number of attributes and quadratic in the number of utility assessments. These results are consistent with our asymptotic analyses. In summary, the algorithms demonstrate efficient running times with considerably better performance of the Twos-Complement algorithm.

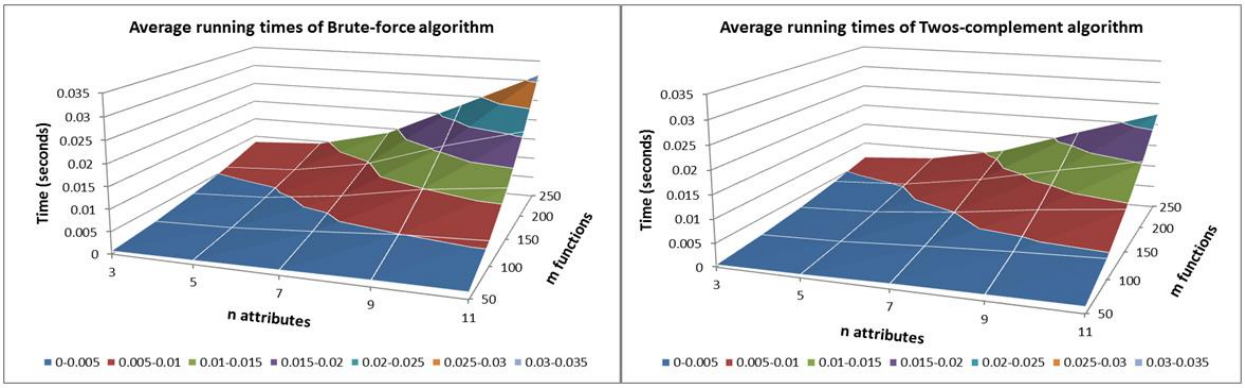


Figure 6.6. Average running times in 3D for: (a) Brute-force algorithm; (b) Twos-complement algorithm

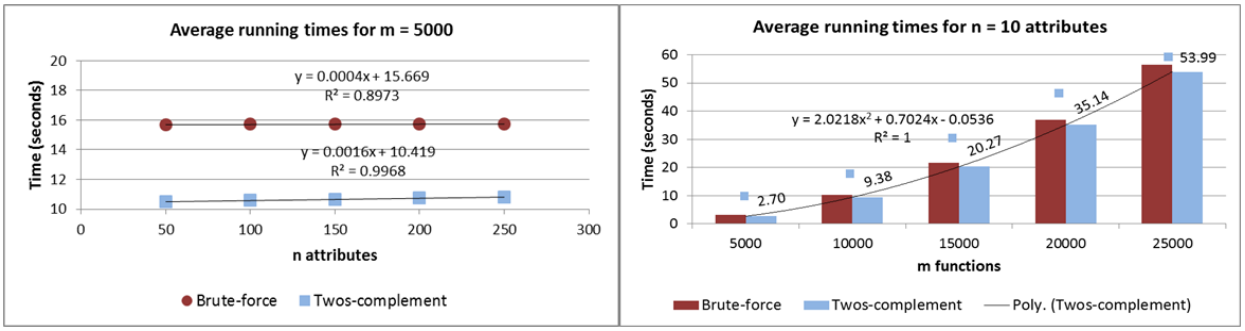


Figure 6.7. Average running times of the algorithms for larger problems with: (a) different number of attributes; (b) different number of functions

6.4 SUMMARY

Elimination of duplicate and redundant utility assessments is an important problem in the field of decision analysis. We presented an efficient method for excluding duplicate and redundant utility assessments in a multiattribute decision problem. We suggested using the base-3 number system to represent the given list of utility assessments in the ternary matrix format. The base-3 number system ideally encodes utility functions used in decision analysis. We presented the twos-complement exclusion algorithm. The asymptotic and simulation analysis of the twos-complement algorithm proves its efficiency.

CHAPTER 7: TERNARY-DECIMAL EXCLUSION ALGORITHM

7.1 INTRODUCTION

The Twos-Complement Exclusion algorithm effectively eliminates redundant utility assessments, and so excludes redundant questions to the decision maker. The main advantage of the algorithm is its simplicity. But can we design a more efficient algorithm? We now introduce a *Ternary-Decimal Exclusion algorithm*.

This approach is fundamentally different in that we transfer the main computational burden from a two-dimensional arena into a single-dimensional one. The efficiency is achieved when the ternary matrix $M(m,n)$ of utility assessments is converted into a decimal vector, $V(m)$. The vector is then sorted and duplicates are eliminated first. Then, the redundant utility assessments are recursively eliminated using the special properties of the ternary and the decimal number systems, which will be introduced later in this Chapter, and the result is converted back into the ternary matrix format. Finally, the ternary matrix is checked for specific redundant assessments. As we will show, the conversion of the main part of the elimination process into one-dimensional space speeds up the algorithm tremendously. We will discuss the logic, implementation, and simulation results in the following sub-sections. Let us first review some necessary concepts.

7.1.1 Review of a conversion from the ternary number system to the decimal

Let us briefly review the conversion between the ternary and the decimal number systems. Recall that conversion of an d -digit ternary number into a decimal one can be accomplished by the following formula:

$$D = t_d R^d + t_{d-1} R^{d-1} + \ldots + t_1 R^1 + t_0 R^0,$$

where R is the base, or *radix*, which is equal to three, and t_d are the digits of the ternary number. As an example, let us convert the ternary number 120 into decimal D:

$$D = 1*3^2 + 2*3^1 + 0*3^0 = 15.$$

7.1.2 Review of a conversion from a decimal number to a ternary number

In order to convert a decimal number D to a ternary T, we continually divide D by three to have a quotient and a remainder of either one or zero until the final quotient equals zero. We store the remainders and then read them at the end in a reverse order. For example, the decimal number 15 is converted into ternary number 120 as shown in Table 7.1. The modulo division stops when the quotient is equal to zero. We then read the remainder in a reverse order so that the last and first remainders become the most and least significant trits of the ternary number, respectively.

Table 7.1: Converting decimal number 15 into ternary number 120.

Modulo division by 3	Quotient	Remainder
15 % 3	5	0 (least significant trit)
5 % 3	1	2
1 % 3	0	1 (most significant trit)

7.2 THE MAIN IDEA

The main idea of the Ternary-Decimal Exclusion algorithm can be best explained by a simple example and Table 7.2. Suppose we have a decision problem with only two attributes, i.e. $n = 2$.

Then the corresponding ternary matrix of utility assessments can contain only some of the $3^2 = 9$ ternary numbers from [0 0] to [2 2] as shown in the left column of Table 7.2. Observe now the following properties in a ternary-decimal number relation:

- If [2 2] exists ($max = 3^n - 1 = 8$ in decimal), then it eliminates (or covers) all other numbers; hence, all other numbers in the ternary matrix can be excluded.
- If [2 1] exists ($max - 1 = 7$ in decimal), then it eliminates all smaller numbers for which the remainder on modulo 3 division is equal to 1. For example, 7 covers 1, because 7 % 3 = 1.
- If [2 0] exists ($max - 2 = 6$ in decimal), then it excludes all smaller numbers for which the remainder on modulo 3 division is equal to 0. For instance, 6 eliminates 3, because 6 % 3 = 0.
- If [1 0] exists (in decimal: $\lfloor 0 + (max - 0)/3 \rfloor = 2$), then it covers all numbers below it.
- If [1 2] exists (in decimal: $\lfloor 0 + 2(max - 0)/3 \rfloor = 5$), then it eliminates all numbers below it up to $\lfloor 0 + (max - 0)/3 \rfloor = 2$, namely, 3, 4, and 5.
- Notice, 2 and 5 divide the range into three intervals of equal size, 1/3 of range each.

Table 7.2: Coverage table for a two-attribute decision problem.

Ternary number T when n = 2	**Decimal number D corresponding to T**	**Coverage: What can D eliminate?**
2 2	8	0 1 2 3 4 5 6 7 8
2 1	7	1 4 7
2 0	6	0 3 6
1 2	5	3 4 5
1 1	4	4
1 0	3	3
0 2	2	0 1 2
0 1	1	1
0 0	0	0

Hence, we can convert a ternary matrix into a decimal vector, sort the vector, and then eliminate duplicate utility assessments in a one-dimensional space. It is also very efficient to eliminate possible duplicate numbers from a sorted vector: we just traverse the list once in one direction and eliminate the duplicates in linear time. Table 7.2 shows the 'coverage table' for $n = 2$ attributes. The properties, which we introduced earlier, hold for any number of attributes, so that we can design a recursive algorithm for eliminating the redundant assessments. We first need to check (i) whether the top element of the sorted vector is 8, 7, or 6, and apply the corresponding elimination mechanisms, and then (ii) recursively divide the vector into three sub-vectors: Left, Mid, and Right, and repeat the elimination mechanism. To divide the sorted vector into these parts we can use the sub-vector bounds: $lb = \lfloor min + (max - min)/3 \rfloor$ and $mb = \lfloor min + 2(max - min)/3 \rfloor$ for the Left and Mid sub-vectors. The initial values of $min = 0$ and $max = 3^n - 1$ will be changed for each sub-interval during every interval division. We will explain these ideas in detail in the following sections.

7.3 PSEUDOCODES

7.3.1 High-level pseudocode of the Ternary-Decimal Exclusion algorithm

Table 7.3 presents the high-level pseudocode of the Ternary-Decimal Exclusion algorithm. It has the following five steps:

1. Convert the ternary matrix into a decimal vector.
2. Sort the vector and eliminate duplicates.
3. Eliminate from the vector decimal numbers corresponding to redundant utility assessments.
4. Convert the decimal vector into a ternary matrix.

5. Eliminate from the ternary matrix specific rows, if necessary. Return the matrix.

Consider a high-level pseudocode of the Ternary-Decimal Exclusion algorithm presented in Table 7.3. The pseudocode is self-explanatory; the only comment is about line 1. The two decimal integers min = 0, max = 3^n - 1, correspond to the theoretical lower and upper bounds of integer numbers in the decimal vector.

Table 7.3. High-level pseudocode of the Ternary-Decimal Exclusion algorithm

Algorithm Ternary-Decimal(***M***):	
Input:	An *m* x *n* ternary matrix ***M*** storing integers: 0, 1, 2.
Output:	The matrix ***M*** without duplicate and redundant rows.
1:	Set min = 0, max = 3^n - 1
2:	Convert rows of ***M***[*m*, *n*] into decimal integers and store in vector ***V***[*m*]: ***V*** = ternary2dec(***M***)
3:	Sort ***V***(*m*) in descending order and eliminate duplicate decimals: ***V*** = exclude_duplicates(***V***)
4:	Eliminate redundant decimals in ***V*** = exclude_redundant(***V***, min, max).
5:	Convert new decimal vector ***V*** to new ternary matrix ***M***: ***M*** = dec2ternary(***V***, *length(**V**)*, *n*)
6:	Eliminate specific redundancies from ***M***, if any: ***M*** = exclude_specific(***M***)
7:	**RETURN** ***M***

7.3.2 Step 1: Converting a ternary matrix to a decimal vector

The conversion of a ternary matrix into a decimal vector can be done by ternary2dec algorithm in $O(m \times n)$ in the worst case as shown in Table 7.4.

Table 7.4. High-level pseudocode of ternary2dec algorithm

Algorithm ternary2dec(***M***):	
Input:	An *m* x *n* ternary matrix ***M*** storing integers: 0, 1, 2.
Output:	A vector of decimal integers, ***V***[*m*].
1:	Initialize vector of zeros ***V***[*m*]; max = 3^n - 1
2:	**FOR** r = 1 : *m*
3:	d = 0
4:	**FOR** c = 1 : *n*
5:	d = d + (***M***[r, c])*3^(*n* - c)
6:	**IF** d = max **THEN RETURN** max, and terminate the program
7:	***V***[r] = d
8:	**RETURN** ***V***

Indeed, we should at least read the matrix in order to convert it into a decimal vector; hence, $O(m \times n)$. Notice, in line 6 we terminate the program if max = 3^n - 1 is found, i.e. vector of twos

exists in the ternary matrix. Recall, a vector of 2s implies that there is no utility independence condition. This is a hypothetical input, but the algorithm must be able to handle it.

7.3.3 Step 2: Sorting the decimal vector and eliminating duplicates

In step 2, one can utilize any existing algorithm for sorting the decimal vector $V(m)$ that runs in $O(m \log m)$ such as MergeSort or the randomized QuickSort. (see, for example, Cormen et al. 1997). In fact, there are built-in sort() functions in almost any integrated development environment (IDE). Having the vector sorted, the elimination of duplicates in $O(m)$ is straightforward. MATLAB has function unique(), which returns the sorted vector without duplicates.

For the sake of completeness, let me provide one possible implementation of the 2nd step of the Ternary-Decimal Exclusion algorithm, exclude_duplicates(V), based on a standard built-in sorting algorithm.

Table 7.5. High-level pseudocode of exclude_duplicates algorithm

```
Algorithm exclude_duplicates(V):
Input:    A vector of decimal integers, V[m].
Output:   A sorted vector of unique decimal integers, V.

1:    Initialize 1st and last indices:  fidx = 1,  lidx = m.
2:    Sort the vector: V = sort(V, fidx, lidx)
3:    i = 1
4:    WHILE (i < lidx)
5:            IF V[i] =  V[i+1]
6:                    V[i] = [ ]    % Eliminate V[i]
7:                    lidx =  lidx - 1
8:                    i =  i - 1
9:            i = i + 1
10:   RETURN V
```

7.3.4 Step 3: Recursive divide and conquer approach for excluding redundancies

Table 7.6 presents the high-level pseudocode of the exclude_redundant algorithm with binary search of right bounds of the Left and the Mid sub-vectors. This algorithm is a divide-and-conquer type of algorithms. It recursively divides a given vector into three sub-vectors based on certain properties of the decimal number system, eliminates the possible redundant numbers in each sub-vector, and merges them back into one vector. Notice that if (max-1) or (max-2) exists in the input vector, then exclude_redundant performs extra modulo 3 divisions on the Right sub-vectors. However, the size of Right sub-vectors decrease very fast with each recursive call.

The exclude_redundant algorithm utilizes the recursive search_bound algorithm. Table 7.7 presents the high-level pseudocode of the search_bound algorithm with binary search of bounds.

Table 7.6. High-level pseudocode of the exclude_redundant algorithm

```
Algorithm exclude_redundant(V, min, max):
Input: A vector of m sorted unique nonnegative decimal integers, V[m], and the range of possible values in V[m]: min and max of m.
Output: A new vector of sorted unique positive decimal integers, V, with length possibly less than m.
m = length(V)
IF m < 2 THEN RETURN V.
IF max(V) = max THEN RETURN V = max.
ELSEIF (V[m] = max - 1) AND (V[m - 1] = max - 2)
        WHILE counter < m - 1 DO eliminate V[counter] IF V[counter] modulo 3 equals to 1 OR 0
ELSEIF (V[m] = max - 1)
        WHILE counter < m DO eliminate V[counter] IF V[counter] modulo 3 equals to 1
ELSEIF (V[m] = max - 2)
        WHILE counter < m DO eliminate V[counter] IF V[counter] modulo 3 equals to 0
Set left bound lb = floor(min + (max - min)/3)
Set middle bound mb = floor(min + 2*(max - min)/3)
Set left, middle, and right vectors empty: L = M = R = [ ]
m = length(V)
idxL = search_bound(V, 1, m, lb)
IF idxL = 0 THEN L = [ ]
ELSEIF V[idxL] = lb THEN L = lb
ELSE L = V[1 : idxL]
idxM = search_bound(V, idxL + 1, m, mb)
IF idxM = idxL THEN M = [ ]
ELSEIF V[idxL] = mb THEN M = mb
ELSE M = V[idxL + 1 : idxM]
IF m = idxM THEN R = [ ]
ELSE R = V[idxM + 1 : m]
[L] = exclude_redundant(L, min, lb)
[M] = exclude_redundant(M, lb, mb);
[R] = exclude_redundant(R, mb, max);
[V] = [L M R];
RETURN V
```

Table 7.7. High-level pseudocode of the search_bound algorithm

```
Algorithm search_bound(V, left, right, x):
Input:    A sorted vector of unique nonnegative decimal integers, V, its first and end indices, and number x to search.
Output:   An index of x if it exists in V, index of element lower than x, otherwise.
1:    IF left > right THEN RETURN idx = right.
2:    ELSE
3:            mid =  floor(left + (right - left)/2)
4:            IF x < V[mid] THEN search_bound(V, left, mid - 1, x)
5:            ELSEIF x > V[mid] THEN search_bound(V, mid + 1, right, x)
6:            ELSE RETURN idx = mid.
```

7.3.5 Step 4: Converting the decimal vector to a ternary matrix

The conversion of the decimal vector back to the ternary matrix can be done with the dec2ternary algorithm. Table 7.8 illustrates the high-level pseudocode of the dec2ternary algorithm which runs in $\Theta(m \times n)$. We provide the pseudocode of this method for the sake of completeness; however, built-in functions there might be available in your IDE. MATLAB, for example, has the dec2base function that can be applied for this task.

Table 7.8. High-level pseudocode of the dec2ternary algorithm

```
Algorithm dec2ternary(V, length, n):
Input:    A sorted vector of unique positive decimal integers, V, its length, and number of attributes, n.
Output:   A ternary matrix M[length, n].
1:    Initialize matrix of zeros M[length, n]
2:    FOR rows = 1 : length
3:            D =  V[length - rows + 1]                  % decimal number
4:            FOR cols = n : -1 : 1
5:                    M[rows, cols]= D modulo 3  % remainder goes to M[rows, cols]
6:                    D = floor(D / 3)                   % D is now a quotient
7:    RETURN  M[rows, cols]
```

7.3.6 Step 5: Eliminating specific redundancies from the ternary matrix

The final step of the algorithm is to eliminate possible specific redundancies. As the number of attributes becomes large, due to the complexity of interdependence among the attributes, there might be some specific redundancies which must be eliminated. To eliminate such terms, we should utilize method exclude_specific; the pseudocode is presented in Table 7.9. This method

may look similar to the Twos-Complement approach; however, it is different. Unlike the Twos-Complement approach, this method does not compare each row with every other row. Since the input matrix is sorted, we partially compare its rows with only the following rows. Hence, the new approach is much faster than the Twos-Complement approach.

Table 7.9. High-level pseudocode of the algorithm exclude_specific

```
Algorithm exclude_specific(M):
Input:   An m x n ternary matrix M storing integers: 0, 1, 2.
Output:  The matrix M without duplicate and redundant rows.
1:       Set index r to be the 1st (unprocessed) row of matrix M.
2:       WHILE an unprocessed row of M exists DO compare all following rows of M with row r as follows:
3:                FORALL n: IF an element of row r is not equal to two THEN set its (column) index into vector twos_compl.
4:                Set index i to be r + 1, the uncompared row of matrix M.
5:                WHILE an uncompared row of M exists DO following:
6:                         IF any corresponding elements of rows r and i indexed in twos_compl are different THEN next i.
7:                         ELSEIF all of them are equal THEN delete row i since it is redundant.
8:
9:       RETURN M
```

7.3.7 An illustrative example

Consider an input ternary matrix as shown in Figure 7.1 (left) containing 20 rows and 4 columns. Let us simulate the outputs of each of the five steps of the Ternary-Decimal Exclusion algorithm. *After the 1st step*, we receive a vector of decimal integers. *After step #2*, we have a sorted vector of unique decimals.

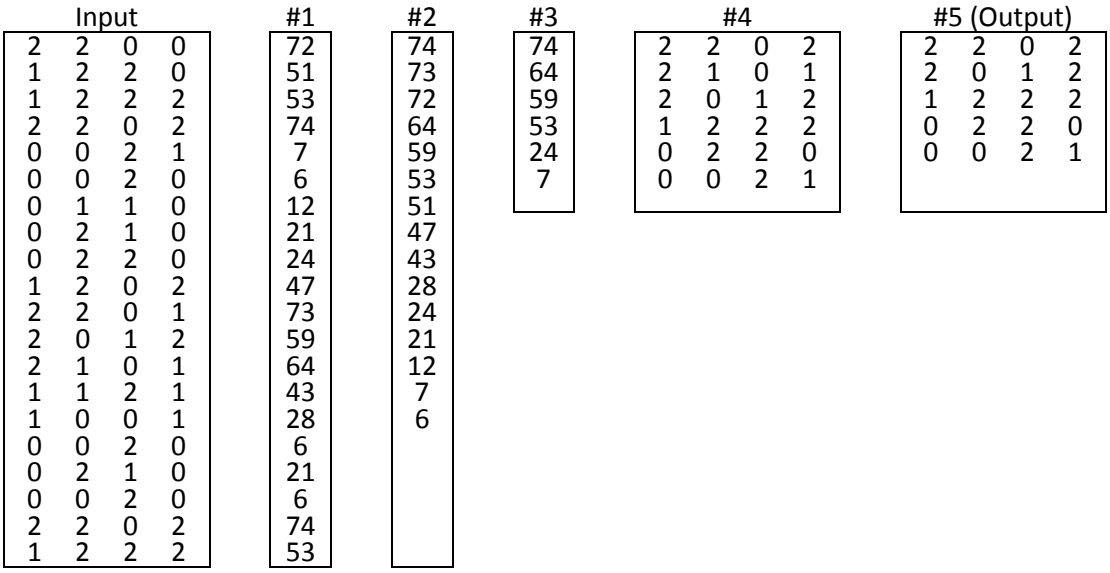


Figure 7.1: An input ternary matrix $M(20,4)$ and the outputs of each of the five steps of the Ternary-Decimal Exclusion algorithm.

In general, *the 3rd step* is to eliminate redundant assessments from this vector if any exist (see Figure 7.2).

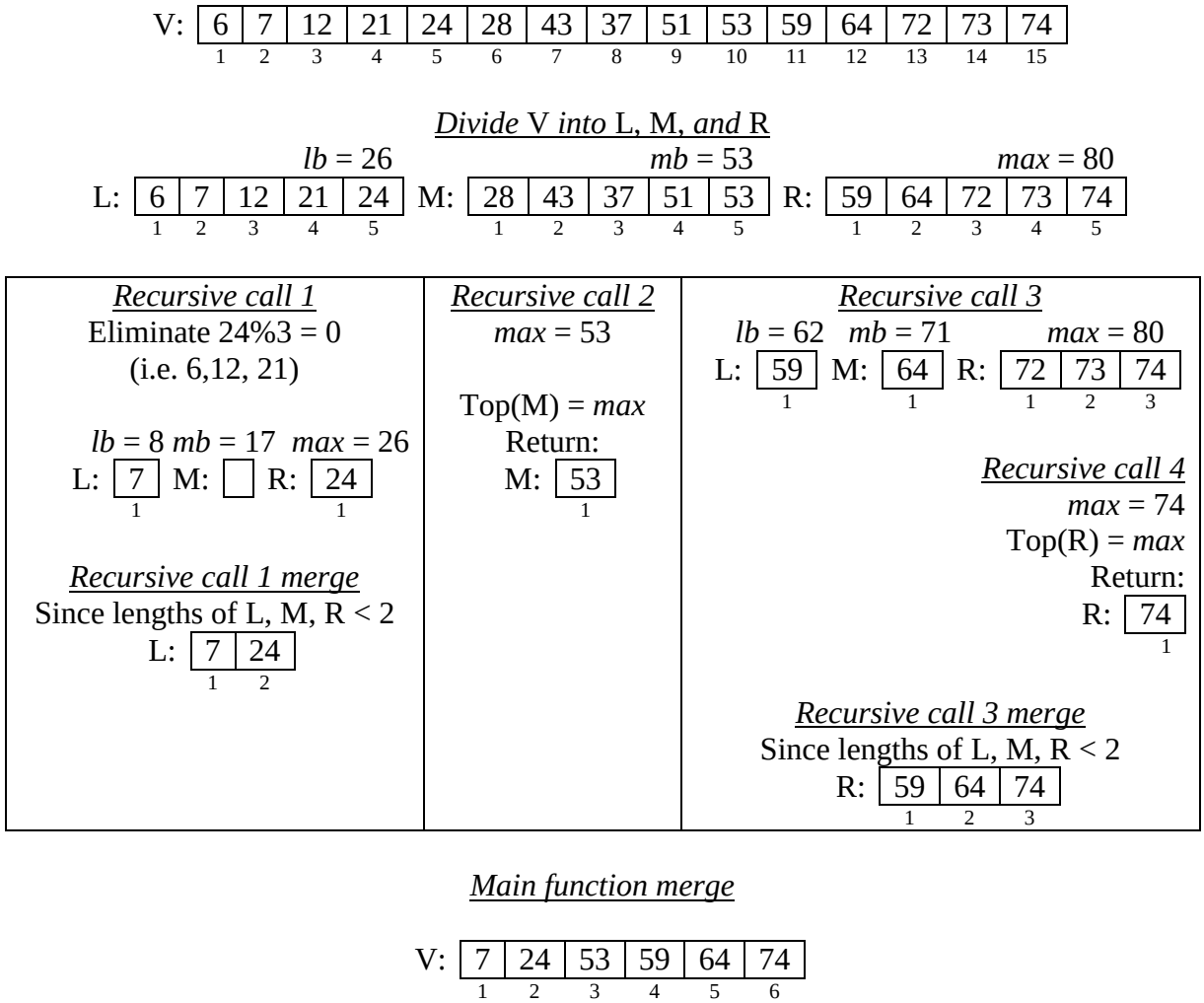


Figure 7.2: The decimal vector V after step 3 of the algorithm.

The 4th step of the algorithm converts the decimal vector V back to the ternary matrix M as shown in Figure 7.1(Step #4). *The 5th step* checks the consistency of the assessments in the ternary matrix from top to bottom. For example, the algorithm compares the twos-complement part of the top row with the 2nd row and determines that the latter is redundant, and must be eliminated. Then, it compares the 1st row with all other rows below it. Similarly, it compares the 2nd row with all rows below it. And so on. There are no any other redundant rows, so the algorithm returns the ternary matrix with 5 rows as shown in Figure 7.3 (Step #5).

7.4 SIMULATION RESULTS

Simulation results below demonstrate that the Ternary-Decimal Exclusion algorithm outperforms the Twos-Complement Exclusion algorithm. MATLAB implementation of the Twos-Complement Exclusion algorithm can be found in Appendix L.

Figure 7.3 shows average running times of the Twos-Complement and the Ternary-Decimal Exclusion algorithms for small numbers of functions for 10,000 runs. Notice, as the number of functions grows, the Ternary-Decimal performs better than the Twos-Complement. This becomes even more evident for larger numbers of functions as depicted in Figure 7.4.

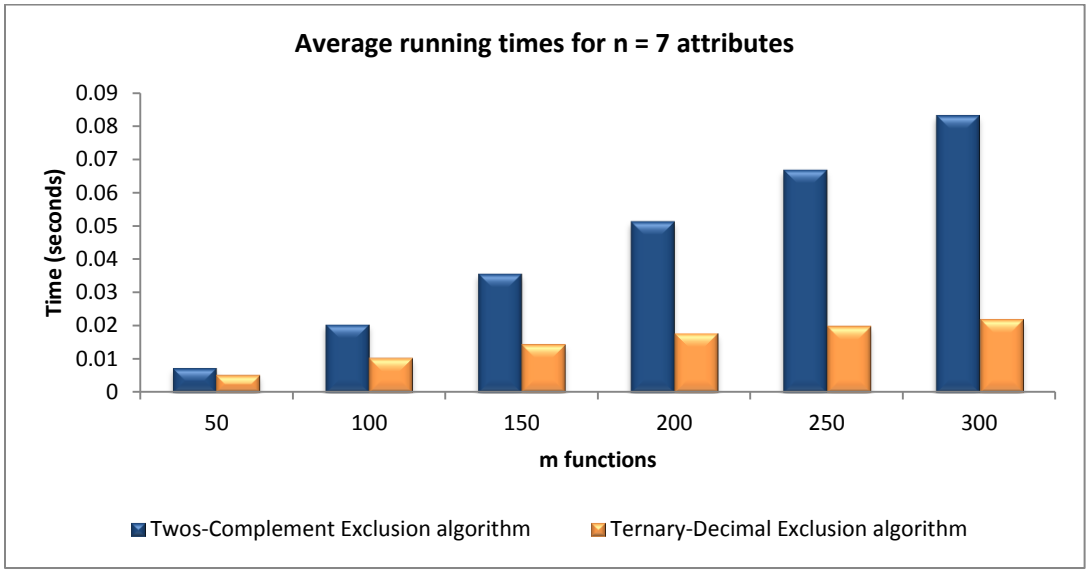


Figure 7.3. Average running times of the algorithms for small numbers of functions

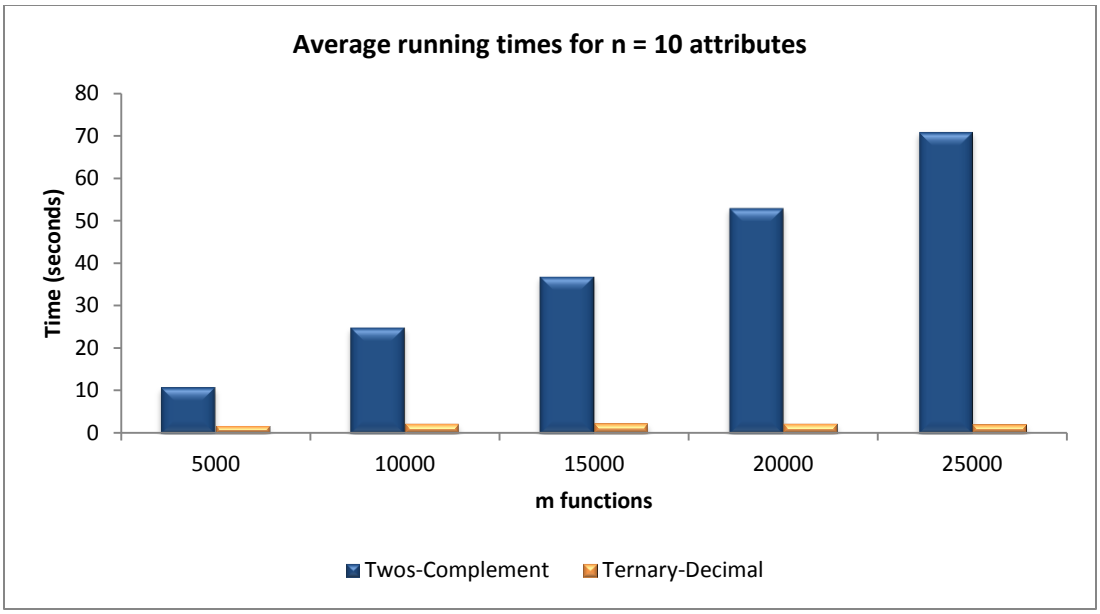


Figure 7.4. Average running times of the algorithms for large numbers of functions

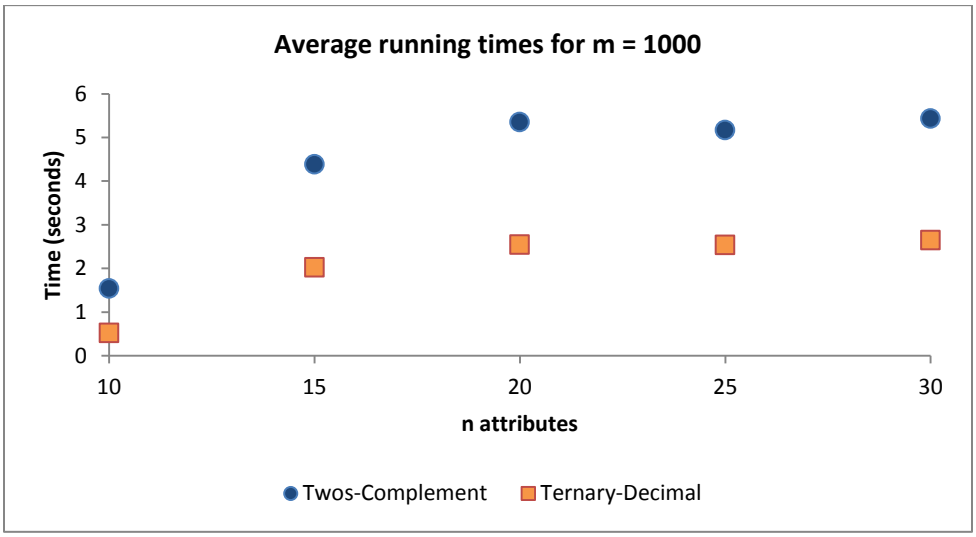


Figure 7.5. Average running times of the algorithms for different numbers of attributes

7.5 APPLICATION TO THE DEEP BOREHOLE FILTER RESTORATION PROBLEM

7.5.1 Three-attribute case: Utility Dependence Matrix #35 from Experiment 2

Consider the three-attribute utility dependence matrix #35 (see UDM #35 in APPENDIX E) assessed during Experiment 2 and also presented in Figure 7.6 for your convenience. The assessor is a nuclear engineer from Argonne National Laboratory and the attributes X_1, X_2, and X_3 were discussed in Section 4.2.2. What are the required assessments for constructing the multiattribute utility function based on this UDM? Expanding the utility function for this UDM by the Iterative Decomposition algorithm (see section 2.3.3) as shown in Figure 7.7 we have 13 utility assessments as illustrated in Figure 7.8 (left).

	X_1	X_2	X_3
X_1	○	○	
X_2	○	○	
X_3			○

Figure 7.6. Utility dependence matrix assessed from a nuclear engineer

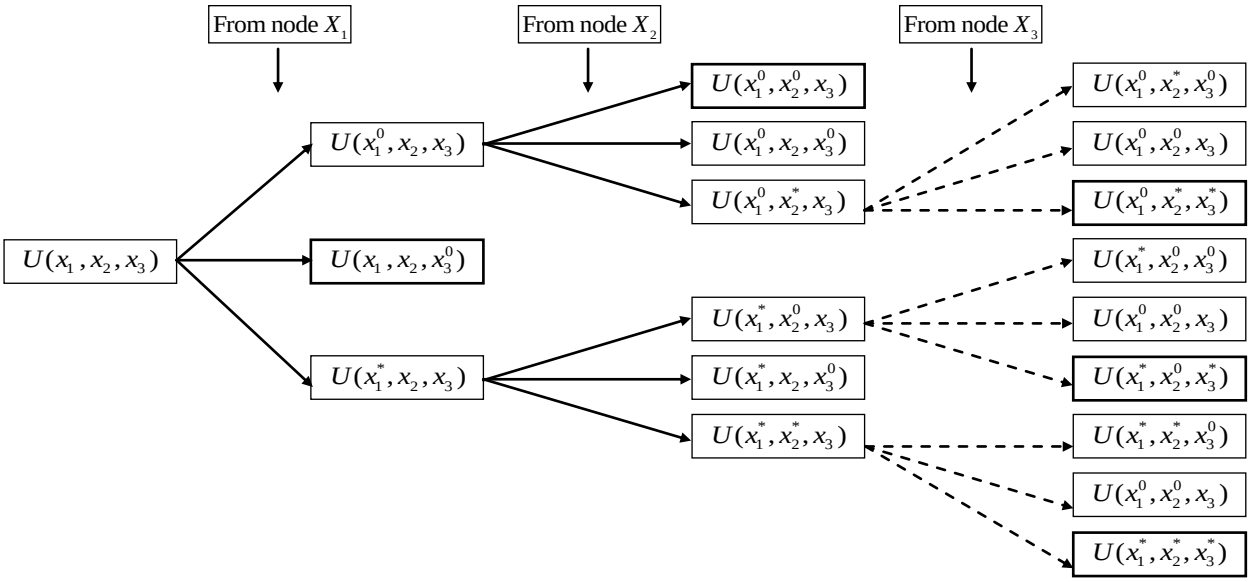


Figure 7.7: Tree representation and the iterative decomposition in the order: X_1,X_2,X_3.

$U(x_1, x_2, x_3^0)$,	2	2	0	24
$U(x_1^0, x_2^0, x_3)$,	0	0	2	2
$U(x_1^0, x_2, x_3^0)$,	0	2	0	6
$U(x_1^*, x_2, x_3^0)$,	1	2	0	15
$U(x_1^0, x_2^*, x_3^0)$,	0	1	0	3
$U(x_1^0, x_2^0, x_3)$,	0	0	2	2
$U(x_1^0, x_2^*, x_3^*)$,	0	1	1	4
$U(x_1^*, x_2^0, x_3^0)$,	1	0	0	9
$U(x_1^0, x_2^0, x_3)$,	0	0	2	2
$U(x_1^*, x_2^0, x_3^*)$,	1	0	1	10
$U(x_1^0, x_2^0, x_3)$,	1	1	0	12
$U(x_1^*, x_2^*, x_3^0)$,	0	0	2	2
$U(x_1^0, x_2^0, x_3)$,	1	1	1	13
$U(x_1^*, x_2^*, x_3^*)$.				

Figure 7.8: Utility assessments (left), their ternary matrix representation (middle), and the corresponding decimal vector representation (right).

We convert these utility assessments into a ternary matrix as shown in Figure 7.8 (middle), which is the input matrix to the Ternary-Decimal Exclusion algorithm. *The 1st step* of the algorithm is to convert this matrix to a decimal vector, V, as shown in Figure 7.8 (right). *The 2nd*

step is to sort this vector and eliminate duplicates; the result is shown in Figure 7.9. As we can see, only three decimals are eliminated. The resulting vector has 10 decimals.

~~2~~	~~2~~	~~2~~	2	3	4	6	9	10	12	13	15	24
			1	2	3	4	5	6	7	8	9	10

Figure 7.9: The decimal vector with eliminated duplicates.

The 3rd step eliminates redundant utility assessments from vector V as shown in Figure 7.10. The algorithm first finds that the top element of the vector is equal to $max - 2 = 24$, so it traverses all elements of the vector and eliminates elements for which the remainder of modulo division 24%3 is equal to zero. Then it recursively divides the vector into three parts: L, M, and R. Since there is no redundant elements in the vector, the algorithm returns the vector.

V:

2	3	4	6	9	10	12	13	15	24
1	2	3	4	5	6	7	8	9	10

Eliminate 24%3 = 0

V:

2	~~3~~	4	~~6~~	~~9~~	10	~~12~~	13	~~15~~	24
1	2	3	4	5	6	7	8	9	10

Divide V *into* L, M, *and* R

lb = 8 *mb* = 17 *max* = 26

L:

2	4
1	2

M:

10
1

R:

13	24
1	2

Recursive call 1

lb = 2 *mb* = 5 *max* = 8

L:

2
1

M:

4
1

R: [ ]

Since lengths of L, M, R < 2

Merge:

V:

2	4
1	2

Recursive call 2

lb = 11 *mb* = 14 *max* = 17

M:

10
1

Lengths M < 2

Return:

V:

10
1

Recursive call 3

lb = 20 *mb* = 23 *max* = 26

L:

13
1

M: [ ]

R:

24
1

Since lengths of L, M, R < 2

Merge:

V:

13	24
1	2

Main function merge

V:

2	4	10	13	24
1	2	3	4	5

Figure 7.10: The decimal vector V after Step 3 of the algorithm.

The 4th step of the algorithm converts the decimal vector V back to the ternary matrix M.

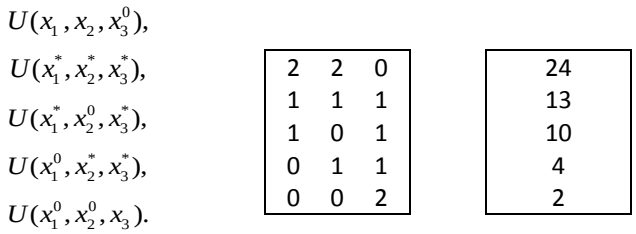


Figure 7.11: After 4th step: the decimal vector (right) is converted to the ternary matrix (middle), which represents the required utility assessments (left).

The 5th step checks the consistency of the assessments in the ternary matrix from top to bottom. In this example, the five assessments are final, so the algorithm returns the matrix as shown in Figure 7.11 (middle). The five required utility assessments are listed in Figure 7.11 (left).

7.5.2 Four-attribute case: Utility Dependence Matrix #4 from Experiment 1

Consider now the four-attribute utility dependence matrix #4 (see UDM #4 in APPENDIX C) assessed during Experiment 1 and also presented in Figure 7.12. The assessor is the Chief of Geological Surveys of a transnational uranium mining company. The attributes X_1, X_2, X_3 and X_4 were discussed in Section 1.3.3. Expanding the utility function for this UDM by the Iterative Decomposition algorithm we have 21 utility assessments as illustrated in Figure 7.13 (left). We are interested in the minimal (required) utility assessments for constructing the multiattribute utility function based on this UDM. We first represent these 21 utility assessments in the form of the ternary matrix as in Figure 7.13 (middle) and then convert them into the decimal vector as shown in Figure 7.13 (right).

	X_1	X_2	X_3	X_4
X_1	○	○	○	
X_2	○	○	○	
X_3	○		○	○
X_4	○			○

Figure 7.12: The utility dependence matrix #4 assessed in Experiment 2

$U(x_1^0, x_2^0, x_3, x_4^0)$,	0 0 2 0	6
$U(x_1^0, x_2^0, x_3^0, x_4)$,	0 0 0 2	2
$U(x_1^0, x_2^0, x_3, x_4^*)$,	0 0 2 1	7
$U(x_1^0, x_2^*, x_3^0, x_4^0)$,	0 1 0 0	9
$U(x_1^0, x_2^0, x_3^0, x_4)$,	0 0 0 2	2
$U(x_1^0, x_2^*, x_3^0, x_4^*)$,	0 1 0 1	10
$U(x_1^0, x_2^*, x_3^*, x_4^0)$,	0 1 1 0	12
$U(x_1^0, x_2^0, x_3^0, x_4)$,	0 0 0 2	2
$U(x_1^0, x_2^*, x_3^*, x_4^*)$,	0 1 1 1	13
$U(x_1, x_2, x_3^0, x_4^0)$,	2 2 0 0	72
$U(x_1, x_2^0, x_3, x_4^0)$,	2 0 2 0	60
$U(x_1, x_2, x_3^*, x_4^0)$,	2 2 1 0	75
$U(x_1^*, x_2^0, x_3, x_4^0)$,	1 0 2 0	33
$U(x_1^*, x_2^0, x_3^0, x_4)$,	1 0 0 2	29
$U(x_1^*, x_2^0, x_3, x_4^*)$,	1 0 2 1	34
$U(x_1^*, x_2^*, x_3^0, x_4^0)$,	1 1 0 0	36
$U(x_1^*, x_2^0, x_3^0, x_4)$,	1 0 0 2	29
$U(x_1^*, x_2^*, x_3^0, x_4^*)$,	1 1 0 1	37
$U(x_1^*, x_2^*, x_3^*, x_4^0)$,	1 1 1 0	39
$U(x_1^*, x_2^0, x_3^0, x_4)$,	1 0 0 2	29
$U(x_1^*, x_2^*, x_3^*, x_4^*)$.	1 1 1 1	40

Figure 7.13: Utility assessments (left), their ternary matrix representation (middle), and the corresponding decimal vector representation (right).

We then sort the decimal vector and eliminate duplicate assessments as shown in Figure 7.14.

~~2~~	~~2~~	2	6	7	9	10	12	13	~~29~~	~~29~~	29	33	34	36	37	39	40	60	72	75
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21

Figure 7.14: The decimal vector with eliminated duplicates.

The 3rd step of the algorithm sets the minimal and maximal possible elements of the vector as: $min = 0$ and $max = 80$, and then it recursively divides the vector into three sub-vectors L, M, and R. Figure 7.15 illustrates six recursive calls on the sub-vector L, Figure 7.16 on sub-vector M, and Figure 7.17 on sub-vector R.

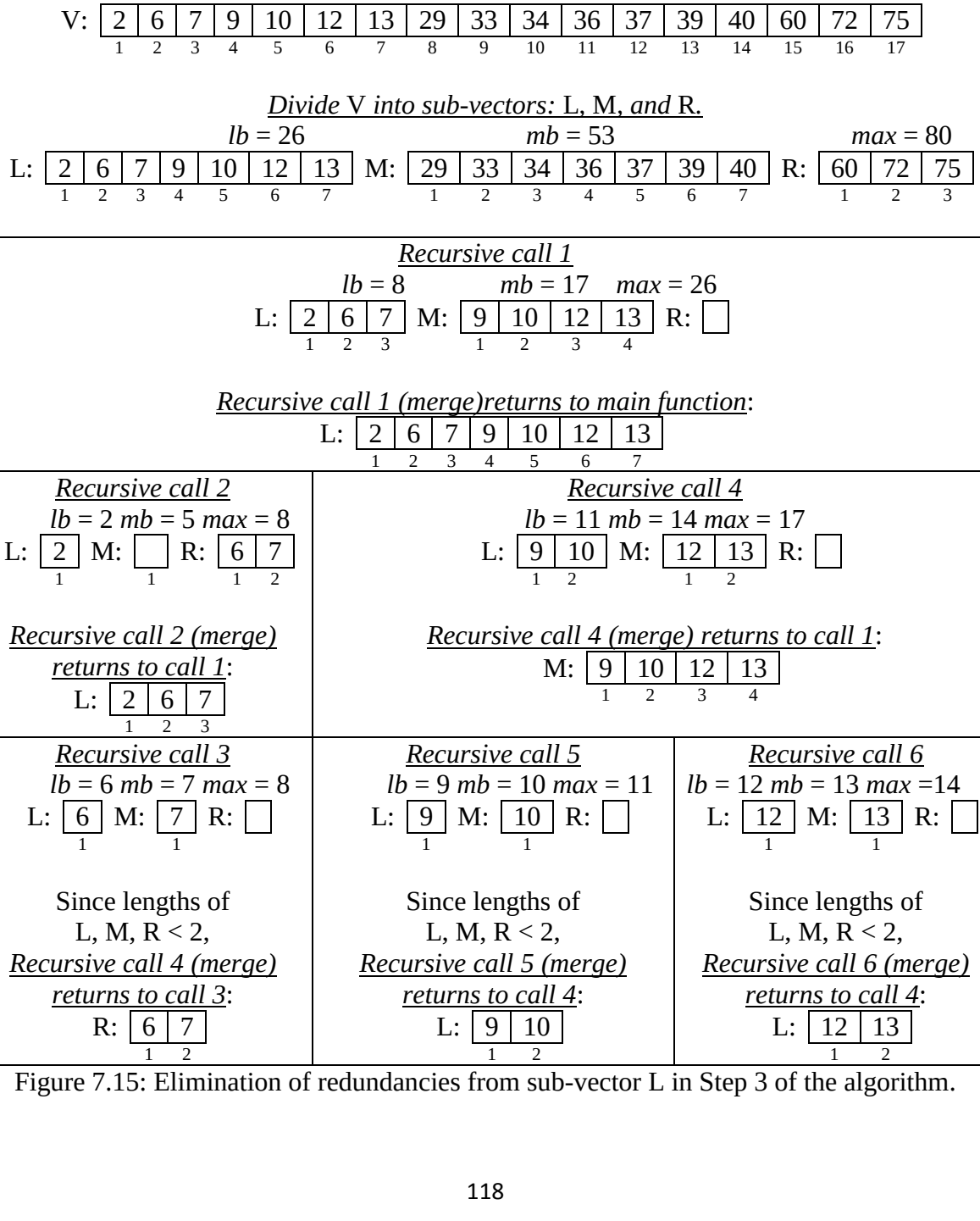


Figure 7.15: Elimination of redundancies from sub-vector L in Step 3 of the algorithm.

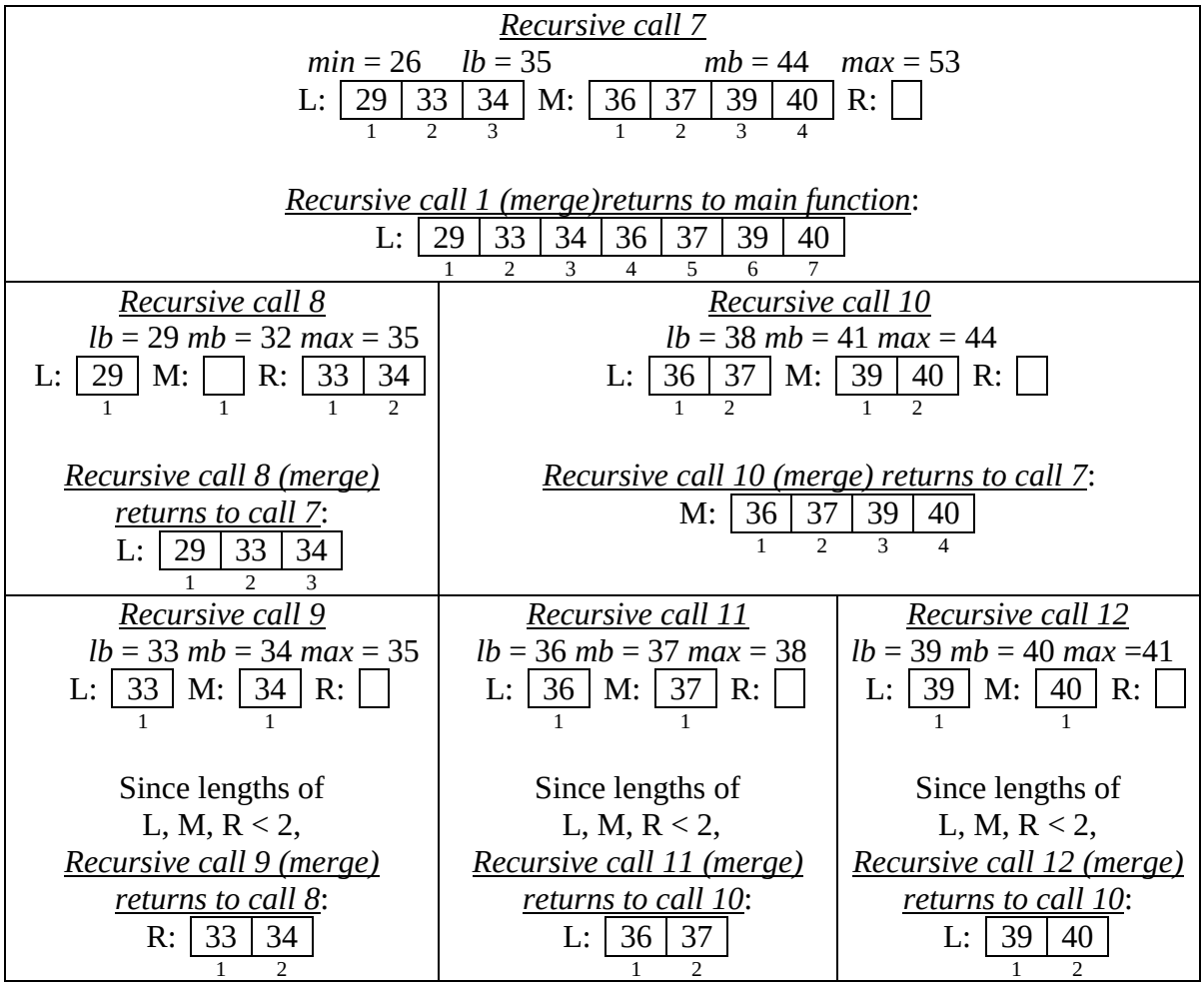


Figure 7.16: Elimination of redundancies from sub-vector M in Step 3 of the algorithm.

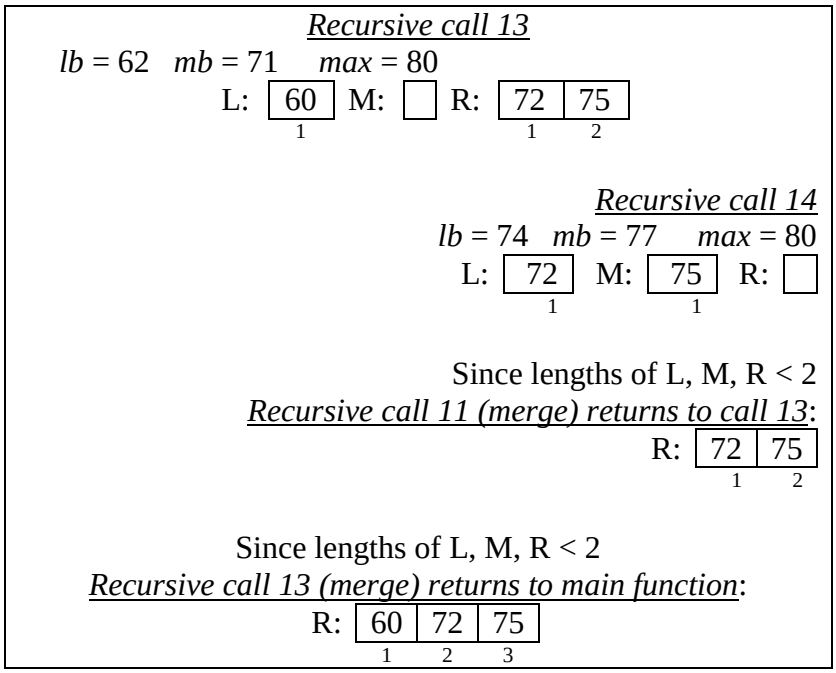


Figure 7.17: Elimination of redundancies from sub-vector R in Step 3 of the algorithm.

2	2	1	0
2	2	0	0
2	0	2	0
1	1	1	1
1	1	0	1
1	0	2	1
1	0	0	2
0	1	1	1
0	1	0	1
0	0	2	1
0	0	0	2

2	2	1	0
2	2	0	0
2	0	2	0
1	1	1	1
1	1	~~1~~	~~0~~
1	1	0	1
1	1	~~0~~	~~0~~
1	0	2	1
1	~~0~~	2	~~0~~
1	0	0	2
0	1	1	1
0	1	1	0
0	1	0	1
0	1	~~0~~	~~0~~
0	0	2	1
0	~~0~~	2	~~0~~
0	0	0	2

75
72
60
40
39
37
36
34
33
29
13
12
10
9
7
6
2

Figure 7.18: Vector V after Step 3 (right), its corresponding ternary matrix after Step 4 (middle), and the output ternary matrix (left) after Step 5.

Figure 7.18 illustrates the utility assessments after Step 3 (right), Step 4 (middle), and Step 5 (left) of the algorithm. Step 5 eliminates six redundancies as shown in Figure 7.18 (middle). The output ternary matrix contains 11 utility assessments as demonstrated in Figure 7.18 (left); 10

terms out of 21 were eliminated. As we can see, the algorithm significantly simplifies the elimination of redundant assessments in a decision problem with even four attributes. This will save more time for analyst as the number of attributes of the decision problem increases.

7.5.3 Five-attribute case: randomly generated Utility Dependence Matrix

We will now illustrate the five-attribute case. Consider the utility dependence matrix presented in Figure 7.19. This UDM is randomly generated for the sake of the example, but it could be assessed from the decision maker in practice. The four attributes could be those depicted in Table 1.4, which we discussed in section 1.3.4. The fifth attribute could be, for example, *the social impact* measured in the number of new jobs (e.g. track drivers) for a local population.

	X_1	X_2	X_3	X_4	X_5
X_1	○		○		
X_2	○	○			
X_3	○	○	○		
X_4	○	○	○	○	
X_5					○

Figure 7.19: Randomly generated UDM for the borehole recovery problem with five attributes

So, let us run the Iterative Decomposition algorithm and find the number of utility assessments after maximal decomposition, i.e. the total number of utility assessments. We then run the Ternary-Decimal Exclusion algorithm to find the minimum number of utility assessments required for construction of the multiattribute utility function. We are interested in the number of

utility assessments eliminated from the total set of the assessments as well as their number after each step of the Ternary-Decimal Exclusion algorithm.

So, the simulations show that there are 59 utility assessments in total for the UDM in Figure 7.19. After excluding duplicate utility assessments in Step 2 of the Ternary-Decimal Exclusion algorithm, we have 44 assessments. After excluding redundancies in Step 3, we have 29 assessments. Step 5 did not find any specific redundancy; the algorithm outputs 29 utility assessments. As we can see, the algorithm excluded 30 out of 59 utility assessments, which is more than 50%.

7.6 APPLICATION TO THE 10-ATTRIBUTE NUCLEAR POWER SITING PROBLEM

In this section, we illustrate the Ternary-Decimal Exclusion algorithm on a larger problem than our deep borehole filter restoration problem. For this, we consider the 10-attribute nuclear power siting problem. The siting of nuclear power facilities is a very complex process (Gros, 1975; Gros et al., 1976; Solomon and Haynes, 1984). The decision maker must consider many factors such as profit, safety, and national interest. For example, Keeney and Sicherman (1983) compare technology options for generating electricity utilizing 14 attributes. Keeney and Raiffa (1976) discuss the objectives and the 10 attributes for siting nuclear power facilities (see Table 7.10). Without discussing the objectives and attributes for this decision problem in detail, we will randomly generate a 10-attribute UDM and identify how our algorithm eliminates redundant utility assessments.

Table 7.10: Attributes for the siting nuclear power facilities (adapted from Keeney and Raiffa, 1976).

Attribute	Objective
X_1	Minimize pollution
X_2	Provide aesthetically pleasing facilities
X_3	Minimize human health hazards
X_4	Provide necessary power
X_5	Minimize consumer power costs
X_6	Maximize economic benefits to local community
X_7	Maximize utility company profits
X_8	Maximize state revenues
X_9	Improve balance of payments
X_{10}	Reduce dependency on foreign fuels

	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
X_1	○							○	○	○
X_2	○	○	○					○	○	
X_3			○	○	○	○	○		○	
X_4	○	○	○	○	○	○	○	○	○	
X_5	○				○	○		○		
X_6		○				○				○
X_7		○	○	○		○	○	○	○	
X_8	○	○		○		○		○		
X_9	○	○			○	○	○		○	
X_{10}	○	○		○	○					○

Figure 7.20: Randomly generated 10-attribute UDM for the nuclear power siting problem

Figure 7.20 illustrates the 10-attribute UDM for the nuclear power siting problem. The total number of utility assessments for this problem is equal to 1755 and it is almost impossible to check the consistency of the assessments manually. Using the Ternary-Decimal Exclusion

algorithm, one can find that the minimal set of utility assessments required for the construction of the multiattribute utility functions must contain 773 terms.

7.7 SUMMARY AND FUTURE WORK

This Chapter introduced an efficient algorithm for eliminating redundant terms in a multiattribute utility function. The redundancy of the terms implies redundant utility assessments or redundant sets of questions for the decision maker; hence, must be eliminated. We demonstrated that the Ternary-Decimal Exclusion algorithm outperforms the Twos-Complement Exclusion algorithm. However, I believe that there is room for future work in this direction. In particular, one can analyze the time and space complexity of the Ternary-Decimal Exclusion algorithm. One can extend the analysis as shown in Table 7.2 and find better recurrence relations, which can eliminate step 5 of the Ternary-Decimal Exclusion algorithm discussed in section 7.3.

CHAPTER 8: CONCLUSION

In this Dissertation, we demonstrated the importance of partial utility independence of attributes to the complexity and quality of decisions. We then explored theoretical aspects of functional forms of the multiattribute utility functions, which incorporate partial UI conditions. We also developed tests for assessing the UDMs with PUI. In order to verify the applicability of canonical forms in practice, two experiments were conducted among the 117 experts in the uranium extraction industry using these novel tests. The uniqueness of the test is that it (i) is the first test developed for assessing partial UI conditions including boundary independence and one-switch independence, (ii) is formal, but elicits the partial UI conditions indirectly, (iii) does not contain any special decision analysis terms and is not difficult for the assessors, and (iv) is easily scalable for any number of attributes. This is the first practical study on the verification of the applicability of canonical forms of a multiattribute decision problem.

We then compared the decisions based on profit analysis, utility independence assumption, and partial utility independence conditions, the true preferences of the decision maker. For this, we defined and structured the deep borehole filter restoration problem in terms of objectives, attributes, alternatives, uncertainties, influence diagrams, decision trees, and multiattribute utility functions, we then evaluated the alternatives of the deep borehole filter restoration problem on the four most important attributes from the ultimate decision maker's point of view. It was found, that when only profit is considered, or utility independence is assumed, then the decisions vary 40% of the time compared to the true preferences of the decision maker. This shows that even though the assessment and construction of the MUF with true preferences is more difficult

compared to other approaches, the results are worth it, because otherwise, our decisions may not be appropriate and we may place people at unnecessary risk.

This dissertation determines the number of canonical forms for an *n*-attribute decision problem graphically represented by the utility dependence matrix (UDM) introduced in (Abbas, 2010a). The dissertation also determines the convergence of the ratio of canonical UDMs to the total number of UDMs as the number of attributes $n \to \infty$. In particular, the absolute number of canonical forms increases with n, indicating the increasing variety of the forms available for the decision maker, while their relative number decreases, demonstrating the increasing complexity of the decision problem with n. The results imply that the decision maker should aggregate the attributes (whenever possible) to decrease the size of utility assessments needed for construction of the multiattribute utility function. These results should help the decision maker better understand the complexity of their decision problem and make a sound decision.

The elimination of utility assessments which are duplicate or redundant for the construction of the MUFs can be an uphill struggle in complex decision problems. This research contributes to this problem by presenting the Twos-Complement Exclusion algorithm and the Ternary-Decimal Exclusion algorithm. First, the dissertation proposes the ternary matrix representation of utility assessments that substantially simplifies their processing and determination. Then, given a set of utility assessments in the form of the ternary matrix, the algorithm excludes subsets of duplicate and redundant assessments and returns a subset that is indispensable for the multiattribute utility function. The Twos-Complement and Ternary-Decimal Exclusion algorithms are also helpful for checking the consistency of the required utility assessments. The Twos-Complement Exclusion algorithm was compared with a naïve (brute-force) approach both analytically and experimentally. The results indicate that the Twos-Complement Exclusion

algorithm is more effective and efficient compared to the naïve approach especially for complex multiattribute problems. Then the Ternary-Decimal Exclusion algorithm was compared with the Twos-Complement Exclusion algorithm and it was demonstrated that it outperforms the Twos-Complement Exclusion algorithm.

The results of this dissertation were applied to an important and complex decision problem in uranium mining, the deep borehole filter restoration problem. The decision analysis framework presented in this dissertation should help decision makers in addressing the worldwide 14% shortage of uranium for nuclear power generation and help 1.3 billion people who live today without access to electricity.

APPENDIX A (TEST 1): Experimental assessments of utility dependence matrices for the deep borehole filter restoration problem in uranium mining.

The efficiency of uranium extraction depends on many factors; the most important ones are presented in Table A.1. If the concentration of the extracting uranium is decreasing, then it can be increased by either carrying out repair and restoration work (chemical, physical, and pulsing methods) or redrilling the well. The ultimate goal is to find the most optimal way of boosting the concentration of uranium from the borehole. The goal of this questionnaire is to determine the importance of the dependence of the attributes for the decision making.

Table A.1: Attributes for the borehole restoration problem, their measures and ranges.

Attribute	Measure	Range	
		best	worst
X - Uranium output from a borehole	U kg/hour	$x^* = 3.2$	$x^0 = 0.016$
Y - Restoration time (idle time)	hour	$y^* = 6$	$y^0 = 200$
Z - Additional expenses (for repair or redrilling)	$	$z^* = 73$	$z^0 = 23.3\text{K}$
W - Health effects of the process (job hazard)	1 – 5 scale	$w^* = 1$	$w^0 = 5$

	X	Y	Z	W
X	◯			
Y		◯		
Z			◯	
W				◯

Figure A.1: The utility dependence matrix.

Analyst: *In the following questions, if your answer for attributes, say* X *and* Y *(as in question 1) is "Yes", then, please, put a node in the cell* XY *of the matrix in Figure A.1. If your answer is "No", then move to the next question.*

1. Analyst: *Will your preferences (decisions) for uncertain situations on the level of the uranium output,* X, *and the number of hours spent on the process,* Y, *depend on particular level (value) of* Y *given that the other two attributes, namely,* Z *and* W, *are held fixed?*

In other words, if your answer is "No", then in the conditional utility function U(X|Y,Z,W), the value of Y does not matter and can be replaced with its minimal (worst) value for the simplification of a subsequent determination of multiattribute utility function, i.e., U(X|Y,Z,W)= U(X|Y^0,Z,W).
Expert: *Yes/No.*

2. Analyst: *Will your preferences (decisions) for uncertain situations on the level of the uranium output,* X, *and the amount of money spent on the process,* Z, *depend on particular level (value) of* Z *given that the other two attributes, namely,* Y *and* W, *are held fixed?*
Expert: *Yes/No*

3. Analyst: *Will your preferences (decisions) for uncertain situations on the level of the uranium output,* X, *and the level of health effects of the process,* W, *depend on particular level (value) of* W *given that the other two attributes, namely,* Y *and* Z, *are held fixed?*
Expert: *Yes/No*

4. Analyst: *Will your preferences (decisions) for uncertain situations on the number of hours spent on the process, Y, and the level of the uranium output, X, depend on particular level (value) of X given that the other two attributes, namely, Z and W, are held fixed?*
Expert: *Yes/No.*

5. Analyst: *Will your preferences (decisions) for uncertain situations on the number of hours spent on the process, Y, and the amount of money spent on the process, Z, depend on particular level (value) of Z given that the other two attributes, namely, X and W, are held fixed?*
Expert: *Yes/No*

6. Analyst: *Will your preferences (decisions) for uncertain situations on the number of hours spent on the process, Y, and the level of health effects of the process, W, depend on particular level (value) of W given that the other two attributes, namely, X and Z, are held fixed?*
Expert: *Yes/No*

7. Analyst: *Will your preferences (decisions) for uncertain situations on the amount of money spent on the process, Z, and the level of the uranium output, X, depend on particular level (value) of X given that the other two attributes, namely, Y and W, are held fixed?*
Expert: *Yes/No.*

8. Analyst: *Will your preferences (decisions) for uncertain situations on the amount of money spent on the process, Z, and the number of hours spent on the process, Y, depend on particular level (value) of Y given that the other two attributes, namely, X and W, are held fixed?*
Expert: *Yes/No*

9. Analyst: *Will your preferences (decisions) for uncertain situations on the amount of money spent on the process, Z, and the level of health effects of the process, W, depend on particular level (value) of W given that the other two attributes, namely, X and Z, are held fixed?*
Expert: *Yes/No*

10. Analyst: *Will your preferences (decisions) for uncertain situations on the level of health effects of the process, W, and the level of the uranium output, X, depend on particular level (value) of X given that the other two attributes, namely, Y and W, are held fixed?*
Expert: *Yes/No.*

11. Analyst: *Will your preferences (decisions) for uncertain situations on the level of health effects of the process, W, and the number of hours spent on the process, Y, depend on particular level (value) of Y given that the other two attributes, namely, X and W, are held fixed?*
Expert: *Yes/No*

12. Analyst: *Will your preferences (decisions) for uncertain situations on the level of health effects of the process, W, and the amount of money spent on the process, Z, depend on particular level (value) of Z given that the other two attributes, namely, X and Y, are held fixed?*
Expert: *Yes/No*

13. Analyst: *Was the test difficult for you (circle one number)?*

	Not at all				Very
Expert:	*1*	*2*	*3*	*4*	*5*

Expert: _____ Signature: _____

Position: _____

APPENDIX B (TEST 2): An experimental assessment of interdependence of attributes

The efficiency of uranium extraction depends on many factors; the most important ones are presented in Table B.1, where the ranges of attributes can vary within their most preferable (best) and least preferable (worst) values. For example, attribute *Health effects of the process* (Z) can gradually/evenly vary as follows:

- $z^* = 0,$ best value, no adverse health effects to the personnel,
- $z = 1,$ very cold/hot weather in the field,
- $z = 2,$ light intoxication,
- $z = 3,$ light intoxication + cold/hot,
- $z^0 = 4,$ worst value, moderate intoxication.

Table B.1: Attributes for the filter restoration problem, their measurement units and ranges.

Attribute	Measurement Unit	Range	
		best	worst
X - Uranium output from a well	U kg/hour	$x^* = 3.2$	$x^0 = 0$
Y - Additional expenses (for repair or redrilling)	\$	$y^* = 73$	$y^0 = 23.3\text{K}$
Z - Health effects of the process (job hazard)	0 – 4 scale	$z^* = 0$	$z^o = 4$

The goal of this experiment is to determine the interdependence of these attributes in decisions related to *a filter restoration problem*. When a filter of a uranium extraction well is clogged, it can be cleaned by different methods (chemical, physical, airflow, etc.) or the well can be redrilled at another place. So, values of the attributes vary depending on the chosen *process*, and the time of the process may vary from 6 hours to 200 hours. The knowledge of interdependence of the attributes will help others make an informed decision. Please answer the questions on the following pages as carefully and thoughtfully as possible keeping in mind the ranges of values of attributes in Table B.1.

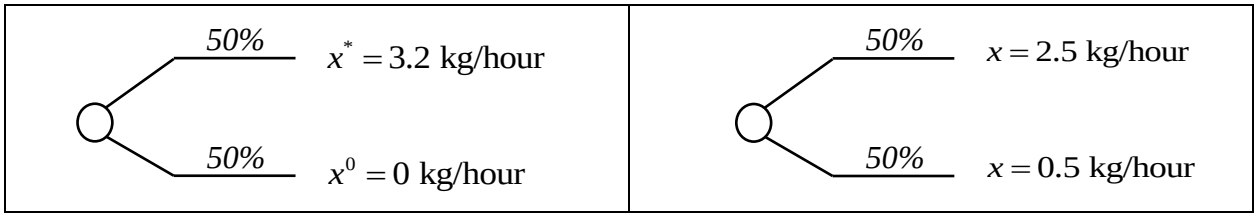
Your cooperation is highly appreciated and may influence future decisions and policies in uranium extraction industry. The data will be used for scientific research purposes only.

Question 1. Dependence: uranium output (X) of additional expenses (Y)

a) *Given two projects A and B with 50-50 chances of receiving different <u>uranium output from a well</u>, which project would you prefer in <u>Scenario 1</u>?*

<u>Scenario 1</u>: in both projects $y^* = \mathbf{73}$ dollars of additional expenses

<u>Project A</u> <u>Project B</u>



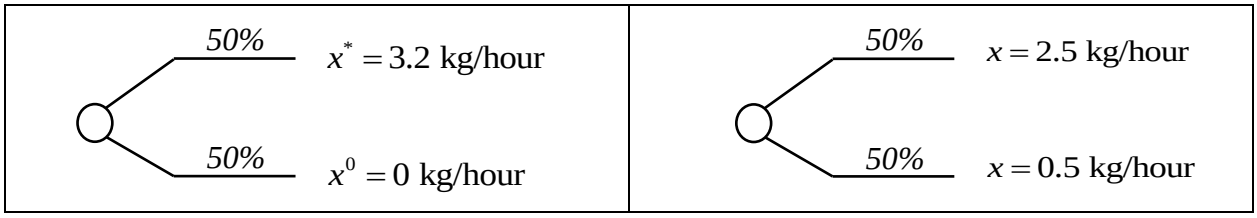
Circle one
out of three: *Project A* *I'm indifferent (A and B are equivalent)* *Project B*

---

b) *Which project would you prefer in <u>Scenario 2</u>?*

<u>Scenario 2</u>: in both projects $y^0 = \mathbf{23.3}$ Thousand dollars of additional expenses

<u>Project A</u> <u>Project B</u>



Circle one
out of three: *Project A* *I'm indifferent (A and B are equivalent)* *Project B*

---

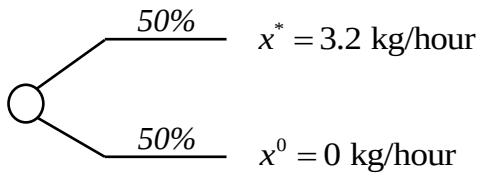
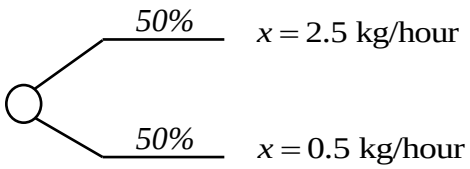
c) *If the value of attribute Y in Scenario 1 (or in Scenario 2) were held fixed at some other fixed value between y^* and y^0, would your answer change?*

Circle your answer: *Yes* *No*

Question 2. Dependence: uranium output (X) of health effects (Z)

a) *Given two projects A and B with 50-50 chances of receiving different <u>uranium output from a well</u>, which project would you prefer in <u>Scenario 1</u>?*

<u>Scenario 1</u>: in both projects $z^* = \mathbf{0}$, no adverse health effects

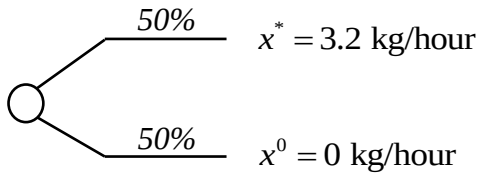
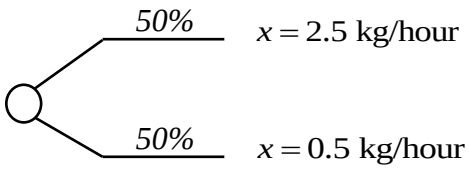
Project A	Project B
	

Circle one
out of three: *Project A* *I'm indifferent (A and B are equivalent)* *Project B*

---

b) *Which project would you prefer in <u>Scenario 2</u>?*

<u>Scenario 2</u>: in both projects $z^0 = \mathbf{4}$, moderate intoxication

Project A	Project B
	

Circle one
out of three: *Project A* *I'm indifferent (A and B are equivalent)* *Project B*

---

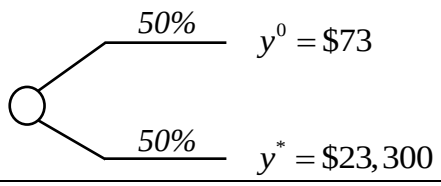
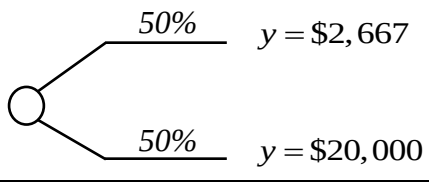
c) *If the value of attribute Z in Scenario 1 (or in Scenario 2) were held fixed at some other fixed value between z^* and z^0, would your answer change?*

Circle your answer: *Yes* *No*

Question 3. Dependence: additional expenses (Y) of uranium output (X)

a) *Given two projects A and B with 50-50 chances of having different <u>additional expenses</u>, which project would you prefer in <u>Scenario 1</u>?*

<u>Scenario 1</u>: in both projects $x^* = \mathbf{3.2}$ kg/hour of uranium output

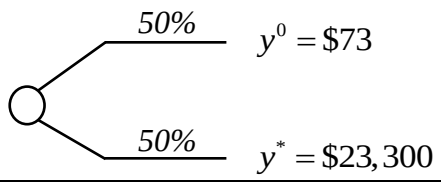
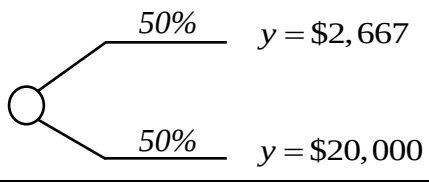
<u>Project A</u>	<u>Project B</u>
	

Circle one
out of three: *Project A* *I'm indifferent (A and B are equivalent)* *Project B*

---

b) *Which project would you prefer in <u>Scenario 2</u>?*

<u>Scenario 2</u>: in both projects $x^0 = \mathbf{0}$ kg/hour of uranium output

<u>Project A</u>	<u>Project B</u>
	

Circle one
out of three: *Project A* *I'm indifferent (A and B are equivalent)* *Project B*

---

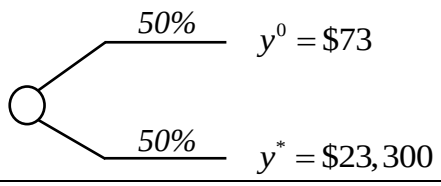
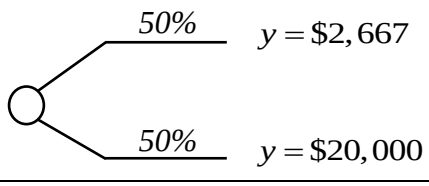
c) *If the value of attribute X in Scenario 1 (or in Scenario 2) were held fixed at some other fixed value between x^* and x^0, would your answer change?*

Circle your answer: *Yes* *No*

Question 4. Dependence: additional expenses (Y) of health effects (Z)

a) *Given two projects A and B with 50-50 chances of having different <u>additional expenses</u>, which project would you prefer in <u>Scenario 1</u>?*

<u>Scenario 1</u>: in both projects $z^* = \mathbf{0}$, no adverse health effects

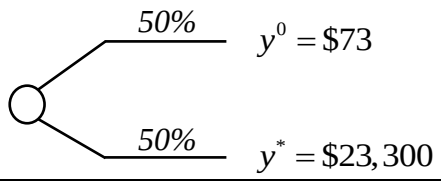
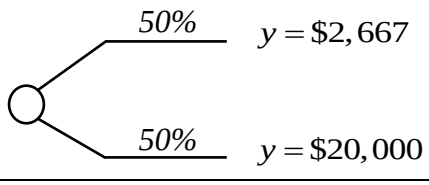
<u>Project A</u>	<u>Project B</u>
	

Circle one
out of three: *Project A* *I'm indifferent (A and B are equivalent)* *Project B*

---

b) *Which project would you prefer in <u>Scenario 2</u>?*

<u>Scenario 2</u>: in both projects $z^0 = \mathbf{4}$, moderate intoxication

<u>Project A</u>	<u>Project B</u>
	

Circle one
out of three: *Project A* *I'm indifferent (A and B are equivalent)* *Project B*

---

c) *If the value of attribute Z in Scenario 1 (or in Scenario 2) were held fixed at some other fixed value between z^* and z^0, would your answer change?*

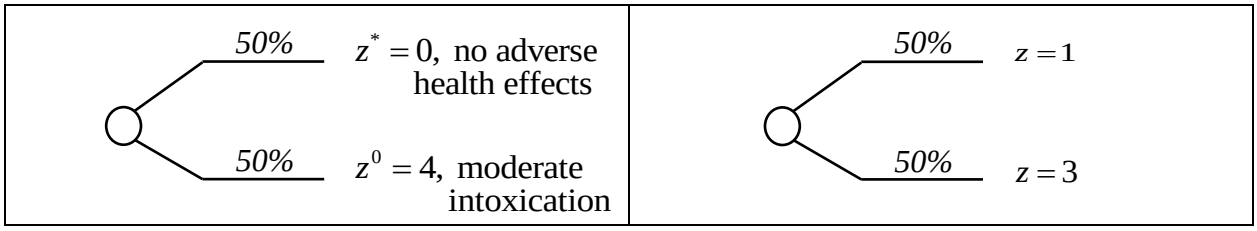
Circle your answer: *Yes* *No*

Question 5. Dependence: health effects (Z) of uranium output (X)

a) *Given two projects A and B with 50-50 chances of receiving different <u>health effects</u>, which project would you prefer in <u>Scenario 1</u>?*

<u>Scenario 1</u>: in both projects $x^* = \mathbf{3.2}$ kg/hour of uranium output

<u>Project A</u> <u>Project B</u>



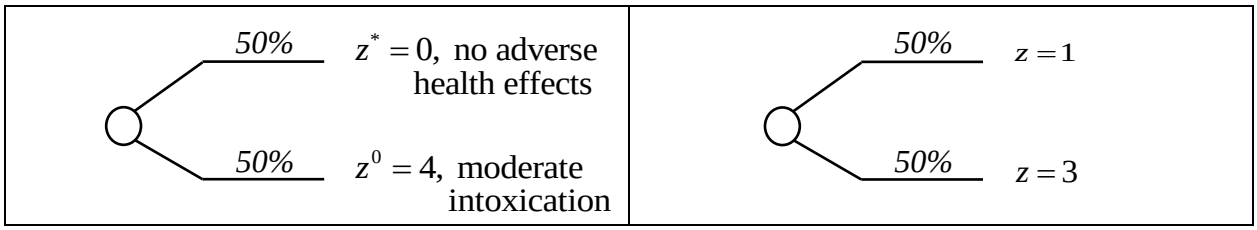
Circle one out of three: *Project A* *I'm indifferent (A and B are equivalent)* *Project B*

---

b) *Which project would you prefer in <u>Scenario 2</u>?*

<u>Scenario 2</u>: in both projects $x^0 = \mathbf{0}$ kg/hour of uranium output

<u>Project A</u> <u>Project B</u>



Circle one out of three: *Project A* *I'm indifferent (A and B are equivalent)* *Project B*

---

c) *If the value of attribute X in Scenario 1 (or in Scenario 2) were held fixed at some other fixed value between x^* and x^0, would your answer change?*

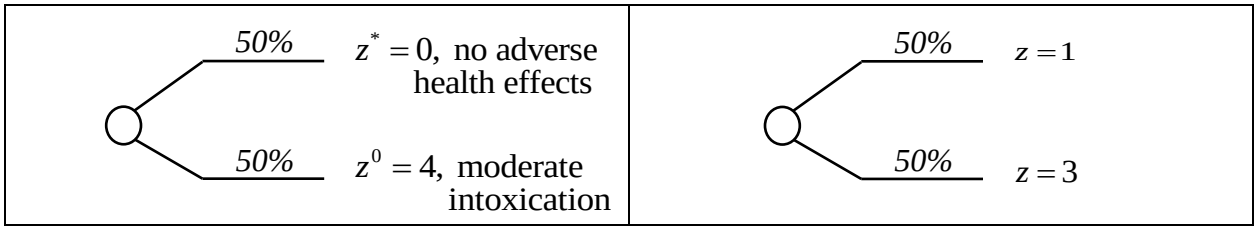
Circle your answer: *Yes* *No*

Question 6. Dependence: health effects (Z) of additional expenses (Y)

a) *Given two projects A and B with 50-50 chances of receiving different <u>health effects</u>, which project would you prefer in <u>Scenario 1</u>?*

<u>Scenario 1</u>: in both projects $y^* = \mathbf{73}$ dollars of additional expenses

<u>Project A</u> <u>Project B</u>



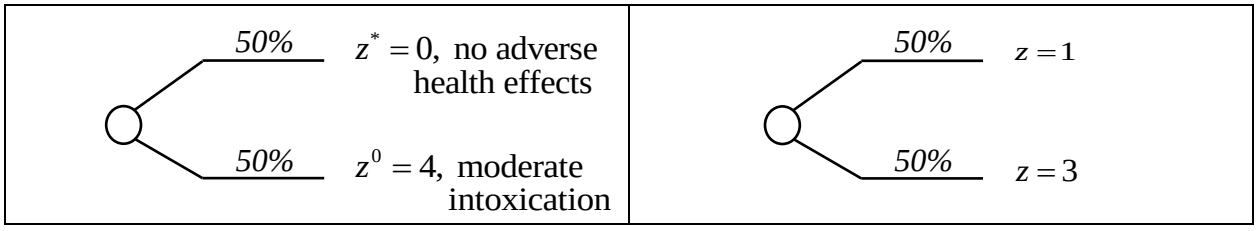
Circle one out of three: *Project A* *I'm indifferent (A and B are equivalent)* *Project B*

---

b) *Which project would you prefer in <u>Scenario 2</u>?*

<u>Scenario 2</u>: in both projects $y^0 = \mathbf{23.3}$ Thousand dollars of additional expenses

<u>Project A</u> <u>Project B</u>



Circle one out of three: *Project A* *I'm indifferent (A and B are equivalent)* *Project B*

---

c) *If the value of attribute Y in Scenario 1 (or in Scenario 2) were held fixed at some other fixed value between y^* and y^0, would your answer change?*

Circle your answer: *Yes* *No*

Was the test difficult for you?

Not at all Very

Circle one number: *1 2 3 4 5*

Your name: _____ Signature: _____

Organization: _____

Position: _____

Your e-mail: _____

APPENDIX C (RESULTS OF TEST 1): 105 4x4 utility dependence matrices assessed for the deep borehole filter restoration problem during Experiment 1.

A1 = [1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1];	A2 = [1 1 1 1 0 1 1 1 1 1 1 0 1 1 1 1];	A3 = [1 1 1 1 1 1 1 0 0 1 1 0 1 1 0 1];	A4 = [1 1 1 0 1 1 1 0 1 0 1 1 1 0 0 1];
A5 = [1 1 0 0 1 1 1 0 1 1 1 0 1 0 0 1];	A6 = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1];	A7 = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1];	A8 = [1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1];
A9 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];	A10 = [1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1];	A11 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];	A12 = [1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1];
A13 = [1 1 0 1 1 1 1 1 1 0 1 1 0 1 0 1];	A14 = [1 0 1 1 1 1 1 0 1 1 1 1 0 0 0 1];	A15 = [1 1 0 1 1 1 1 0 0 1 1 1 1 0 0 1];	A16 = [1 1 1 1 1 1 1 1 0 0 1 1 1 1 1 1];
A17 = [1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1];	A18 = [1 1 1 1 1 1 1 0 1 1 1 1 0 1 1 1];	A19 = [1 1 1 0 1 1 1 1 0 1 1 0 1 0 1 1];	A20 = [1 0 1 1 1 1 1 1 1 0 1 0 0 1 1 1];
A21 = [1 1 1 0 1 1 1 1 1 0 1 1 1 1 0 1];	A22 = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];	A23 = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1];	A24 = [1 0 1 1 1 1 1 1 1 1 1 1 1 0 0 1];
A25 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];	A26 = [1 0 1 1 1 1 1 1 0 0 1 0 0 1 1 1];	A27 = [1 1 1 1 1 1 1 1 0 1 1 1 0 1 1 1];	A28 = [1 1 1 1 1 1 1 0 0 0 1 0 1 1 0 1];
A29 = [1 1 0 1 1 1 1 1 1 0 1 1 0 1 0 1];	A30 = [1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1];	A31 = [1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1];	A32 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];

A33 = [1 0 1 0 1 1 0 1 1 1 1 1 1 1 0 1];	A34 = [1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1];	A35 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];	A36 = [1 1 1 1 1 1 1 1 1 0 1 0 1 0 0 1];
A37 = [1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1];	A38 = [1 1 1 1 1 1 1 0 1 1 1 0 1 0 0 1];	A39 = [1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1];	A40 = [1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1];
A41 = [1 1 1 1 1 1 1 1 1 0 1 0 1 0 0 1];	A42 = [1 1 1 1 1 1 1 1 0 0 1 0 1 0 0 1];	A43 = [1 1 1 1 1 1 1 1 0 0 1 0 1 0 0 1];	A44 = [1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 1];
A45 = [1 1 1 1 1 1 1 0 1 1 1 0 1 1 0 1];	A46 = [1 1 1 1 0 1 0 0 0 1 1 1 1 1 0 1];	A47 = [1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 1];	A48 = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
A49 = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];	A50 = [1 1 1 1 1 1 1 1 1 0 1 1 0 1 1 1];	A51 = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];	A52 = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1];
A53 = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1];	A54 = [1 1 1 1 0 1 1 0 1 1 1 1 0 1 1 1];	A55 = [1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1];	A56 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];
A57 = [1 1 0 1 0 1 1 1 0 0 1 0 0 1 0 1];	A58 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];	A59 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];	A60 = [1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1];
A61 = [1 1 1 0 1 1 1 0 1 0 1 0 1 0 0 1];	A62 = [1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1];	A63 = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1];	A64 = [1 1 1 0 1 1 1 0 1 0 1 0 1 0 1 1];

A65 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];	A66 = [1 1 1 1 1 1 0 1 1 0 1 1 1 0 0 1];	A67 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];	A68 = [1 1 0 1 1 1 0 1 0 1 1 0 0 0 0 1];
A69 = [1 1 0 1 1 1 1 0 1 1 1 0 1 1 0 1];	A70 = [1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1];	A71 = [1 0 0 1 1 1 0 1 1 1 1 1 0 0 0 1];	A72 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];
A73 = [1 1 1 0 1 1 1 0 1 1 1 0 1 0 0 1];	A74 = [1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1];	A75 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];	A76 = [1 1 1 1 1 1 0 1 1 1 1 1 1 0 0 1];
A77 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];	A78 = [1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1];	A79 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];	A80 = [1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1];
A81 = [1 1 1 1 1 1 1 1 1 1 1 0 1 1 0 1];	A82 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];	A83 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1];	A84 = [1 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1];
A85 = [1 1 1 1 1 1 1 1 1 0 1 1 0 0 0 1];	A86 = [1 1 1 1 1 1 1 1 0 0 1 1 0 0 0 1];	A87 = [1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1];	A88 = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1];
A89 = [1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1];	A90 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1];	A91 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1];	A92 = [1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 1];
A93 = [1 1 1 1 1 1 1 1 1 0 1 1 1 1 0 1];	A94 = [1 1 1 1 1 1 1 1 1 0 1 1 1 0 0 1];	A95 = [1 1 1 1 1 1 1 1 0 1 1 0 1 0 0 1];	A96 = [1 0 1 1 0 1 1 0 1 1 1 1 0 1 0 1];

A97 = [1 1 1 1 1 1 1 1 1 0 1 0 1 1 1 1];	A98 = [1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1];	A99 = [1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1];	A100 = [1 1 1 1 1 1 1 1 0 1 1 1 1 0 1 1];
A101 = [1 1 1 1 1 1 1 1 0 1 1 1 0 0 0 1];	A102 = [1 1 1 1 1 1 0 0 1 0 1 1 1 0 1 1];	A103 = [1 1 1 0 0 1 1 0 1 1 1 1 0 1 0 1];	A104 = [1 1 1 1 1 1 1 1 0 1 1 1 1 0 0 1];
A105 = [1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1];			

APPENDIX D (RESULTS OF TEST 2): Actual responses of 40 experts in Experiment 2

Answers to the questions:

Experts	1			2			3			4			5			6			Complexity of the test
	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	a	b	c	
1	I	A	N	B	B	Y	I	A	Y	A	B	Y	I	A	N	A	B	N	5
2	A	I	Y	A	A	Y	I	A	N	I	I	Y	B	B	Y	A	I	N	3
3	A	I	Y	A	A	Y	I	B	N	A	A	Y	I	A	Y	A	I	N	4
4	B	I	Y	I	A	Y	A	B	N	A	A	N	I	A	N	I	A	N	4
5	A	A	N	B	A	Y	B	I	N	A	A	N	I	A	Y	A	I	N	5
6	I	A	N	A	I	N	A	I	N	B	I	N	A	I	N	A	I	N	3
7	A	A	Y	I	A	N	I	I	Y	A	I	Y	B	I	Y	I	A	Y	3
8	A	B	Y	B	B	N	A	I	Y	B	I		I	A	Y	I	I	N	
9	B	A	Y	A	A	N	B	I	Y	A	A	Y	B	A	Y	A	B	N	5
10	A	I	Y	A	A	N	A	B	Y	A	B	N	I	I	N	A	B	Y	5
11	A	I	N	A	I	Y	B	A	N	B	I	Y	A	A	N	A	I	N	4
12	I	A	N	B	I	N	A	I	Y	B	I	N	I	A	N	A	I	Y	3
13	A	I	N	A	I	N	B	I	Y	A	I	Y	B	I	N	I	B	Y	3
14	A	I	N	B	I	Y	A	I	N	B	B	Y	A	I	N	A	I	Y	4
15	A	I	N	B	A	N	A	I	N	B	A	Y	B	A	Y	B	A	N	4
16	B	B	N	B	I	N	I	I	Y	I	I	Y	B	B	Y	B	B	Y	2
17	B	A	Y	B	B	Y	A	B	N	A	A	Y	B	B	Y	A	B	N	3
18	A	A	Y	A	A	Y	B	I		B	B		A	I		A	A		3
19	A	A	Y	A	A	Y	A	B	Y	B	B	Y	A	A	Y	A	A	Y	4
20	B	B	N	B	B	N	A	A	N	B	B	N	A	A	N	A	A	Y	3
21	A	A	N	A	A	N	B	B	N	B	B	N	B	B	N	B	B	N	2
22	B	I	Y	A	B	N	A	A	Y	I	B	Y	B	I	Y	B	B	N	2
23	I	I	Y	A	A	N	B	B	N	I	B	Y	B	B	N	A	B	Y	2
24	B	B	Y	B	B	Y	B	A	Y	A	A	Y	B	B	Y	B	B	Y	3
25	B	B	Y	B	B	Y	B	A	Y	B	B	Y	A	A	N	B	A	Y	2
26	A	A	N	A	B	Y	B	B	N	B	B	N	B	B	N	B	B	N	3
27	A	A	N	A	A	N	B	B	N	B	B	N	B	B	N	B	B	N	2
28	A	B	N	A	B	Y	B	B	N	B	B	N	B	B	N	B	B	N	2
29	I	B	N	B	A	Y	A	A	N	B	A	Y	A	A	N	A	B	N	2
30	B	B	N	B	A	Y	B	A	Y	A	A	N	A	B	Y	B	A	N	2
31	B	B	Y	B	B	Y	B	A	Y	A	A	Y	B	B	Y	B	B	Y	2
32	B	B	Y	B	B	Y	B	A	Y	A	A	Y	B	B	Y	B	B	Y	3
33	I	B	N	B	A	Y	A	A	N	B	A	Y	A	A	N	A	B	N	2
34	B	B	Y	B	B	Y	B	A	Y	A	A	Y	B	B	Y	B	B	Y	2
35	A	B	Y	B	B	N	A	B	Y	B	B	N	B	B	N	B	B	N	2
36	A	A	N	A	A	N	A	A	N	I	I	N	I	I	N	I	I	N	3
37	A	B	Y	A	B	Y	A	B	Y	I	I	N	B	I	Y	B	B	N	2
38	B	I	Y	B	I	Y	B	I	Y	B	I	Y	B	I	Y	B	B	N	3
39	B	A	Y	B	A	Y	A	I	Y	A	I	Y	I	I	N	I	I	N	3
40	B	B	Y	B	B	Y	B	A	Y	A	A	Y	B	B	Y	B	B	Y	2
																		Average:	2.97

APPENDIX E (RESULTS OF EXPERIMENT 2): 40 UDMs converted to a binary form

1

o	o	o
o	o	o
o	o	o

2

o	o	o
o	o	o
o	o	o

3

o	o	o
o	o	o
o	o	o

4

o	o	o
o	o	
o	o	o

5

o		o
o	o	
o	o	o

6

o	o	o
o	o	o
o	o	o

7

o	o	o
o	o	o
o	o	o

8

o	o	
o	o	o
o		o

9

o	o	
o	o	o
o	o	o

10

o	o	
o	o	o
	o	o

11

o	o	o
o	o	o
	o	o

12

o	o	o
o	o	o
o	o	o

13

o	o	o
o	o	o
o	o	o

14

o	o	o
o	o	o
o	o	o

15

o	o	o
o	o	o
o	o	o

16

o		o
	o	o
o	o	o

17

o	o	o
o	o	o
o	o	o

18

o	o	o
o	o	o
o	o	o

19

o	o	o
o	o	o
o	o	o

20

o		
	o	
		o

21

o		
	o	
		o

22

o	o	o
o	o	o
o		o

23

o		
	o	o
	o	o

24

o	o	o
o	o	o
o	o	o

25

o	o	
o	o	o
	o	o

26

o		o
	o	
		o

27

o		
	o	
		o

28

o	o	o
	o	
		o

29

o	o	o
	o	o
	o	o

30

o		o
o	o	
o	o	o

31

o	o	o
o	o	o
o	o	o

32

o	o	o
o	o	o
o	o	o

33

o	o	o
	o	o
	o	o

34

o	o	o
o	o	o
o	o	o

35

o	o	
o	o	
		o

36

o		
	o	
		o

37

o	o	o
o	o	o
o		o

38

o	o	o
o	o	o
		o

39

o		
	o	
		o

40

o	o	o
o	o	
o		o

APPENDIX F: DETERMINING THE NUMBER AND CONVERGENCE OF SPECIAL FUNCTIONAL FORMS FOR PARTIAL UTILITY INDEPENDENCE

F1: New definitions

We now determine the number and convergence of the canonical forms and introduce a new utility condition that will be helpful for a decomposition of MUFs with PUI conditions. Appendix F presents some work (e.g., Theorems F.1 and F.2) published in (Abdildin and Abbas, 2013).

Definition F.1: *If every attribute,* X_i, $i = 1,...,n$, *is utility independent of at least one attribute in its complement, then the decision problem asserts proper partial utility independence*.

Definition F.2: *A utility dependence matrix of a decision problem with proper partial utility independence is defined as a proper partial utility dependence matrix.*

Graphically, the utility dependence matrix (UDM) is a proper partial UDM if all of its rows have at least one vacant cell.

Definition F.3: *The assessment of the instantiation (value) of a multiattribute utility function, in which every attribute is in its boundary (minimal or maximal) instantiations, is defined as the utility value assessment.*

The decomposition of the proper partial UDM leads to *utility value assessments*, the smallest possible assessments.

F2: Determining the number of the canonical forms

The purpose of this subsection is to determine the number of canonical representations of UDMs. This may help the DM to better understand the complexity of a decision problem and

appropriately formulate it for further assessments. We are interested in the number of available canonical UDMs for an *n*-attribute decision problem and its ratio to the total number of UDMs.

Theorem F.1. *The total number of possible canonical utility dependence matrices for an n-attribute decision problem is equal to* $\sum_{r=0}^{n} C_r^n 2^{(n-r)r} - n$.

Proof. When the UDM is the unit matrix (the matrix without non-vacant cells), i.e. $r = 0,$ then UDM is in canonical form by definition. When $r = 1,$ any of n rows of the UDM should have at least one empty cell in $(n-1)$ cells of the row, which results in $(2^{n-1}-1)n$ canonical UDMs. If $r \geq 2,$ then UDM is canonical when all symmetric pairs in incomplete rows are vacant, but their other $(n-r)r$ cells can be either full or vacant; hence, there are $2^{(n-r)r}$ possible canonical UDMs in each of the $C_r^n = \dfrac{n!}{r!(n-r)!}$ rows. So, the total number of canonical forms of UDM for an *n*-attribute decision problem is equal to $c = 1 + (2^{n-1}-1)n + \sum_{r=2}^{n} \dfrac{n!2^{(n-r)r}}{r!(n-r)!}.$

Expanding and rearranging terms, we have

$$c = 1 + n2^{n-1} - n + \sum_{r=2}^{n} \frac{n!2^{(n-r)r}}{r!(n-r)!} = \frac{n!}{0!(n-0)!} 2^{(n-0)0} + \frac{n!}{1!(n-1)!} 2^{(n-1)1} - n + \sum_{r=2}^{n} \frac{n!2^{(n-r)r}}{r!(n-r)!}$$

$$= \sum_{r=0}^{n} \frac{n!2^{(n-r)r}}{r!(n-r)!} - n = \sum_{r=0}^{n} C_r^n 2^{(n-r)r} - n.$$

Q.E.D.

Theorem F.1 tells us that the absolute number of canonical UDMs grows with the number of attributes. Increasing variety of canonical forms will help the DM to choose the appropriate one for their decision problem.

Using Theorem F.1, we can find the total number of possible canonical UDMs. For example, for three attributes, there are 23 different canonical UDMs as shown in Figure F.1. In the upper lane utility dependence matrices (UDMs from 1 to 13) the number of incomplete rows of the UDM, $r \geq 2$, but all symmetric pairs (colored and marked by $\times$) are vacant; hence, these UDMs are canonical. In the lower lane, for all UDMs the condition of $r < 2$ holds; hence, they are also canonical by definition. Notice, the UDMs number 1 and 23 are special cases, because the former is an identity matrix, and the latter is a unit matrix. Hence, canonical forms also include utility dependence and utility independence conditions as special cases, which is a very good property.

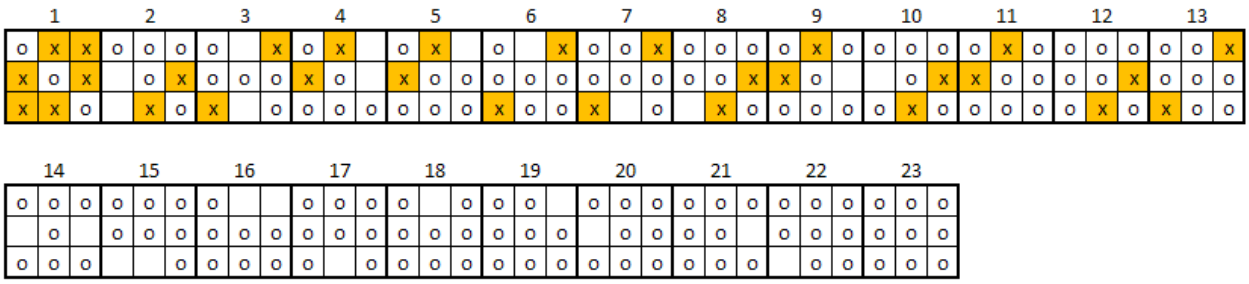


Figure F.1: Various canonical UDMs for a three-attribute decision problem.

F3: Determining the convergence of the canonical forms

Theorem F.1 tells us the total number of canonical UDMs for a given number of attributes, but it is interesting to know how the relative number of canonical forms alters with changes in the number of attributes of a decision problem. The following Theorem determines the convergence of the ratio of canonical UDMs to the total number of UDMs as the number of attributes of a decision problem changes.

Theorem F.2. *Let* c_n *be the number of UDMs in the canonical form for an n-attribute decision problem, and* t_n *be the total number of UDMs. Then,* $\lim_{n\to\infty} c_n / t_n = 0.$

Proof. The total number of possible utility dependence matrices for a decision problem with $n \geq 2$ attributes is equal to $2^{n(n-1)}$. To see this, recall that a UDM is a binary matrix with ones in the main diagonal. Therefore, only $n(n-1)$ cells of the UDM (except the cells in the main diagonal) can be either vacant or non-vacant; hence, the total number of UDMs for an $n \times n$ matrix is equal to $t_n = 2^{n(n-1)}$.

From Theorem F.1 we have the number of possible canonical UDMs $c_n = \sum_{r=0}^{n} \frac{n!2^{(n-r)r}}{r!(n-r)!} - n$.

Hence, we have

$$\lim_{n\to\infty} c_n / t_n = \lim_{n\to\infty} \frac{\sum_{r=0}^{n} \frac{n!2^{(n-r)r}}{r!(n-r)!} - n}{2^{n(n-1)}} = \lim_{n\to\infty} \frac{1 + n2^{n-1} + n(n-1)2^{2n-5} + \ldots + \frac{n!}{((n/2)!)^2} 2^{n^2/4} + \ldots + 1 - n}{2^{n(n-1)}}.$$

Without loss of generality, we can assume that n is even. Then, since each of the terms of the summation is less than or equal to the midterm $\frac{n!}{((n/2)!)^2} \frac{2^{n^2/4}}{2^{n(n-1)}}$, the summation of $n+2$ terms can be approximated as $S = (n+2) \frac{n!}{((n/2)!)^2} \frac{2^{n^2/4}}{2^{n(n-1)}}$. Since $c_n / t_n \leq S$, we can show that $\lim_{n\to\infty} S = 0$, and therefore $\lim_{n\to\infty} c_n / t_n = 0$. Using Stirling's approximation for $n!$,

$$\sqrt{2\pi} n^{n+1/2} e^{-n} \leq n! \leq e n^{n+1/2} e^{-n},$$

$$c_n / t_n \leq S = (n+2) \frac{n!}{((n/2)!)^2} \frac{2^{n^2/4}}{2^{n(n-1)}} \approx \frac{(n+2) e n^{n+1/2} e^{-n}}{(\sqrt{2\pi}(n/2)^{n/2+1/2} e^{-n/2})^2} \frac{2^{n^2/4}}{2^{n(n-1)}} = \frac{(n+2) n^{n+1/2} e^{1-n} 2^{n^2/4-(n^2-n)}}{2\pi (n/2)^{n+1} e^{-n}}$$

$$= \frac{e(n+2)n^{n+1/2}2^{n-3n^2/4}2^{n+1}}{2\pi n^{n+1}} = \frac{e(n+2)}{2\pi 2^{3n^2/4-2n-1}\sqrt{n}}.$$

We now show that $\lim_{n\to\infty} \frac{e(n+2)}{2\pi 2^{3n^2/4-2n-1}\sqrt{n}} = \frac{\infty}{\infty}$; indeterminate, so l'Hopital's rule can be applied after some algebraic manipulations: $\frac{e(n+2)}{2\pi 2^{3n^2/4-2n-1}\sqrt{n}} = \frac{e}{2\pi}\left[\frac{n}{2^{3n^2/4-2n-1}\sqrt{n}} + \frac{2}{2^{3n^2/4-2n-1}\sqrt{n}}\right]$. It is clear that $\lim_{n\to\infty} \frac{2}{2^{3n^2/4-2n-1}\sqrt{n}} = 0$ and $\frac{e}{2\pi} = const.$

Now, $\lim_{n\to\infty} \frac{n}{2^{3n^2/4-2n-1}\sqrt{n}} = \lim_{n\to\infty} \frac{1}{(2^{3n^2/4-2n-1})'\sqrt{n} + 2^{3n^2/4-2n-1}(\sqrt{n})'} =$

$$= \lim_{n\to\infty} \frac{1}{2^{3n^2/4-2n-1}\ln 2(3n/2-2)\sqrt{n} + 2^{3n^2/4-2n-1}/2\sqrt{n}}$$

$= \lim_{n\to\infty} \frac{1}{2^{3n^2/4-2n-1}[\ln 2(3n/2-2)\sqrt{n} + 1/2\sqrt{n}]} = 0$; therefore, $\lim_{n\to\infty} c_n / t_n = 0$ is also holds. Q.E.D.

By Theorem F.2, the relative number of canonical UDMs decreases with the number of attributes, indicating the increasing complexity of a decision problem with *n.* Theorem F.2 implies that the DM should try to aggregate the related attributes of the decision problem. This will substantially reduce the test size; provide an easier determination and assessment of canonical forms.

F4: Determining the number and convergence of proper partial forms

This subsection analyzes properties of proper partial forms since we are interested in their availability for decision problems under uncertainty.

Theorem F.3: *Let* p_n *be the number of proper partial UDMs for a decision problem with* n *attributes, and* t_n *be the total number of UDMs. Then,* $\lim_{n\to\infty} p_n \big/ t_n = 1.$

Proof. The total number of possible UDMs for a decision problem with $n \geq 2$ attributes is equal to $t_n = 2^{n(n-1)}$ as discussed previously.

To determine the number of possible proper partial UDMs, first note that any of the rows of the UDM can either contain a node in every of its column (corresponding to complement attributes) or does not contain, which counts for 2^{n-1} ways, then exclude a case when there are nodes in all columns (i.e., row is full). Finally, for a decision problem with $n \geq 2$ attributes the number of possible proper partial UDMs is equal to $p_n = (2^{n-1}-1)^n$.

Hence, we have $\lim_{n\to\infty} p_n \big/ t_n = \lim_{n\to\infty} \frac{(2^{n-1}-1)^n}{2^{n(n-1)}} = \lim_{n\to\infty} \left(\frac{2^{n-1}-1}{2^{n-1}} \right)^n = \lim_{n\to\infty} \left(1 - \frac{1}{2^{n-1}} \right)^n.$

We now present two alternative proofs to show that this limit converges to 1 as $n \to \infty$.

Short proof: It is known that if $\lim_{n\to\infty} a_n = a > 0$ and if $\lim_{n\to\infty} b_n = b,$ then $\lim_{n\to\infty} a_n^{b_n} = a^b$. By the known identity, $\lim_{n\to\infty} \left(1 - \frac{1}{2^{n-1}} \right)^{2^{n-1}} = \frac{1}{e} > 0,$ and utilizing l'Hopital's rule: $\lim_{n\to\infty} \frac{n}{2^{n-1}} = \lim_{n\to\infty} \frac{1}{2^{n-1}\ln 2} = 0;$ so

$$\lim_{n\to\infty} \left(1 - \frac{1}{2^{n-1}} \right)^n = \lim_{n\to\infty} \left[\left(1 - \frac{1}{2^{n-1}} \right)^{2^{n-1}} \right]^{\frac{n}{2^{n-1}}} = \lim_{n\to\infty} \left[\lim_{n\to\infty} \left(1 - \frac{1}{2^{n-1}} \right)^{2^{n-1}} \right]^{\frac{n}{2^{n-1}}} = \left(\frac{1}{e} \right)^0 = 1.$$

Hence, $\lim_{n\to\infty} p_n \big/ t_n = 1.$ Q.E.D.

Alternative proof: Let $y = \lim_{n\to\infty}\left(1-\frac{1}{2^{n-1}}\right)^n$.

Then $\ln y = \lim_{n\to\infty}\ln\left(1-\frac{1}{2^{n-1}}\right)^n = \lim_{n\to\infty} n\ln\left(1-\frac{1}{2^{n-1}}\right) = \lim_{n\to\infty}\frac{\ln\left(1-\frac{1}{2^{n-1}}\right)}{1/n} = \frac{0}{0}$; indeterminate, so we can utilize l'Hopital's rule after some algebraic manipulations.

$$\lim_{n\to\infty}\frac{\ln\left(1-\frac{1}{2^{n-1}}\right)}{1/n} = \lim_{n\to\infty}\frac{\ln\left(\frac{2^{n-1}-1}{2^{n-1}}\right)}{1/n} = \lim_{n\to\infty}\frac{\ln(2^{n-1}-1)-\ln(2^{n-1})}{1/n} =$$

$$\lim_{n\to\infty}\frac{\frac{1}{2^{n-1}-1}(2^{n-1}\ln 2)-\frac{1}{2^{n-1}}(2^{n-1}\ln 2)}{-1/n^2} =$$

$\lim_{n\to\infty}\frac{\frac{2^{n-1}\ln 2}{2^{n-1}-1}-\ln 2}{-1/n^2} = \lim_{n\to\infty}\frac{\ln 2\left(\frac{2^{n-1}}{2^{n-1}-1}-1\right)}{-1/n^2} = \lim_{n\to\infty}\frac{\frac{\ln 2}{2^{n-1}-1}}{-1/n^2} = \frac{0}{0}$; indeterminate, so we can apply l'Hopital's rule again: $\lim_{n\to\infty}\frac{\frac{\ln 2}{2^{n-1}-1}}{-1/n^2} = \lim_{n\to\infty}\frac{n^2\ln 2}{1-2^{n-1}} = \lim_{n\to\infty}\frac{2n\ln 2}{-2^{n-1}\ln 2} = \lim_{n\to\infty}\frac{n}{-2^{n-2}} = \frac{\infty}{-\infty}$; indeterminate, so we can apply l'Hopital's rule again: $\lim_{n\to\infty}\frac{n}{-2^{n-2}} = \lim_{n\to\infty}\frac{1}{-2^{n-2}\ln 2} = 0.$

So, $\ln y = 0; e^{\ln y} = e^0; y = 1.$

Hence, $y = \lim_{n\to\infty}\left(1-\frac{1}{2^{n-1}}\right)^n = \lim_{n\to\infty} p_n/t_n = 1.$ Q.E.D.

Theorem F.3 implies that the number of proper partial UDMs convergences to the total number of UDMs as $n\to\infty$. In other words, when the number of attributes is large, an arbitrary UDM is proper partial.

APPENDIX G: Elicitation of the utility function of attribute X_3 (safety of operations) in uranium extraction industry (in number of days without accident)

In tests below, please, write down in blank (____) spaces the number of days without accident, X_3, which make you indifferent between receiving the number for certain and drawing a lottery on the right hand side of the indifference mark ~.

(1) $X_{0.50}$ = _____days ~ 50% 1,825 days; 50% 365 days
(your answer)

(2) $X_{0.75}$ = _____days ~ 50% 1,825 days; 50% _____days put $X_{0.50}$ from (1)
(your answer)

(3) $X_{0.25}$ = _____days ~ 50% _____days put $X_{0.50}$ from (1); 50% 365 days
(your answer)

(4) X_{check} = _____days ~ 50% _____days put $X_{0.75}$ from (2); 50% _____days put $X_{0.25}$ from (3)
(your answer)

Was the test difficult for you?

Not at all Very

Circle one number: *1 2 3 4 5*

Your name: _____ Signature: _____

APPENDIX H: Assessment of scaling constants for the multilinear form of the multiattribute utility function

The attributes for the borehole restoration problem, their measures and ranges:

Attribute	Measure	Range	
		Best, x^*	Worst, x^0
X_1 - Profit (annual)	$	161,856	-70,866
X_2 - Labor saving	employees	10	0
X_3 - Safety of operations	days	1825	365
X_4 - Preserved area	m^2	6495	4000

1. What would you prefer between (please circle your preference; circle both in case of indifference):

a) $(x_1^0, x_2^*, x_3^*, x_4^*)$ and $(x_1^*, x_2^0, x_3^*, x_4^*)$

b) $(x_1^*, x_2^0, x_3^*, x_4^*)$ and $(x_1^*, x_2^*, x_3^0, x_4^*)$

c) $(x_1^*, x_2^*, x_3^0, x_4^*)$ and $(x_1^*, x_2^*, x_3^*, x_4^0)$

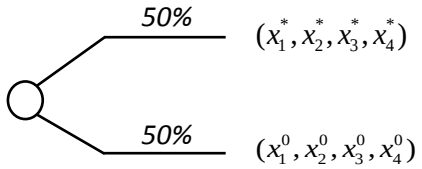
d) $(x_1^0, x_2^*, x_3^*, x_4^*)$ and $(x_1^*, x_2^*, x_3^0, x_4^*)$

e) $(x_1^0, x_2^*, x_3^*, x_4^*)$ and $(x_1^*, x_2^*, x_3^*, x_4^0)$

f) $(x_1^*, x_2^0, x_3^*, x_4^*)$ and $(x_1^*, x_2^*, x_3^*, x_4^0)$

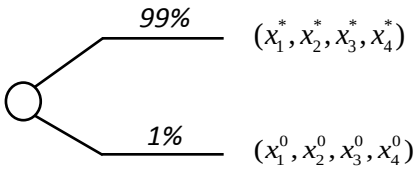
Consider the certain $(x_1^*, x_2^*, x_3^*, x_4^0)$ on the left hand side (LHS) and the lottery with 50-50 chances having either $(x_1^*, x_2^*, x_3^*, x_4^*)$ or $(x_1^0, x_2^0, x_3^0, x_4^0)$ on the right hand side (RHS).

2. Would you prefer LHS, RHS, or you are indifferent between LHS and RHS?

$(x_1^*, x_2^0, x_3^*, x_4^*)$ ~ 

Circle one out of three:	**1** *LHS*	**2** *I'm indifferent (LHS and RHS are equivalent)*	**3** *RHS*

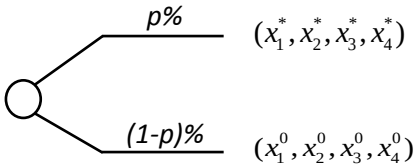
3. Would you prefer LHS, RHS, or you are indifferent between LHS and RHS?

$(x_1^*, x_2^0, x_3^*, x_4^*)$ ~ 

Circle one out of three:	**1** *LHS*	**2** *I'm indifferent (LHS and RHS are equivalent)*	**3** *RHS*

In tests below, what probability p makes you indifferent between LHS and RHS?

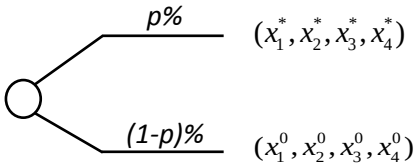
4. What probability p makes you indifferent between LHS and RHS?

$(x_1^*, x_2^0, x_3^*, x_4^*)$ ~ 

Your answer: $p =$ _____ %

---

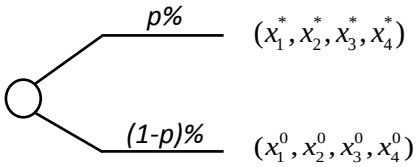
5. What probability p makes you indifferent between LHS and RHS?

$(x_1^*, x_2^*, x_3^0, x_4^*)$ ~ 

Your answer: $p =$ _____ %

---

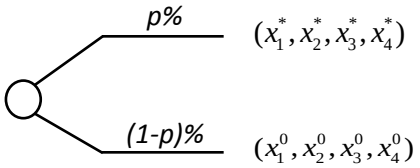
6. What probability p makes you indifferent between LHS and RHS?

$(x_1^*, x_2^*, x_3^*, x_4^0)$ ~ 

Your answer: $p =$ _____ %

---

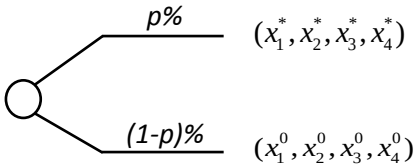
7. What probability p makes you indifferent between LHS and RHS?

$(x_1^0, x_2^*, x_3^*, x_4^*)$ ~ 

Your answer: $p =$ _____ %

---

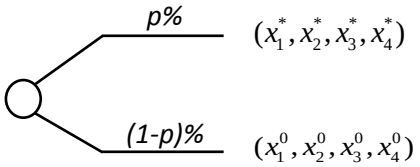
8. What probability p makes you indifferent between LHS and RHS?

$(x_1^0, x_2^0, x_3^*, x_4^*)$ ~ 

Your answer: $p =$ _____ %

---

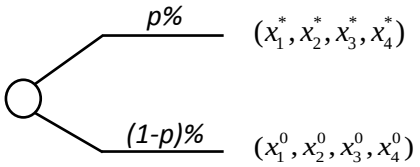
9. What probability p makes you indifferent between LHS and RHS?

$(x_1^0, x_2^*, x_3^0, x_4^*)$ ~ 

Your answer: $p =$ _____ %

---

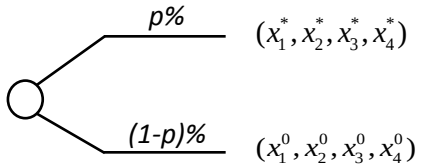
10. What probability p makes you indifferent between LHS and RHS?

$(x_1^0, x_2^*, x_3^*, x_4^0)$ ~ 

Your answer: $p =$ _____ %

---

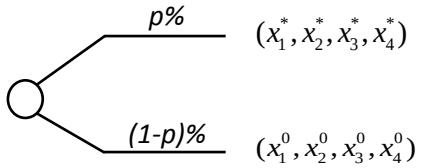
11. What probability p makes you indifferent between LHS and RHS?

$(x_1^*, x_2^0, x_3^0, x_4^*)$ ~ 

Your answer: $p =$ _____%

---

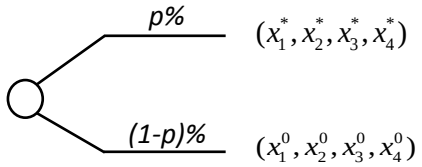
12. What probability p makes you indifferent between LHS and RHS?

$(x_1^*, x_2^0, x_3^*, x_4^0)$ ~ 

Your answer: $p =$ _____%

---

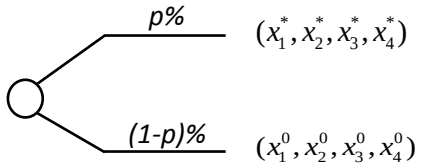
13. What probability p makes you indifferent between LHS and RHS?

$(x_1^*, x_2^*, x_3^0, x_4^0)$ ~ 

Your answer: $p =$ _____%

---

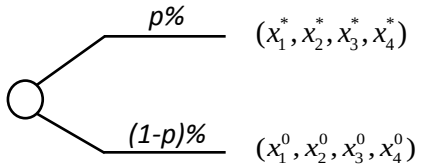
14. What probability p makes you indifferent between LHS and RHS?

$(x_1^0, x_2^0, x_3^0, x_4^*)$ ~ 

Your answer: $p =$ _____%

---

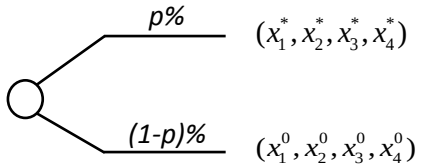
15. What probability p makes you indifferent between LHS and RHS?

$(x_1^0, x_2^0, x_3^*, x_4^0)$ ~ 

Your answer: $p =$ _____%

---

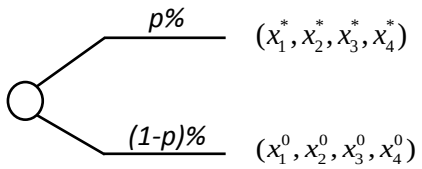
16. What probability p makes you indifferent between LHS and RHS?

$(x_1^0, x_2^*, x_3^0, x_4^0)$ ~ 

Your answer: $p =$ _____%

---

17. What probability p makes you indifferent between LHS and RHS?

$(x_1^*, x_2^0, x_3^0, x_4^0)$ ~ 

Your answer: $p =$ _____%

Was the test difficult for you?

	Not at all				Very
Circle one number:	*1*	*2*	*3*	*4*	*5*

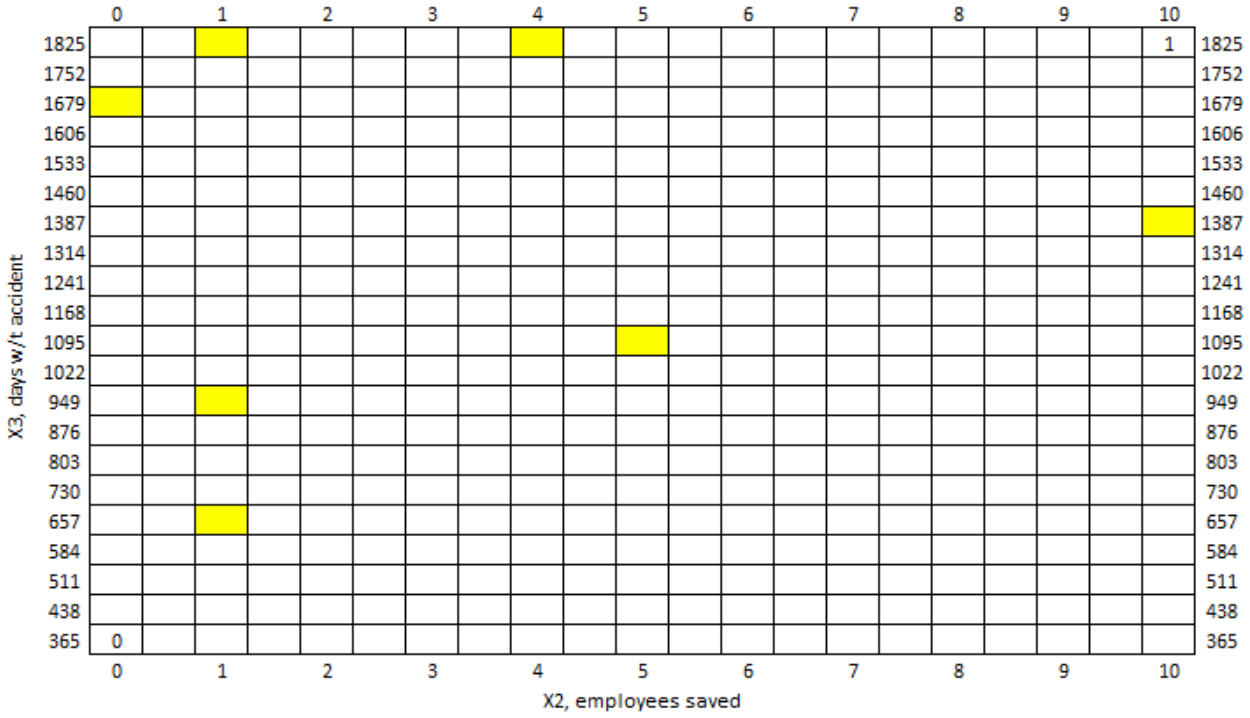
Your name: _____ **Signature:** _____

APPENDIX I: Assessments of the value function $V(x_2, x_3)$ when $(x_1 = x_1^*, x_4 = x_4^*)$. (The 1st and 2nd pages only).

The attributes for the borehole restoration problem, their measures and ranges:

Attribute	Measure	Range	
		Best, x^*	Worst, x^0
X_1 - Profit (annual)	$	161,856	-70,866
X_2 - Labor protection	employees	10	0
X_3 - Safety of operations	days	1825	365
X_4 - Preserved area	m^2	6495	4000

Consider a two-dimensional space of attributes X_2 and X_3, where the attributes range from their worst values to their best values shown in Table above. Let us set $V(x_2^0, x_3^0) = 0$ and $V(x_2^*, x_3^*) = 1$. Assume that the other two attributes are at their best values, $(x_1 = x_1^*, x_4 = x_4^*)$. Please answer the questions below. You can give a rough answer.



1. Suppose that $(x_2 = 1 \text{ employee}, x_3 = 657 \text{ days})$. Since $V(x_2^0, x_3^0) = 0$ and $V(x_2^*, x_3^*) = 1$, what numerical value would you assign for point $V(x_2 = 1 \text{ employee}, x_3 = 657 \text{ days})$?

Your answer: $V(x_2 = 1 \text{ employee}, x_3 = 657 \text{ days}) =$ _____.

2. Keep $(x_2 = 1 \text{ employee}, x_3 = 657 \text{ days})$. If X_2 were <u>decreased</u> to 0, how many days in X_3 would you need to compensate it? What will be the new value of X_3?

Your answer: $x_3 =$ _____ days

---

3. Keep $(x_2 = 1 \text{ employee}, x_3 = 657 \text{ days})$. If X_2 were <u>increased</u> up to 3 employees $(x_2 = 3)$, how many days you will give up in X_3? What will be the new value of X_3?

Your answer: $x_3 =$ _____ days

---

4. Suppose that $(x_2 = 1 \text{ employee}, x_3 = 949 \text{ days})$. Since $V(x_2^0, x_3^0) = 0$ and $V(x_2^*, x_3^*) = 1$, what numerical value would you assign for point $V(x_2 = 1 \text{ employee}, x_3 = 949 \text{ days})$?

Your answer: $V(x_2 = 1 \text{ employee}, x_3 = 949 \text{ days}) =$ _____.

---

5. Suppose that $(x_2 = 1 \text{ employee}, x_3 = 949 \text{ days})$. If X_2 were <u>decreased</u> by 1 unit, how many days in X_3 would you need just to compensate it? What would be the new value of X_3?

Your answer: $x_3 =$ _____ days

---

6. Keep $(x_2 = 1 \text{ employee}, x_3 = 949 \text{ days})$. If X_2 were <u>increased</u> by 1 unit, how many days you will give up in X_3? What will be the new value of X_3?

Your answer: $x_3 =$ _____ days

---

7. Keep $(x_2 = 1 \text{ employee}, x_3 = 949 \text{ days})$. If X_2 were <u>increased</u> to $x_2 = 4$, how many days you will give up in X_3? What will be the new value of X_3?

Your answer: $x_3 =$ _____ days

---

8. Suppose that you can increase X_2 for 1 unit. Would you pay the same amount of X_3 days to go from $x_2 =$ 0 to 1 as from $x_2 =$ 6 to 7?

Circle one out of three:	**1** *Pay more to go from* $x_2 =$ 0 to 1	**2** *Same*	**3** *Pay more to go from* $x_2 =$ 6 to 7

APPENDIX J: Elicitation of the utility function over the value function

$U_V(V(x_2, x_3))$ **when** $(x_1 = x_1^*, x_4 = x_4^*)$.

In tests below, please, write down in blank (____) spaces a numerical value that makes you indifferent between receiving the value for sure and drawing a lottery on the right hand side of the indifference mark ~. Recall, $V(x_2, x_3)$ is in employee-days. Assume that the values of the other two attributes are fixed at $(x_1 = x_1^*, x_4 = x_4^*)$.

(1) $V_{0.50}(x_2, x_3) =$ _____ **(your answer)** ~ 50% $V(x_2, x_3) = 1$; 50% $V(x_2, x_3) = 0$

(2) $V_{0.75}(x_2, x_3) =$ _____ **(your answer)** ~ 50% $V(x_2, x_3) = 1$; 50% _____ put $V_{0.50}$ from (1)

(3) $V_{0.25}(x_2, x_3) =$ _____ **(your answer)** ~ 50% _____ put $V_{0.50}$ from (1); 50% $V(x_2, x_3) = 0$

(4) $V_{check}(x_2, x_3) =$ _____ **(your answer)** ~ 50% _____ put $V_{0.75}$ from (2); 50% _____ put $V_{0.25}$ from (3)

Was the test difficult for you?

Not at all Very

Circle one number: *1 2 3 4 5*

Your name: _____ **Signature:** _____

APPENDIX K: C++ code of the exclusion algorithms

```
// File name: main.cpp
// Comparison of Version 1 (brute-force approach) and
// Version 2 (Twos-Complement Exclusion algorithm).
// Yerkin Abdildin, abdildin@illinois.edu
// Implemented as a vector of vectors on Dec 16, 2011.
// Updated on July 3, 2014.

#include <iostream>
using std::cout;
using std::cin;
using std::endl;

#include <vector>
using std::vector;

#include <cstdlib>
using std::rand;
using std::srand;

#include <ctime>

// Functions prototypes
void  exclusionV1(int m, int n, vector< vector<short> > &M1);
void  exclusionV2(int m, int n, vector< vector<short> > &M2);

const int testNums = 10000;      // NUMBER OF TESTS
const int n = 10;                // no. or cols of matrix M
int m = 5000;                    // no. or rows of matrix M

double time1 = 0;
double totalTime1 = 0;
double time2 = 0;
double totalTime2 = 0;

int main() {
        printf("m = %d, n = %d\n", m, n);
        int curTest = 1;
        int testsPassed = 0;

        for (int i = 0; i < testNums; i++){
                // Allocate a two-dimensional mxn matrix M of short integers
                vector< vector<short> > M(m, vector<short>(n));

                vector< vector<short> > M1;
                vector< vector<short> > M2;

                //Randomization of input data for M
                srand(time(0) + i*2 + 1);
                for (int i = 0; i < m; i++){
                        for (int j = 0; j < n; j++) {
                                M[i][j] = (rand() % 3);
                        }
                }

                M1 = M; //using overloaded assignment operator
                M2 = M;
                M.clear();

                //cout << "ALGORITHM 1 " << endl;
                clock_t startTime1 = clock(); // shows execution time of Alg 1
                exclusionV1(m, n, M1);
                time1 = (double( clock() - startTime1 ) / (double)CLOCKS_PER_SEC); // stops time
                //cout << time1 << " seconds." << endl;
                totalTime1 = totalTime1 + time1;
                time1 = 0;
```

```
                //cout << "ALGORITHM 2 " << endl;
                clock_t startTime2 = clock(); // shows execution time of Alg 2
                exclusionV2(m, n, M2);
                time2 = (double( clock() - startTime2 ) / (double)CLOCKS_PER_SEC); // stops time
                //cout << time2 << " seconds." << endl;
                totalTime2 = totalTime2 + time2;
                time2 = 0;

                //Test if M1out and M2out are identical
                if (M1 == M2){
                        testsPassed = testsPassed + 1;
                }
                if (testsPassed == testNums)
                        cout << "All " << testsPassed << " tests are passed!" << endl;

                M1.clear();
                M2.clear();
                curTest = curTest + 1;
        } // end of For loop of testNums
        cout << "Average time of Algorithm 1 = " << totalTime1 / testNums  << " seconds." << endl;
        cout << "Average time of Algorithm 2 = " << totalTime2 / testNums  << " seconds." << endl;
        cout << "Total time of all tests = " << totalTime1 + totalTime2 << " seconds." << endl;

    system("PAUSE");
    return 0;
}

// Version 1 (brute-force approach)
void exclusionV1(int m, int n, vector< vector<short> > &M1)      {
                int rows = m;
                int r = 0; // outerloop index starts with 1st row
                while (r < rows){
                        int i = 0; // inner loop index
                        while (i < rows){
                                if (i != r){
                                        //start comparing row r and row i
                                        bool flgGE = true;
                                        for (int j = 0; j < n; j++){
                                                if (M1[r][j] < M1[i][j]){
                                                        flgGE = false;
                                                        break; //break out of the for loop
                                                }
                                        }
                                        if (flgGE == true){
                                                //if all elmns of row r >= elmns of row i
                                                bool redundant = true;
                                                for (int j = 0; j < n; j++) {
                                                        if ((M1[r][j] > M1[i][j]) && (M1[r][j] == 1)){
                                                                redundant = false;
                                                                break; //break out of the for loop
                                                        }
                                                }
                                                //Remove redundant row i
                                                if (redundant == true) {
                                                        M1.erase(M1.begin() + i);
                                                        rows = rows - 1;
                                                        if (i < r)
                                                                r = r - 1;
                                                        i = i - 1;
                                                }
                                        }
                                }
                                i = i + 1;
                        } // end of inner loop
                        r = r + 1;
                } // end of outer loop
}
```

```
//Version 2 (Twos-Complement Exclusion algorithm)
void exclusionV2(int m, int n, vector< vector<short> > &M2)      {

                int rows = m;
                //define a set var or vector twos_compl
                //to store indices of 1s and 0s in row r
                vector<short> twos_compl;

                int r = 0; // outerloop index starts with 1st row
                while (r < rows){
                        for (int j = 0; j < n; j++) {
                                if (M2[r][j] != 2)
                                        twos_compl.push_back(j);
                        }

                        if (twos_compl.empty()) {
                                vector<short> twos;
                                twos.assign(n,2);
                                M2.clear();
                                M2.push_back(twos);
                                break;
                        }

                        int i = 0; // inner loop index
                        while (i < rows){
                                if (i != r){

                                        //check if elmns of twos_compl of r and i are identical
                                        int size = twos_compl.size();
                                        bool identical = true;
                                        for ( int j = 0; j < size; j++ ) {
                                                int curr = twos_compl[j];
                                                if (M2[r][ curr ] !=  M2[i][ curr ]) {
                                                        identical = false;
                                                        break; // break out the for loop
                                                }
                                        } // end of for loop

                                        if (identical == true) {
                                                M2.erase(M2.begin() + i);
                                                rows = rows - 1;
                                                if (i < r)
                                                        r = r - 1;
                                                i = i - 1;
                                        }
                                }
                                i = i + 1;
                        } // end of inner loop
                        twos_compl.clear();
                        r = r + 1;
                } // end of outer loop
}
```

APPENDIX L: MATLAB code of the Ternary-Decimal Exclusion algorithm

```
function [M] = Ternary_Decimal(M)
 % Removes redundant/duplicate rows from a matrix M(m,n), returns new M.
 % Here, M is a ternary matrix of m utility assessments for n-attribute
 % decision problem.
 % Yerkin Abdildin, abdildin@illinois.edu, 2014
 % Updated July 3, 2014
 % Notations:
 % 0 = x0, least preferable value of attribute X
 % 1 = x*, most preferable value of attribute X
 % 2 = x, all values of attribute X
 % x includes both x* and x0, but the latter two are distinct.

    m = size(M,1);      % find the # of rows of matrix M
    n = size(M,2);      % find the # of cols of matrix M
    min = 0;
    max = 3^n-1;

    % Step 1. Convert rows of M into decimal numbers and store them in
    % vector V(m)
    V = zeros(1,m);
    for i = 1:m
        d = 0;
        for j = 1:n
            d = d + (M(i,j))*3^(n-j);
        end
        if d == max     % if vector of 2s is found, then terminate
            M = M(i,:);
            return;
        end
        V(i) = d;
    end

    % Step 2. Sort V and eliminate duplicates:
    V = unique(V);

    % Step 3. Eliminate redundant assessments from V:
    V = exclude_redundant(V, min, max);

    % Step 4. Convert V back to M
    rows = length(V);
    M = dec2ternary(V, rows, n);

    % Step 5. Eliminate from M specific redundant rows, if necessary.
    r = 1;                              % start from the first row
    while r < rows + 1
        twos_compl = find(M(r,:) ~= 2); % indices of 0s and 1s in row r
        i = r + 1;                      % row i will be compared with row r
        while i < rows + 1
            % compare elms of r and i by indices in twos_compl
            if isequal(M(r,twos_compl), M(i,twos_compl))
                M(i,:) = [];
                rows = rows - 1;
                i = i - 1;
```

```
            end
            i = i + 1;
        end % end of inner loop
        r = r + 1;
    end % end of outer loop
end


function [V] = exclude_redundant(V, min, max)
 % Removes redundant decimal numbers from V, returns new V.
 % Yerkin Abdildin, abdildin@illinois.edu, 2014

    m = length(V);
    if m < 2                            % base condition
        return;
    end

    if (V(m) == max - 1) && (V(m - 1) == max - 2)
        counter = 1;
        while (counter < m-1)
            res = mod(V(counter),3);
            if (res == 1) || (res == 0)
                V(counter) = [];
                m = m - 1;
                counter = counter - 1;
            end
            counter = counter + 1;
        end
    elseif V(m) == max - 1
        counter = 1;
        while (counter < m)
            res = mod(V(counter),3);
            if res == 1
                V(counter) = [];
                m = m - 1;
                counter = counter - 1;
            end
            counter = counter + 1;
        end
    elseif V(m) == max - 2
        counter = 1;
        while (counter < m)
            res = mod(V(counter),3);
            if res == 0
                V(counter) = [];
                m = m - 1;
                counter = counter - 1;
            end
            counter = counter + 1;
        end
    end
    m = length(V);

    lb = floor(min + (max-min)/3);      % left bound
    mb = floor(min + 2*(max-min)/3);    % mid bound
```

```
    % find elm <= lb to create L, Left sub-vector
    idxL = search_bound(V, 1, m, lb);
    if idxL == 0
        L = [];
    elseif V(idxL) == lb
        L = lb;
    else
        L = V(1 : idxL);
    end

    % find elm <= mb to create M, Mid sub-vector
    idxM = search_bound(V, idxL + 1, m, mb);
    if idxM == idxL
        M = [];
    elseif V(idxM) == mb
        M = mb;
    else
        M = V(idxL + 1 : idxM);
    end

    % create R, Right sub-vector
    if m == idxM
        R = [];
    else
        R = V(idxM + 1 : m);
    end

    % recursive calls
    [L] = exclude_redundant(L, min, lb);
    [M] = exclude_redundant(M, lb, mb);
    [R] = exclude_redundant(R, mb, max);

    % merging sub-vectors L, M, and R
    [V] =[L M R];
end

function [idx] = search_bound(V, left, right, x)
 % The recursive version of a binary search of x (right bound) in V
 % Yerkin Abdildin, abdildin@illinois.edu, 2014

    if left > right
        idx = right;
        return;
    else
        mid = floor(left + (right - left)/2);
        if x < V(mid)
            [idx] = search_bound(V, left, mid - 1, x);
        elseif x > V(mid)
            [idx] = search_bound(V, mid + 1, right, x);
        else
            idx = mid;
            return;
        end
    end
end
```

```
function M = dec2ternary(V, length, n)
 % Converts a decimal vector V(length) to a ternary matrix M(length, n).
 % Yerkin Abdildin, abdildin@illinois.edu, 2014

    M = zeros(length, n);
    for rows = 1:length
        D = V(length - rows + 1);        % D is a decimal number
        for cols = n:-1:1
            M(rows,cols) = mod(D, 3);    % remainder goes to M(rows,cols)
            D = floor(D/3);              % D is now a quotient
        end
    end
end
```

REFERENCES

Abbas, A.E., "An Entropy Approach for Utility Assignment in Decision Analysis," In: C. Williams (ed.), Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Moscow, Idaho, August 3-7, 2002, AIP Conference Proceedings 659, American Institute of Physics, pages 328-338, Melville, New York, 2002a.

Abbas, A.E., "Entropy Methods for Univariate Distributions in Decision Analysis," In: C. Williams (ed.), Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Moscow, Idaho, August 3-7, 2002, AIP Conference Proceedings 659, American Institute of Physics, pages 339-349, Melville, New York, 2002b.

Abbas, A.E., "Entropy Methods for Adaptive Utility Elicitation," IEEE Transactions on Systems, Science and Cybernetics, 34 (2), pages 169-178, 2004.

Abbas, A.E., "Maximum Entropy Distributions between Upper and Lower Bounds," In: K.H. Knuth, A.E. Abbas, R.D. Morris, J.P. Castle (eds.), Bayesian Inference and Maximum Entropy Methods in Science and Engineering, San Jose, California, August 7-12, 2005, AIP Conference Proceedings 803, American Institute of Physics, pages 25-42, Melville, New York, 2005.

Abbas, A.E., "Entropy Methods for Joint Distributions in Decision Analysis," IEEE Transactions on Engineering Management, 53 (1), pages 146-159, 2006.

Abbas, A.E., "Invariant Utility Functions and Certain Equivalent Transformations," Decision Analysis, 4 (1), pages 17-31, 2007.

Abbas, A.E., "Invariant Multiattribute Utility Functions," Theory and Decision, Special Issue on Foundations of Utility and Risk, (FUR), 68 (1, 2), pages 69-99, 2009a.

Abbas, A.E., "From Bayesian Nets to Utility Nets," Bayesian Inference and Maximum Entropy Methods in Science and Engineering, Oxford, Mississippi, AIP Conference Proceedings 1193, pages 3-12, 2009b.

Abbas, A.E., "Multiattribute Utility Copulas," Operations Research, 57 (6), pages 1367-1383, 2009c.

Abbas, A.E., "General decompositions of multiattribute utility functions with partial utility independence." Journal of Multicriteria Decision Analysis, 17, pages 37-59, 2010a.

Abbas, A.E, "Constructing Multiattribute Utility Functions for Decision Analysis," INFORMS Tutorials in Operations Research, editor Hasenbein, J. Hanover, Maryland, pages 62-98, 2010b.

Abbas, A.E., "Decomposing the Cross Derivatives of a Multiattribute Utility Function into Risk Attitude and Value," Decision Analysis, 8 (2), pages 103-116, 2011a.

Abbas, A.E., "The Multiattribute Utility Tree," Decision Analysis, 8 (3), pages 165-169, 2011b.

Abbas, A.E., "Valuing Changes in Investment Opportunities," Operations Research, 60 (6), pages 1451-1460, 2012.

Abbas, A.E., "Utility Copula Functions Matching all Boundary Assessments," Operations Research, 61(2), pages 359-371, 2013.

Abbas, A.E. and J. Aczél, "The Role of Some Functional Equations in Decision Analysis," Decision Analysis, 7 (2), pages 215-228, 2010.

Abbas, A.E., J. Aczél, and J. Chudziak, "Invariance of Multiattribute Utility Functions Under Shift Transformations," Results in Mathematics, 54, pages 1-13, 2009.

Abbas, A.E. and D.E. Bell, "One-Switch Independence for Multiattribute Utility Functions," Operations Research, 59 (3), pages 764-771, 2011.

Abbas, A.E. and D.E. Bell, "One-Switch Conditions for Multiattribute Utility Functions," Operations Research, 60 (5), pages 1199-1212, 2012.

Abbas, A.E. and J. Chudziak, "One-Switch Utility Functions with Annuity Payments," Applied Mathematics and Computation, 219 (14), pages 7699-7710, 2013.

Abbas, A.E. and R.A. Howard, "Attribute Dominance Utility," Decision Analysis, 2 (4), pages 185-206, 2005.

Abbas, A.E., J.E. Matheson and R.F. Bordley, "Effective Utility Functions Induced by Organizational Target-Based Incentives," Managerial and Decision Economics, 30 (4), pages 235-251, 2009.

Abbas, A.E. and J.E. Matheson, "Normative Decision Making with Multiattribute Performance Targets," Journal of Multicriteria Decision Analysis, 16 (3, 4), pages 67-78, 2010. Winner of Best Decision Analysis Publication Award for 2010.

Abbas, A.E., D.V. Budescu, H.T. Yu, and R. Haggerty, "A Comparison of Two Probability Encoding Methods: Fixed Probability Wheel vs. Fixed Variable Values," Decision Analysis, 5 (4), pages 190-202, 2008.

Abdildin, Y.G. and A.E. Abbas, "An Algorithm for Excluding Redundant Assessments in a Multiattribute Utility Problem," Proceedings of the International Conference on Computational Science, ICCS 2012, pages 802-811, Omaha, Nebraska, June 4-6, 2012.

Abdildin, Y.G. and A.E. Abbas, "Canonical Multiattribute Utility Functions: Enumeration, Verification, and Application," Proceedings of the International Conference on Computational Science, ICCS 2013, pages 2288-2297, Barcelona, Spain, June 5-7, 2013.

Anderson, R.M. and R. Clemen, "Toward an Improved Methodology to Construct and Reconcile Decision Analytic Preference Judgments," Decision Analysis, 10 (2), pages 121-134, 2013.

Arrow, K.J., "The theory of risk aversion," in *Aspects of the Theory of Risk-Bearing*, Yrjö Jahnssonin Säätiö, Helsinki, Finland, 1965.

Bacchus, F. and A. Grove, "Graphical models for preference and utility," in Proceedings of the Eleventh conference on Uncertainty in Artificial Intelligence, pages 3-10, 1995

Bakır, N.O., "A Decision Tree Model for Evaluating Countermeasures to Secure Cargo at United States Southwestern Ports of Entry," Decision Analysis, 5 (4), pages 230-248, 2008.

Baucells, M. and C. Rata, "Survey Study of Factors Influencing Risk-Taking Behavior in Real-World Decisions Under Uncertainty," Decision Analysis, 3 (3), pages 163-176, 2006.

Bell, D.E., "Multiattribute utility functions: Decompositions using interpolation," Management Science, 25 (8), pages 744-753, 1979a.

Bell, D.E., "Consistent assessment procedures using conditional utility functions," Operations Research, 27 (5), pages 1054-1066, 1979b.

Bell, D.E., "One-Switch Utility Functions and a Measure of Risk," Management Science, 34 (12), pages 1416-1424, 1988.

Bell, D.E. and P.C. Fishburn, "Strong One-Switch Utility," Management Science, 47 (4), pages 601-604, 2001.

Bodily, S.E. and P.E. Pfeifer, "Darden's Luckiest Student: Lessons from a High-Stakes Risk Experiment," Decision Analysis, 7 (4), pages 331-345, 2010.

Borcherding, K., T. Eppel, and D. von Winterfeldt, "Comparison of Weighting Judgments in Multiattribute Utility Measurement," Management Science, 37 (12), pages 1603-1619, 1991.

Boutilier, C., "The Influence of Influence Diagrams on Artificial Intelligence," Decision Analysis, 2 (4), pages 229-231, 2005.

Boutilier, C., R.I. Brafman, H.H. Hoos, and D. Poole, "Reasoning with conditional ceteris paribus preference statements," In Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence, pages 71-80, Stockholm, 1999.

Boutilier, C., F. Bacchus, and R.I. Brafman, "UCP-networks: a directed graphical representation of conditional utilities," In Proceedings of the 17th Conference on Uncertainty in Artificial Intelligence, pages 56-64, Seattle, 2001.

Budescu, D., A. Abbas, and L. Wu, "Does probability weighting matter in probability elicitation?" Journal of Mathematical Psychology, 55, pages 320-327, 2011.

Buede, D.M., "Influence Diagrams: A Practitioner's Perspective," Decision Analysis, 2 (4), pages 235-237, 2005.

Burns, W.J. and R.T. Clemen, "Covariance Structure Models and Influence Diagrams," Management Science, 39 (7), pages 816-834, 1993.

Caswell, D.J., R.A. Howard, and M.E. Paté-Cornell, "Analysis of National Strategies to Counter a Country's Nuclear Weapons Program," Decision Analysis, 8 (1), pages 30-45, 2011.

Chudziak, J., "On a Class of One-Switch Multiattribute Utility Functions," ACTA PHYSICA POLONICA A, 121 (2-B), pages B-11-B-15, 2012.

Cormen, T.H., C.E. Leiserson, and R.L. Rivest. *Introduction to Algorithms*. The MIT Press, 1997.

Detwarasiti, A. and R.D. Shachter, "Influence Diagrams for Team Decision Analysis," Decision Analysis, 2 (4), pages 207-228, 2005.

DeWispelare, A.R., L.T. Herren, and R.T. Clemen, "The use of probability elicitation in high-level nuclear waste regulation program," International Journal of Forecasting, 11, pages 5-24, 1995.

Dyer, J.S., T. Edmunds, J.C. Butler and J. Jia, "A Multiattribute Utility Analysis of Alternatives for the Disposition of Surplus Weapons-Grade Plutonium," Operations Research, 46 (6), pages 749-762, 1998.

Engel, Y. and M. Wellman, "CUI networks: a graphical representation for conditional utility independence," Journal of Artificial Intelligence Research, 31, pages 83-112, 2008.

Eppel, T. and D. von Winterfeldt, "Value-of-Information Analysis for Nuclear Storage Tanks," Decision Analysis, 5 (3), pages 157-167, 2008.

Farquhar, P.H., "Utility Assessment Methods," Management Science, 30 (11), pages 1283-1300, 1984.

Farquhar, P.H., "A Fractional Hypercube Decomposition Theorem for Multiattribute Utility Functions," Operations Research, 23 (5), pages 941-967, 1975.

Felli, J.C. and G.B. Hazen, "Javelin Diagrams: A Graphical Tool for Probabilistic Sensitivity Analysis," Decision Analysis, 1 (2), pages 93-107, 2004.

Fishburn, P.C., "Approximations of Two-Attribute Utility Functions," Mathematics of Operations Research, 2 (1), pages 30-44, 1977.

Goodrich, M.T., R. Tamassia, and D. Mount. *Data Structures and Algorithms in C++*. John Wiley & Sons, Inc., 2004.

Gros, J., "Power Plant Siting: a Paretian Environmental Approach," Nuclear Engineering and Design, 34, pages 281-292, 1975.

Gros, J., R. Avenhaus, J. Linnerooth, P.D. Pahner, and H.J. Otway, "A systems analysis approach to nuclear facility siting," Behavioral Science, 21 (2), pages 116-127, 1976.

Harvey, C.M., "Multiattribute Risk Linearity," Management Science, 39 (3), pages 389-394, 1993.

Hayes, B., "Third Base," American Scientist, 89 (6), 2001.

Howard, R.A., "Decision Analysis: Applied Decision Theory," Proceedings of the Fourth International Conference on Operational Research, pages 55-77, 1966.

Howard, R.A. and J.E. Matheson, "Influence Diagrams" (1981) in *The principles and Applications of Decision Analysis*, Vol. II, pages 719-762, R.A. Howard and J.E. Matheson (eds.). Strategic Decisions Group, Menlo Park, California, USA, 1983.

Howard, R.A. and J.E. Matheson. *The Principles and Applications of Decision Analysis*. Menlo Park: Strategic Decisions Group. Volumes I and II. 1983.

Howard, R.A., "Knowledge Maps," Management Science, 35 (8), pages 903-922, 1989.

Howard, R.A., "In praise of the old time religion," in *Utility Theories: Measurements and Applications*. Kluwer, Boston, pages 27-55, 1992.

Howard, R.A., "Decision Analysis: Practice and Promise," Management Science, 34 (6), pages 679-695, 1988.

Howard, R. and J. Matheson, "Influence Diagrams," Decision Analysis, 2 (3), pages 127-143, 2005a.

Howard, R. and J. Matheson, "Influence Diagram Retrospective," Decision Analysis, 2 (3), pages 144-147, 2005b.

IEO2013, International Energy Outlook 2013, http://www.eia.gov/forecasts/ieo/

Jantzen, S.G., B.J.G. Sutherland, D. R. Minkley, and B. F. Koop, "GO Trimming: Systematically reducing redundancy in large Gene Ontology datasets," BMC Research Notes, 4 (267), pages 1-9, 2011.

Kazatomprom, 2014. http://kazatomprom.kz/en/#!/industry/uranium/uranium_mining

Keefer, D.L., "Resource Allocation Models with Risk Aversion and Probabilistic Dependence: Offshore Oil and Gas Bidding," Management Science, 37 (4), pages 377-395, 1991.

Keeney, R.L. and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value tradeoffs*. Wiley, New York. 1976.

Keeney, R.L., "Analysis of Preference Dependencies among Objectives," Operations Research, 29 (6), pages 1105-1120, 1981.

Keeney, R.L. and A. Sicherman, "Illustrative Comparison of One Utility's Coal and Nuclear Choices," Operations Research, 31 (1), pages 50-83, 1983.

Keeney, R.L., "Making better decision makers," Decision Analysis 1 (4), pages 193-204, 2004.

Keeney, R.L., O. Renn, and D. von Winterfeldt, "Structuring West Germany's energy objectives," Energy Policy, 15, pages 352-362, 1987.

Kirkwood, C.W., "Parametrically Dependent Preferences for Multiattributed Consequences," Operations Research, 24 (1), pages 92-103, 1976.

Kirkland, G. 2010. Chevron's Channel on YouTube.
http://www.youtube.com/chevron#p/u/12/JRCxZA6ay3M

Kleinberg, J. and E. Tardos. *Algorithm Design*. Addison Wesley, 2006.

Knoop, J., O. Ruthing, and B. Steffen, "Lazy code motion," ACM SIGPLAN Notices, 27 (7), pages 224-234, 1992.

Knuth, D. E. *The Art of Computer Programming, Volume 4A: Combinatorial Algorithms.* Pearson Education, Inc., reprint of 2011.

Matheson, J.E. and A.E. Abbas, "Utility Transversality: A Value-Based Approach," Journal of Multicriteria Decision Analysis, 13, pages 229-238, 2005.

Merkhofer, M.W. and R.L. Keeney, "Multiattribute Utility Analysis of Alternative Sites for the Disposal of Nuclear Waste," Risk Analysis, 7 (2), pages 173-194, 1987.

Merrick, J.R.W. and L.A. McLay, "Is Screening Cargo Containers for Smuggled Nuclear Threats Worthwhile?" Decision Analysis, 7 (2), pages 155-171, 2010.

Miklas, M.P., J. Norwine, A.R. DeWispelare, L.T. Herren, and R.T. Clemen, "Future climate at Yucca Mountain, Nevada proposed high-level radioactive waste repository," Global Environment Change, 5 (3), pages 221-234, 1995.

Morel, E. and C. Renvoise, "Global optimization by suppression of partial redundancies," Communication of the ACM, 22 (2), pages 96-103, 1979.

Owen, D.L., J.E. Matheson, and R.A. Howard, "The Value of Life and Nuclear Design" (1978) in *The principles and Applications of Decision Analysis,* Vol. II, pages 507-520, R.A. Howard and J.E. Matheson (eds.). Strategic Decisions Group, Menlo Park, California, USA, 1983.

Paleri, V.K., Y.N. Srikant, and P. Shankar, "Partial redundancy elimination: a simple, pragmatic, and provably correct algorithm," Science of Computer Programming, 48, pages 1-20, 2003.

Pauker, S.G. and J.B. Wong, "The Influence of Influence Diagrams in Medicine," Decision Analysis, 2 (4), pages 238-244, 2005.

Pratt, J., "Risk aversion in the small and in the large," Econometrica, 32 (1, 2), pages 122-136, 1964.

Ross, S.A., "Some stronger measures of risk aversion in the small and the large with applications," Econometrica, 49 (3), pages 621-638, 1981.

Shaffer, C.A. *A Practical Introduction to Data Structures and Algorithm Analysis*. Prentice Hall, (2nd eds.), 2001.

Solomon, B.D. and K.E. Haynes, "A survey and critique of multiobjective power plant siting decision rules," Socio-Economic Planning Sciences, 18 (2), pages 71-79, 1984

Standish, T.A. *Data Structures, Algorithms, and Software Principles in C*. Addison Wesley, 1995.

Sushko, S.M., A.D. Begun., and G.N. Abdildin, "Features of the Construction of the Technological Wells for Exploitation in Uranium Ore Deposits of Kazakhstan," in *Uranium of Kazakhstan,* VOLKOVGEOLOGIA, Almaty, 2008.

Tamura, H. and Y. Nakamura, "Decompositions of Multiattribute Utility Functions Based on Convex Dependence," Operations Research, 31 (3), pages 488-506, 1983.

Uraniumletter International, http://uraniumletterint.com

U.S. Energy Information Administration, http://www.eia.gov/tools/faqs/faq.cfm?id=447&t=1 Accessibility verified on 6/20/2014.

von Neumann, J. and O. Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, Princeton, NJ, 1947.

Webster, M., "Incorporating Path Dependency into Decision-Analytic Methods: An Application to Global Climate-Change Policy," Decision Analysis, 5(2), pages 60-75, 2008.

World Energy Council, www.WorldEnergy.org, 2013

World Nuclear Association, http://www.world-nuclear.org/info/inf23.html